# Förord

Denna manual är avsedd för programmering av PLC-systemen MELSEC QnA(S) och Q från Mitsubishi Electric.

Manualen är uppdelad i två delar, dels en operand- och funktionsbeskrivningsdel, dels en instruktionsdel som omfattar nedanstående system.

## Operand- och funktionsdel, kap 2-4

Denna del är en introduktion till MELSEC Q och QnA(S), avsedd att beskriva operander och funktioner på ett lättbegripligt sätt.

## Instruktionsdel kap, 5-8

Denna del ska du använda då du behöver veta mer i detalj hur du programmerar dina MELSEC QnA(S) och Q-PLC-system.

Denna manual omfattar följande modeller i QnA(S)och Q-serien

| | |
|---|---|
| QnA | Q2A<br>Q2A-S1<br>Q3A<br>Q4A |
| QnAS | Q2AS<br>Q2AS-S1 |
| QnA(S) | QnA+QnAS |
| Q4AR | Q4AR |
| Q | Q00J, Q00,<br>Q01, Q02,<br>Q02H, Q06H,<br>Q12H, Q25H |
| Alla | QnA(S)+Q4AR+<br>Q |

© Beijer Electronics AB, MA00509A, 2003-12

Alla exempel i manualen är enbart ämnade för att öka förståelsen av systemets funktionalitet och handhavande. Beijer Electronics AB tar inget ansvar om dessa exempel används i verkliga applikationer.
På grund av det stora antalet användningsområden för systemet, måste användaren själv inhämta tillräckligt med kunskap för att rätt använda denna i sin speciella applikation.
BEIJER ELECTRONICS AB FRÅNTAGER SIG ALLT ANSVAR FÖR DIREKTA ELLER INDIREKTA SKADOR SOM UPPKOMMIT I SAMBAND MED INSTALLATION, ANVÄNDNING ELLER REPARATION AV DENNA UTRUSTNING.

**Synpunkter på manualen kan noteras i formuläret. Denna sida, eller en kopia av den, kan skickas till oss enligt nedanstående adresser.**

| Synpunkter på MA00509A | Sida |
|---|---|
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

Namn .....

Företag .....

Adress .....

Postadress .....

Land .....

Telefon .....Telefax.....

E-post .....

**Faxa till Marknadsavdelningen, Beijer Electronics AB 040-93 23 01 eller skicka med post till Marknadsavdelningen, Beijer Electronics AB, Box 426, 201 24 MALMÖ.**

**Det går också bra att skicka kommentarer till manual@beijer.se.**

# Innehåll

**1 Säkerhetsinformation ..... 1-1**

**2 Inledning ..... 2-1**

2.1 Programbearbetning hos PLC-system ..... 2-1
2.1.1 Avläsning via I/O-RAM ..... 2-2
2.1.2 Direkt I/O-avläsning ..... 2-3
2.1.3 Programcykeltid ..... 2-4
2.1.4 Tidsövervakning av programcykel (WDT -Watch Dog Timer) ..... 2-6
2.2 Programminnesarea/användarminne, MELSEC QnA(S) ..... 2-7
2.3 Programminnesarea/användarminne MELSEC Q ..... 2-10

**3 Beskrivning av operander ..... 3-1**

3.1 Ingångar X / Utgångar Y ..... 3-1
3.2 Minnesceller ..... 3-2
3.2.1 Minnesceller M och L ..... 3-2
3.2.2 Kommunikationsminnesceller B ..... 3-3
3.2.3 Felminnesceller F ..... 3-3
3.3 Tidskretsar T ..... 3-4
3.3.1 High speed-/low speed-tidskretsar ..... 3-5
3.3.2 Ackumulerande tidskrets ST ..... 3-7
3.4 Räknare ..... 3-8
3.4.1 Räknare C ..... 3-8
3.4.2 Interrupträknare C ..... 3-9
3.5 Register ..... 3-10
3.5.1 Dataregister D ..... 3-10
3.5.2 Kommunikationsregister W ..... 3-12
3.5.3 Filregister R ..... 3-13
3.5.4 Indexregister Z ..... 3-14
3.6 Master Control-markörer N ..... 3-15
3.7 Pekare ..... 3-17
3.7.1 Pekare P ..... 3-17
3.7.2 Interruptpekare I ..... 3-19
3.8 Konstanter ..... 3-25

**4 I/O-adressering och specialfunktioner ..... 4-1**

4.1 Regler för in-/ utgångsadressering ..... 4-1
4.1.1 Utbyggnadsmöjligheter ..... 4-3
4.2 Utbyte av in-/utgångsmodul under drift ..... 4-4
4.3 Förberedelser vid programmering ..... 4-6
4.3.1 Batterimatade minnesceller ..... 4-6
4.3.2 Meddelande vid fel ..... 4-6
4.3.3 Felmeddelande med felminne F ..... 4-7
4.3.4 STOP till RUN utgångsstatus ..... 4-7
4.3.5 Användning av interruptprogram ..... 4-7

4.4 Specialfunktioner ... 4-8
4.4.1 Behörighetskod ... 4-8
4.4.2 Realtidsklocka ... 4-9

4.5 Parametrar ... 4-12
4.5.1 Inställning av parametrar ... 4-12

**5 Grundläggande instruktionsbeskrivning ... 5-1**

5.1 Symbolförklaringar ... 5-1

5.2 Hantering av numeriska värden ... 5-3

5.3 Indexering ... 5-6

5.4 Felmeddelande ... 5-7

**6 Instruktionsöversikt ... 6-1**

6.1 Sekvensinstruktioner ... 6-2
6.1.1 Ingångsinstruktioner ... 6-2
6.1.2 Anslutningsinstruktioner ... 6-3
6.1.3 Utgångsinstruktioner ... 6-3
6.1.4 Skiftinstruktioner ... 6-4
6.1.5 Master control instruktioner ... 6-4
6.1.6 Programavslutande instruktioner ... 6-4
6.1.7 Diverse instruktioner ... 6-4

6.2 Tillämpade instruktioner del 1 ... 6-5
6.2.1 Jämförelseinstruktioner ... 6-5
6.2.2 Aritmetiska instruktioner ... 6-8
6.2.3 Omvandlingsinstruktioner ... 6-11
6.2.4 Överföringsinstruktioner ... 6-13
6.2.5 Förgreningsinstruktioner ... 6-14
6.2.6 Interruptinstruktioner ... 6-14
6.2.7 Uppdateringsinstruktioner ... 6-14
6.2.8 Andra behändiga instruktioner ... 6-15

6.3 Tillämpade instruktioner del 2 ... 6-17
6.3.1 Logiska instruktioner ... 6-17
6.3.2 Rotationsinstruktioner ... 6-19
6.3.3 Skiftinstruktioner ... 6-19
6.3.4 Bitinstruktioner ... 6-21
6.3.5 Datainstruktioner ... 6-21
6.3.6 Strukturerade programinstruktioner ... 6-24
6.3.7 Tabellinstruktioner ... 6-26
6.3.8 Buffertminnesinstruktioner ... 6-26
6.3.9 Displayinstruktioner ... 6-28
6.3.10 Debug- och feldiagnosinstruktioner ... 6-30
6.3.11 Teckensträngsinstruktioner ... 6-31
6.3.12 Specialfunktionsinstruktioner ... 6-37
6.3.13 Datakontrollinstruktioner ... 6-38
6.3.14 Växlingsinstruktioner för filregister ... 6-38
6.3.15 Klockinstruktioner ... 6-40
6.3.16 Yttre operandinstruktioner ... 6-40
6.3.17 Programinstruktioner ... 6-41
6.3.18 Andra instruktioner ... 6-42

6.4 Datalänkinstruktioner ..... 6-43
6.4.1 Nätverksinstruktioner ..... 6-43
6.4.2 Dedicerade datalänk instruktioner ..... 6-43
6.4.3 A-serien kompatibla datalänkinstruktioner ..... 6-44
6.4.4 Läs/Skriv routing information ..... 6-44

**7 Instruktionernas uppbyggnad ..... 7-1**

7.1 Instruktionsstruktur ..... 7-1
7.1.1 Källa (S) ..... 7-1
7.1.2 Destination (D) ..... 7-2
7.1.3 Antal (n) ..... 7-2

7.2 Instruktionernas skrivsätt ..... 7-3
7.2.1 Enkel- eller dubbelord och pulsfunktion ..... 7-3
7.2.2 MELSEC och IEC ..... 7-3
7.2.3 Ytterligare beskrivning av skrivsättet ..... 7-5
7.2.4 Specifikation av skrivsättet ..... 7-5

7.3 Programmering av variabler ..... 7-6

7.4 Datatyper ..... 7-8
7.4.1 Databehandling ..... 7-9
7.4.2 Flyttal ..... 7-14
7.4.3 Vektorer och register i GX IEC Developer ..... 7-16
7.4.4 Att använda strängar (STRING) ..... 7-18

7.5 Indexering ..... 7-19

7.6 Exekveringsfel ..... 7-22
7.6.1 Verifiering av adressområde ..... 7-22
7.6.2 Verifiering av operanddata ..... 7-24

7.7 Instruktioners exekveringsvillkor ..... 7-25
7.7.1 Exekveringsvillkor ..... 7-25
7.7.2 EN-ingången och ENO-utgången ..... 7-26

7.8 Antal programsteg ..... 7-27

**8 Instruktioner ..... 8-1**

8.1 Allmänna anvisningar ..... 8-1

8.2 Ingångsinstruktioner ..... 8-3
8.2.1 Ingångsvillkor, NO- och NC-kontakt, LD, AND, OR ..... 8-3
8.2.2 Ingångsvillkor, NO- och NC-kontakt, med pulsfunktion, LDP, ANDP, ORP ..... 8-5

8.3 Anslutningsinstruktioner ..... 8-6
8.3.1 Sammankoppling av villkorsblock, ANB, ORB ..... 8-6
8.3.2 Förgreningsinstruktioner, MPS, MPP, MRD ..... 8-8
8.3.3 Invers, INV ..... 8-11

8.4 Utgångsinstruktioner ..... 8-12
8.4.1 Aktivera utgångskontakt, OUT ..... 8-12
8.4.2 Aktivera tidskretskontakt, OUT T ..... 8-13
8.4.3 Aktivera räknarkontakt, OUT C ..... 8-15
8.4.4 Ett- och nollställning av operand, SET, RST ..... 8-17
8.4.5 Puls på positiv eller negativ flank, PLS, PLF ..... 8-19
8.4.6 T-vippa (Flip-Flop), FF ..... 8-21

8.5 Skiftinstruktioner ... 8-22
8.5.1 Skift av bitar, SFT ... 8-22

8.6 Master Control instruktioner ... 8-24
8.6.1 Aktivera och återställa överordnat villkor, MC, MCR ... 8-24

8.7 Programavslutande instruktioner ... 8-26
8.7.1 Programslut, FEND, END ... 8-26

8.8 Diverse instruktioner ... 8-28
8.8.1 Programavbrott, STOP ... 8-28
8.8.2 Tomt programsteg, NOP ... 8-29

8.9 Jämförelseinstruktioner ... 8-31
8.9.1 Jämförelse av villkor, 16 bitar, binärt ... 8-31
8.9.2 Jämförelse av villkor, 32 bitar, binärt ... 8-33
8.9.3 Jämförelse av villkor, flyttal ... 8-35
8.9.4 Jämförelse av villkor, teckensträngar ... 8-36

8.10 Aritmetiska instruktioner ... 8-38
8.10.1 Binär addition och subtraktion, +, - ... 8-38
8.10.2 Binär addition och subtraktion, D+, D- ... 8-40
8.10.3 Binär multiplikation och division, *, / ... 8-42
8.10.4 Binär multiplikation och division, D*, D/ ... 8-44
8.10.5 BCD-addition och BCD-subtraktion, 4 siffror, B+, B- ... 8-46
8.10.6 BCD-addition och BCD-subtraktion, 8 siffror, DB+, DB- ... 8-48
8.10.7 BCD-multiplikation och -division, 4 siffror, B*, B/ ... 8-50
8.10.8 BCD-multiplikation och -division, 8 siffror, DB*, DB/ ... 8-52
8.10.9 Addition och subtraktion av flyttal, E+, E- ... 8-54
8.10.10 Multiplikation och division av flyttal, E*, E/ ... 8-56
8.10.11 Addition av teckensträngar, $+ ... 8-58
8.10.12 Öka och minska operand med 1, INC, DEC ... 8-59
8.10.13 Öka och minska operand med 1, DINC, DDEC ... 8-61

8.11 Omvandlingsinstruktioner ... 8-63
8.11.1 Omvandling till BCD-format, BCD ... 8-63
8.11.2 Omvandla till binärformat, BIN ... 8-66
8.11.3 Omvandling till flyttalsformat, FLT ... 8-69
8.11.4 Omvandling till heltalsformat, INT ... 8-71
8.11.5 Omvandling till dubbelord, DBL ... 8-73
8.11.6 Omvandling till enkelord, WORD ... 8-74
8.11.7 Ändra tecken på heltal, NEG ... 8-75
8.11.8 Ändra tecken på flyttal, ENEG ... 8-77

8.12 Dataöverföringsinstruktioner ... 8-78
8.12.1 Överföring av binär data, MOV ... 8-78
8.12.2 Överföring av flyttalsdata, EMOV ... 8-80
8.12.3 Överföring av teckensträngar, $MOV ... 8-81
8.12.4 Invertering och överföring, CML ... 8-83
8.12.5 Blocköverföring, BMOV ... 8-86
8.12.6 Blocköverföring, FMOV ... 8-88
8.12.7 Växling av ord, XCH ... 8-90
8.12.8 Växlar hög och låg byte i enkelord, SWAP ... 8-92

8.13 Programgrensinstruktioner ... 8-93
8.13.1 Programhopp, CJ, SCJ, JMP, GOEND ... 8-93

8.14 Programkontrollinstruktioner ..... 8-96
8.14.1 Styrning av interruptprogram, EI, DI, IMASK ..... 8-96
8.14.2 Slut på interruptprogram, IRET ..... 8-99

8.15 Uppdateringsinstruktioner ..... 8-100
8.15.1 Uppdatering av in-/utgångar, RFS ..... 8-100

8.16 Logiska instruktioner ..... 8-101
8.16.1 Logisk AND, WAND ..... 8-101
8.16.2 Logisk OR, WOR ..... 8-103
8.16.3 Logisk exklusiv OR, WXOR ..... 8-105
8.16.4 Logisk exklusiv NOR, WXNR ..... 8-107

8.17 Rotationsinstruktioner ..... 8-109
8.17.1 Rotation åt höger, ROR, RCR ..... 8-109
8.17.2 Rotation åt vänster, ROL, RCL ..... 8-111
8.17.3 Rotation till höger, DROR, DRCR ..... 8-113
8.17.4 Rotation till vänster, DROL, DRCL ..... 8-115

8.18 Dataskiftinstruktioner ..... 8-117
8.18.1 Skift av enkelord, SFR, SFL ..... 8-117
8.18.2 Skift av n bitar, ett steg, BSFR, BSFL ..... 8-119
8.18.3 Skift av enkelord, ett steg, DSFR, DSFL ..... 8-121

8.19 Bitinstruktioner ..... 8-123
8.19.1 Påverkan av bit i enkelord, BSET, BRST ..... 8-123
8.19.2 Kontroll av bitstatus i ord, TEST ..... 8-125

8.20 Datainstruktioner ..... 8-127
8.20.1 Sökning av data, SER ..... 8-127
8.20.2 Summering av ettställda bitar, SUM ..... 8-129
8.20.3 Avkodning, DECO ..... 8-132
8.20.4 Kodning, ENCO ..... 8-133
8.20.5 Uppackning av data, DIS ..... 8-135
8.20.6 Nerpackning av data, UNI ..... 8-137

8.21 Strukturerade programinstruktioner ..... 8-139
8.21.1 Upprepande instruktioner, FOR, NEXT ..... 8-139
8.21.2 Avbrytning av upprepande instruktion, BREAK ..... 8-141
8.21.3 Anrop av subrutinprogram, CALL ..... 8-143
8.21.4 Slut på subrutinprogram, RET ..... 8-145
8.21.5 Nollställning av utgångar i subrutinprogram, FCALL ..... 8-147
8.21.6 Uppdatera nätverkskommunikation, COM ..... 8-148

8.22 Datatabellinstruktioner ..... 8-150
8.22.1 Skriv till FIFO-register, FIFW ..... 8-150
8.22.2 Läs från FIFO-register, FIFR ..... 8-151
8.22.3 Läs data från slutet av FIFO-register, FPOP ..... 8-153

8.23 Buffertminnesinstruktioner ..... 8-155
8.23.1 Läs från funktionsmodul, FROM ..... 8-155
8.23.2 Skriv till funktionsmodul, TO ..... 8-157

8.24 Teckensträninginstruktioner ..... 8-160
8.24.1 Omvandla till ASCII-format, ASC ..... 8-160
8.24.2 Omvandla till HEX-format, HEX ..... 8-162

8.25 Specialfunktioner ... 8-164
8.25.1 Sinusomvandling av flyttal, SIN ... 8-164
8.25.2 Cosinusomvandling av flyttal, COS ... 8-166
8.25.3 Tangensomvandling av flyttal, TAN ... 8-168
8.25.4 Arcussinusomvandling av flyttal, ASIN ... 8-170
8.25.5 Arcuscosinusomvandling av flyttal, ACOS ... 8-172
8.25.6 Arcustangensomvandling av flyttal, ATAN ... 8-174
8.25.7 Radianomvandling av flyttal, RAD ... 8-176
8.25.8 Gradomvandling av flyttal, DEG ... 8-177
8.25.9 Kvardatrot ur flyttal, SQR ... 8-178
8.25.10 Logaritmomvandling av flyttal, EXP ... 8-179
8.25.11 Logaritmomvandling av flyttal, LOG ... 8-181
8.25.12 Generering av slumptal, RND, SRND ... 8-183

8.26 Datakontrollinstruktioner ... 8-184
8.26.1 Begränsning av utgångsvärden, LIMIT ... 8-184
8.26.2 Dödbandskontroll, BAND ... 8-186
8.26.3 Zonkontroll, ZONE ... 8-188

8.27 Filregister, växlingsinstruktion ... 8-190
8.27.1 Val av filregisterblock, RSET ... 8-190

8.28 Andra instruktioner ... 8-192
8.28.1 Cykeltidsövervakning, WDT ... 8-192
8.28.2 Programmerbart pulståg, DUTY ... 8-193

**9 Felkoder ... 9-1**

9.1 Felkoder för MELSEC Q och QnA ... 9-1

**10 Specialminnesceller ... 10-1**

10.1 Tabell specialminnesceller ... 10-2

**11 Specialdataregister ... 11-1**

11.1 Tabell specialdataregister ... 11-2

**Sakregister ... I**

# 1 Säkerhetsinformation

Denna handbok vänder sig till elfackutbildade användare som har kännedom om säkerhetsstandarder och kunskap i automationsteknik. Projektering, installation, provning och besiktning av anläggningar med MELSEC QnA(S) och Q skall utföras av behörig personal.

## Allmänt

– Läs noga igenom säkerhetsförskrifterna.

– Kontrollera leveransen för att upptäcka eventuella transportskador. Meddela leverantören snarast om skador upptäcks.

– Använd inte systemet i en miljö där det finns hög risk för explosioner.

– Leverantören tar inte ansvar för modifierad, ändrad eller ombyggd utrustning.

– Endast reservdelar och tillbehör tillverkade enligt specifikation av leverantören får användas.

– Läs installations- och användarbeskrivningen noga innan systemet installeras, används eller repareras.

– Vätska får aldrig hällas i springor eller hål i systemet. Detta kan orsaka brand eller att utrustningen blir strömförande.

– Systemet ska hanteras av personer med adekvat utbildning.

## Vid installation

– Systemet är konstruerat för fasta installationer på en plan yta.

– Installera systemet enligt installationsbeskrivning.

– Jordning ska ske enligt installationsbeskrivning.

– Installation ska göras av personer med adekvat utbildning.

– Högspännings-, signal- och spänningskablar måste separeras.

– Fastställ att spänning och polaritet från kraftkällan är korrekt innan systemet spänningssätts.

– Öppningarna i höljet är avsedda för luftcirkulation och får inte övertäckas.

– Placera ej systemet där det finns risk att den utsätts för starka magnetfält.

– Kringutrustning måste vara lämplig där den används.

## Vid användning

– Håll systemet rent.

– Nödstoppsfunktioner eller andra säkerhetsfunktioner får ej styras från systemet.

## Service och underhåll

– Garanti gäller enligt avtal.

– Reparationer ska utföras av personer med adekvat utbildning.

## Vid nedmontering och skrotning

– Återvinning av terminalen eller delar av terminalen skall ske enligt gällande regler i respektive land.

– Beakta att följande komponenter innehåller ämnen som kan vara skadliga för hälsa och miljö: litiumbatteri, elektrolytkondensatorer samt display.

## Varningssymbol

Denna manual innehåller en varningssymbol med olika innehåll som skall uppmärksamma viktig information och syftar till att öka personsäkerheten och skydda utrustningen. När symbolen med dess innehåll upptäcks skall texten läsas och klargöras.

**Försiktighet!**
Uppmärksammar att mindre skada kan komma att orsakas på person eller egendom om inte nödvändiga försiktighetsåtgärder vidtas.

# 2 Inledning

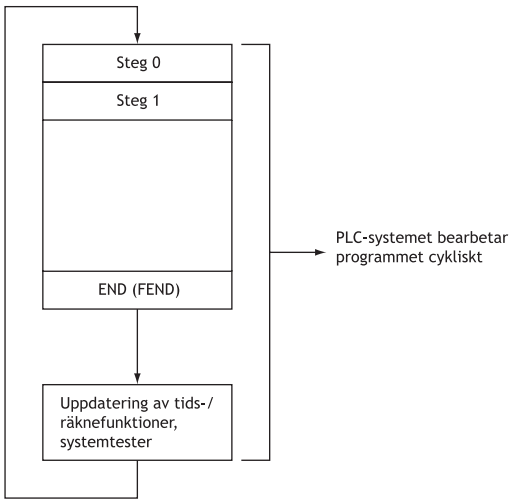
## 2.1 Programbearbetning hos PLC-system

### Grundprincipen för programbearbetning

PLC-programmet består av ett antal instruktioner lagrade sekventiellt i programminnet. Programbearbetningen sker instruktion för instruktion i den ordning de är lagrade med start från den första instruktionen på programrad 0. Denna bearbetning fortlöper ända tills någon slutinstruktion (FEND eller END) påträffas. Efter slutinstruktionen utförs vissa systemfunktioner som t ex uppdatering av in- och utgångsstatus samt felövervakning.

När dessa funktioner är avklarade startas programbearbetningen åter från programrad 0. Denna cykliska programbearbetning är karakteristisk för PLC-system.

Den tid som åtgår för att utföra en arbetscykel, dvs gå från programsteg 0 till att återkomma till steg 0, kallas cykeltid. Cykeltiden är givetvis beroende av PLC-programmets storlek eller rättare sagt hur många instruktioner som bearbetas under ett varv.

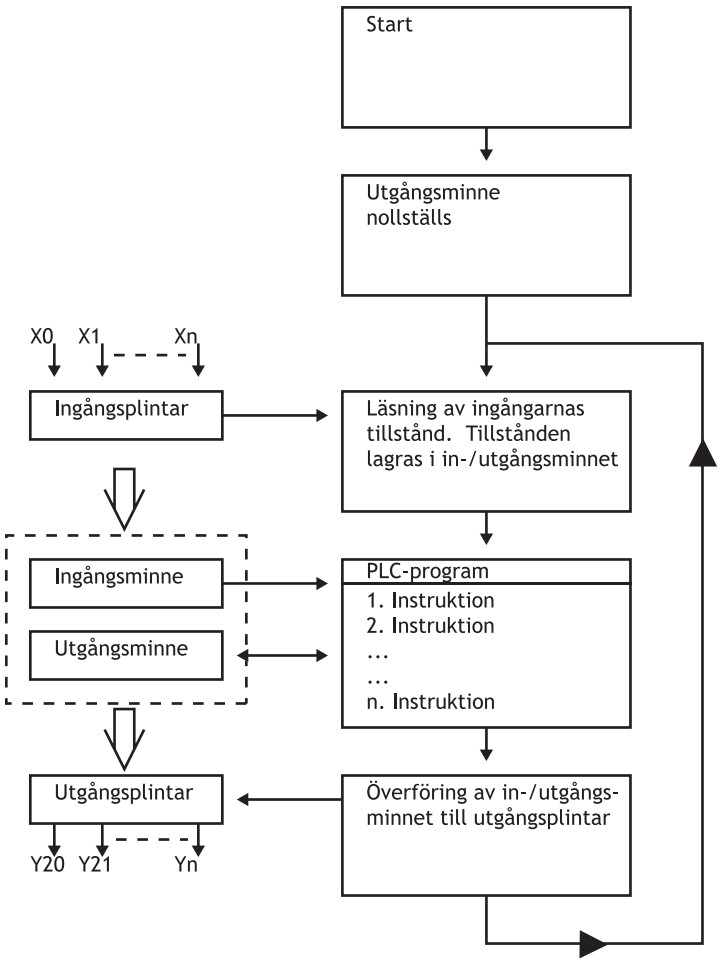


## 2.1.1 Avläsning via I/O-RAM

MELSEC Q och QnA(S) arbetar med I/O-kopiering. Refresh mode är ett annat namn för denna metod.

Hos ett PLC-system med I/O-kopiering består en komplett arbetscykel av tre delmoment, utförda i nämnd ordning:

1. Avläsning av samtliga ingångar
2. Programbearbetning
3. Överföring av beräknad utgångsstatus till samtliga utgångar




Nedan förklaras varje delmoment mer i detalj

### Avläsning av samtliga ingångar

I detta första moment läser centralenheten av status hos samtliga ingångar och lagrar status i ett speciellt minne, ett så kallat I/O-RAM.

### Programbearbetning

Här bearbetar centralenheten instruktionerna i den ordning de är lagrade. När en instruktion med en ingångsadress exekveras (t ex AND X10), avläses den minnescell som representerar denna ingång. Under hela programbearbetningsfasen arbetar centralenheten mot I/O-RAM.

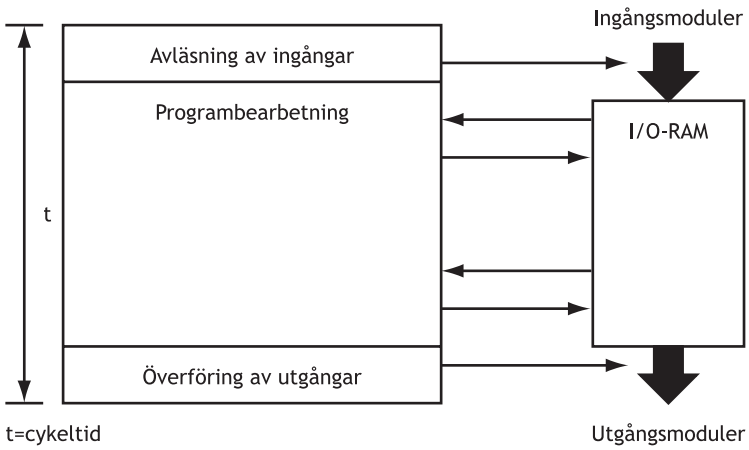
Ett resultat av detta är att ingångarnas status, som representeras av I/O-RAM, ligger stabilt under hela programbearbetningscykeln.

### Överföring av beräknad utgångsstatus till samtliga utgångar

Under programbearbetningen beräknas ny utgångsstatus. Dessa nya statusvärden påverkar inte direkt utgångarna utan lagras endast i I/O-RAM. Varje utgång har också en minnescell tilldelad i I/O-RAM. Först när en slutinstruktion (FEND eller END) utförts överförs de nya statusvärdena i I/O-RAM till utgångarna.

### Programcykel

De tre beskrivna delfunktionerna för en komplett arbetscykel utförs efter varandra. Tiden för genomförandet av en komplett cykel kallas cykeltid.




## 2.1.2 Direkt I/O-avläsning

För att uppdatera status på digitala in-och utgångar under pågående programcykel används instruktionen RFS eller så kallad direkt adressering. Med instruktionen RFS anger man första adressen som skall uppdateras och antalet adresser. Ett alternativ till att använda instruktionen RFS är att använda direkt adressering av in-och utgångar. Detta görs genom att lägga till D före adressen.

T ex DX9 istället för X9, eller DY20 istället för Y20. Med detta sätt att adressera in- och utgångar uppdateras status direkt.

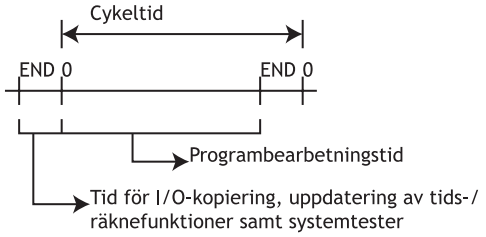
## 2.1.3 Programcykeltid

Cykeltiden är den tid PLC-systemet behöver för att bearbeta en komplett programcykel dvs den tid det tar att gå från programsteg 0 via I/O-kopieringen och åter till steg 0. Denna tid lagras i specialregister i enheter om 1 ms. Avläsning kan ske med en programmeringsenhet i monitormode.

– SD520: Aktuell cykeltid

– SD524: Kortast uppmätta cykeltid

– SD526: Längsta uppmätta cykeltid

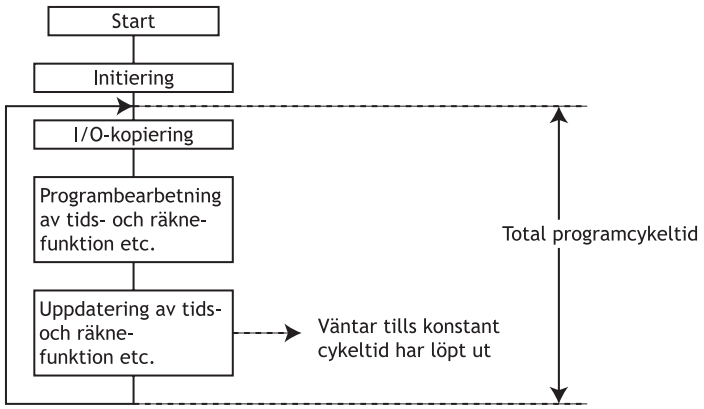



Cykeltiden beräknas enligt följande:

Mer data om cykeltider finns tillgänglig, *se kapitel 11*.

### Konstant programcykeltid

I vissa applikationer kan det vara en fördel med konstant cykeltid då den karakteristiska tidsvariationen mellan olika programcykler undviks helt. T ex kan programmet fås att bearbetas 10 gånger per sekund.

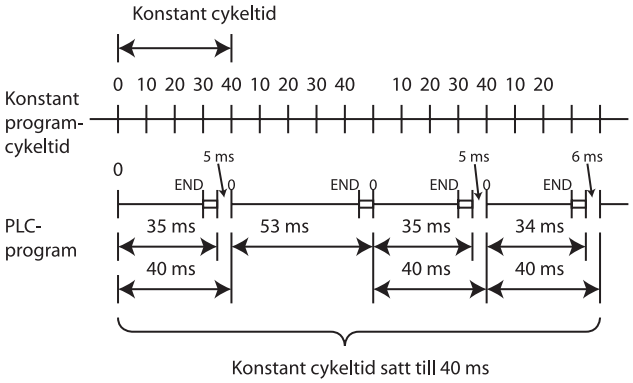


## Inställning av konstant programcykel

Den konstanta cykeltiden för QnA och QnA(S) kan programmeras i intervallet 5 - 2 000 ms i steg om 1 ms. För Q02-Q25H kan den konstanta cykeltiden programmeras i intervallet 0,5 - 2 000 ms i steg om 0,5 ms. För Q00(J) och Q01 kan den konstanta cykeltiden programmeras i intervallet 1 - 2 000 ms i steg om 1 ms.

Tiden för övervakningsfunktionen WDT bör vara större än den konstanta cykeltiden. WDT kan annars aktiveras.

Den konstanta cykeltiden måste sättas till ett värde som är större än maximala effektiva cykeltiden. Är inte detta uppfyllt erhålls ingen konstant cykeltid utan programmet bearbetas så fort som möjligt.




## Noggrannhet

Ett eller flera interrupt kan inträffa under den tid systemet väntar tills den konstanta cykeltiden har löpt ut. Den konstanta cykeltiden kan därför bli förlängd med interruptprogrammets exekveringstid. Inträffar mer än ett interrupt under väntetiden är det summan av interruptprogrammens exekveringstider med vilken den konstanta cykeltiden förlängs.

Konstanta cykeltiden kan också förlängas med längsta exekveringstiden för en instruktion i ett low-speed program (GX Developer) + low-speed programmets END-instruktion.

## 2.1.4 Tidsövervakning av programcykel (WDT - Watch Dog Timer)

Watch Dog Timer är en tidsfunktion som övervakar att den totala programcykeltiden inte överstiger ett inställt förvalsvärde.

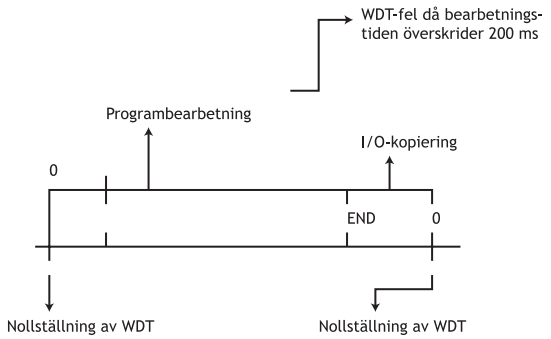
Förvalsvärdet är förinställt på 200 ms, men kan justeras med en programmeringsenhet.

Varje gång PLC-systemets centralenhet bearbetar en END- (FEND)-instruktion, eller instruktionen på programrad 0, återställs Watchdog-timern.

Skulle WDT överskrida förvalsvärdet erhålls ett felmeddelande och följande sker:

– Programbearbetningen avbryts och samtliga utgångar tvångsnollställs

– Lysdioden RUN på centralenheten släcks eller blinkar



# 2.2 Programminnesarea/användar-minne, MELSEC QnA(S)

I detta avsnitt beskrivs arbetsminnet och extraminne (PC-kort).

## Filer

All data som laddas ned till centralenheten lagras som filer, t ex

– Parametrar, *.QPA

– Program, *.QPG

– Filregister, *.QDR

## Programfiler

I GX IEC Developer skapas alltid bara en programfil som innehåller all PLC-programkod.

I GX Developer kan man skapa flera programfiler. Filerna kan vara av olika typ:

– Scan = exekveras varje cykel.

– Wait = exekveras endast vid anrop t ex som subrutin eller interruptrutin.

– Initial = exekveras första cykeln efter STOP-RUN.

– Low speed = exekveras under tid som ej används av scan-program vid konstant cykeltid.

## Inbyggt RAM-minne

Q och QnA(S) centralenheter har inbyggt RAM-minne för lagring av parametrar och program. Innehållet skyddas av ett backup-batteri. Innan RAM-minnet används skall det formateras. Formateringen utförs med GX IEC Developer eller GX Developer.

| Centralenhet | Antal programsteg |
|---|---|
| Q2A/Q2AS(H) | 28 k |
| Q2A/Q2AS(H) -S1 | 60 k |
| Q3A | 92 k |
| Q4A/Q4AR | 124 k |

## Extraminne - PC-kort

Centralenheterna QnA och QnAS kan kompletteras med extraminne i form av PC-kort med RAM-minne eller kombinerat RAM- och EEPROM-minne. RAM-minnet har ett eget backup-batteri. QnA-enheter har plats för två kort och QnAS-enheter har plats för ett kort.

PC-kort behövs t ex

– när filregister ska användas

– för lagring av program i EEPROM (ej beroende av batteribackup)

– när Sampling trace, Status latch eller Program trace ska användas (RAM)

– när symbolisk information (programstrukturen från GX IEC Developer) ska lagras

## Formatera minnet

Innan ett PC-kort kan användas måste det formateras:

– Sätt i kortet i centralenheten.

– Slå till switchen märkt Card Pull-out prohibited/permitted. Dioden i knappen ska tändas.

– Formatera det/de minnen som ska användas. Det görs via GX IEC Developer (välj meny Online. Format drive) eller GX Developer (välj meny Online. Format PLC memory).

– Minnet kan därefter användas.

## Filregister

Till filregister används vanligen RAM-minne. Det går även att använda EEPROM-minne men läsning och skrivning blir långsammare. Man måste även ta hänsyn till att livslängden på ett EEPROM är ca 10 000 skrivningar.

| Minne | Typ | Storlek | Antal register |
|---|---|---|---|
| Q1MEM-1MS | RAM | 1 MB | 500 k |
| Q1MEM-2MS | RAM | 2 MB | 1000 k |

## EEPROM

Val av PC-kort för lagring av program i EEPROM

| Minne | Typ | Storlek | Programsteg |
|---|---|---|---|
| Q1MEM-256SE | RAM+EEPROM | 128 kB RAM+128 kB ROM | 28 k |
| Q1MEM-512SE | RAM+EEPROM | 256 kB RAM+256 kB ROM | 60 k |
| Q1MEM-1MSE | RAM+EEPROM | 512 kB RAM+512 kB ROM | 124 k |

## Boot-operation

Om program och parametrar lagras på PC-kort ska Boot-operation konfigureras.

QnA-enheter:

| Switch nummer | Internt RAM | PC-kort, plats A | | PC-kort, plats B | |
|---|---|---|---|---|---|
| | | RAM | EEPROM | RAM | EEPROM |
| 1 | OFF | ON | ON | ON | ON |
| 2 | OFF | ON | OFF | ON | OFF |
| 3 | OFF | OFF | ON | ON | OFF |
| 4 | OFF | OFF | OFF | OFF | ON |

QnAS-enheter:

| Switch nummer | Internt RAM | PC-kort, plats A | |
|---|---|---|---|
| | | RAM | EEPROM |
| 2 | OFF | OFF | OFF |
| 3 | OFF | OFF | ON |
| 4 | OFF | ON | OFF |
| 5 | OFF | ON | ON |

Observera att ordningen på omkopplarna skiljer mellan QnA och QnAS.

## Drive-nummer

I GX IEC Developer ska man ange ett nummer för att specificera till vilken drive nedladdning ska ske.

| Minne | Drive | Minne | Drive |
|---|---|---|---|
| Internt RAM | 0 | PC RAM plats B | 3 |
| PC RAM plats A | 1 | PC ROM plats B | 4 |
| PC ROM plats A | 2 | | |

# 2.3 Programminnesarea/användarminne MELSEC Q

I detta avsnitt beskrivs arbetsminnet och extraminne (litet PC-kort).

## Filer

All data som laddas ned till centralenheten lagras som filer, t ex

– Parametrar, *.QPA

– Program, *.QPG

– Filregister, *.QDR

## Programfiler

I GX IEC Developer skapas bara en programfil som innehåller all PLC-programkod.

I GX Developer kan man skapa flera programfiler. Filerna kan vara av olika typ:

– Scan = exekveras varje cykel.

– Wait = exekveras endast vid anrop t ex som subrutin eller interruptrutin.

– Initial = exekveras första cykeln efter STOP-RUN.

– Low speed = exekveras under tid som ej används av scan-program vid konstant cykeltid.

– Fixed scan = exekveras med inställbart tidsintervall 0.5 ms till 60 s.

I Q00(J) och Q01 används bara arbetssätt *Scan.*

## Inbyggt programminne (RAM-typ)

MELSEC Q har inbyggt RAM-minne för lagring av parametrar och program.

Innehållet skyddas av ett backup-batteri. Innan programminnet används första gången ska det formateras. Formatering görs med GX IEC Developer eller GX Developer.

| Centralenhet | Antal programsteg |
|---|---|
| Q00J | 8 k |
| Q00 | 8 k |
| Q01 | 14 k |
| Q02(H) | 28 k[1] |
| Q06H | 60 k[1] |
| Q12H | 124 k[1] |
| Q25H | 252 k[1] |

[1]Max programstorlek beror på hur minnet används.
Annat som kan ta plats:
- parametrar, 1 k
- area för utökad onlineändring (MBOC), 0 - 4 k
- Versionsinformation GX IEC Developer, 1 k
- Parameterfil för funktionsmoduler (iparam.mpa), 1 k

## Inbyggt ROM-minne - Standard ROM

MELSEC Q centralenheter har även inbyggt flash-ROM med samma lagringskapacitet som RAM-minnet. Det kan användas för lagring av parametrar och program. ROM-minnet behöver inte formateras.

## Inbyggt RAM-minne - Standard RAM

MELSEC Q centralenheter har även ett inbyggt extra RAM-minne för filregister (ej Q00J). Innan RAM-minnet används första gången ska det formateras. Formatering görs med GX IEC Developer eller GX Developer.

| Centralenhet | kB |
|---|---|
| Q00J | – |
| Q00 | 64 |
| Q01 | 64 |
| Q02 | 64 |
| Q02H | 128 |
| Q06H | 128 |
| Q12H | 256 |
| Q25H | 256 |

OBS! Minnesstorleken är versionsberoende

## Utbyggt minne - litet PC-kort

MELSEC Q centralenheter (ej Q00(J) och Q01)kan kompletteras med ett extra minne i form av litet PC-kort med SRAM-minne, Flash-ROM eller ATA-minne. SRAM-minnet har ett eget backup-batteri.

PC-kort behövs t ex:

– när mer än 32k filregister ska användas

– för lagring av program som ska skickas till kund

– när Sampling trace ska användas (SRAM)

– när symbolisk information (programstrukturen från GX IEC Developer) ska lagras

## Formatera minne

PC-kort och interna minnen formateras med hjälp av GX IEC Developer eller GX Developer.

– Sätt i PC-kort (om det används) i centralenheten.

– Formatera det/de minnen som ska användas. Välj meny Online. Format drive i GX IEC Developer eller meny Online. Format PLC memory i GX Developer.

– Minnet kan därefter användas.

## Filregister

Till filregister kan drive 1, 2 (enbart med FlashROM) och 3 användas (se rubrik Drivenummer nedan). Upp till 1018k filregister kan skapas beroende på vilket minne som används.

Varje filregister förbrukar 2 byte minne.

## Boot-operation (Q02(H)-Q25H)

Bootoperation innebär att filer kopieras till programminnet vid uppstart.

Om program och parametrar lagras i interna flash-ROM minnet eller i PC-kort ska alltså bootoperation konfigureras. Två inställningar ska göras.

– I GX Developer anger man i PLC-parametrarna vilka filer som ska kopieras till programminnet och från vilket minne dom ska kopieras. I GX IEC Developer anger man till vilket minne projektet ska laddas ned, och därmed från vilket minne filerna ska kopieras.

– På den 5-poliga omkopplaren på centralenheten väljer man varifrån PLC-parametrarna ska läsas.

| Omkopplare | | Parametrar läses från |
|---|---|---|
| 2 | 3 | |
| OFF | OFF | Programminne (Drive 0) |
| ON | OFF | SRAM-kort (Drive 1) |
| OFF | ON | Flash-ROM eller ATA-kort (Drive 2) |
| ON | ON | Inbyggt flash-ROM (Drive 4) |

## Boot-operation (Q00(J), Q01)

Inställning av Bootoperation görs i PLC-parametrarna.

## Skriva program och parametrar till ROM

I GX Developer finns det två sätt att skriva till ett ROM-minne:

– Välj meny Online…Write to PLC(Flash ROM)…Write the program memory to ROM. Välj sedan vilket minne programmet ska skrivas till. De program och parametrar som finns i programminnet (internt RAM) kopieras till valt ROM-minne.

– Välj meny Online…Write to PLC (Flash ROM). Write to PLC(Flash ROM). Välj sedan vilka filer som ska skrivas och till vilket minne. Valda filer skrivs från GX Developer till valt ROM-minne.

I GX IEC Developer väljer man meny Online…Transfer setup…Projekt.

Välj vilket drivenummer projektet ska skrivas till.

## Drive-nummer

I GX IEC Developer anges varje minne med ett drivenummer.

| Minne | Drive |
|---|---|
| Programminne (internt RAM) | 0 |
| PC-kort, SRAM[1] | 1 |
| PC-kort, Flash-ROM eller ATA[1] | 2 |
| Standard RAM (internt) | 3 |
| Standard ROM (internt) | 4 |

[1]Ej Q00J, Q00, Q01

# 3 Beskrivning av operander

Ett PLC-program består av ett antal olika kombinationer av instruktioner lagrade efter varandra. En instruktion består av en funktion och en operand. Funktionen anger vad som ska ske (t ex OCH-, ELLER-funktion). Operanden bestämmer med vilken information funktionen ska genomföras (t ex ingång X, utgång Y).

Det förekommer också instruktioner som bara består av en funktion utan operand (t ex ANB-, ORB-instruktioner). Varje operand är försedd med en speciell adress (t ex X0, Y10, T15) och är därmed entydigt bestämd.

Antalet minnesceller, register, tidskretsar, räknare är ställbart via PLC-parametrarna.

## 3.1 Ingångar X / Utgångar Y

### Ingångar X

Med hjälp av in- och utgångar sker kommunikationen mellan PLC-systemet och den externa utrustningen. Över ingången X mottar styrsystemet signaler från externa givare eller kontakter. De förekommer både som normalt slutande eller normalt brytande kontakter.

Adresseringen av ingångarna X sker med hjälp av hexadecimala tal 0-9, A-F.

### Utgångar Y

Med utgångarna överför PLC-systemet styrsignaler till externa enheter som t ex kontaktorer, ventiler eller indikeringslampor. Utgångsstatus kan användas internt i PLC-programmet på samma sätt som en ingång (se nedan). Det finns inga begränsningar beträffande antalet gånger en
in-/utgång kan användas som villkor i ett program. Detta gäller såväl användning av slutande som brytande kontakt.

Adressering av utgångarna Y sker med hjälp av hexadecimala tal 0-9, A-F.

*status för utgångarna Y beror på RUN-kontaktens läge*

| RUN-kontakt | Utgångsstatus |
|---|---|
| STOP | Utgångarna är nollställda |
| RUN | Programresultatet styr utgångarna |
| PAUSE | Status från RUN-läge bevaras |
| STOP → RUN | Programförloppet börjar med utgångsstatus nollställd |
| PAUSE → RUN | Programbearbetningen fortsätts med utgångsstatus i PAUSE-läget |

# 3.2 Minnesceller

## 3.2.1 Minnesceller M och L

Minnesceller används vid mellanlagring av beräknade logiska värden vid den interna programbearbetningen.

### Minnesceller M

Antalet M-minnesceller kan ställas in i PLC-parametrarna.
Grundinställning är 8 192.

Status hos minnescellerna M nollställs antingen genom latch clear, nätspänningsavbrott eller genom att RESET-omkopplaren på centralenhetens front ställs i läge RESET.

### Minnesceller L

Antalet L-minnesceller kan ställas in i PLC-parametrarna.
Grundinställning är 8 192, för QnA(S) och Q (2 048, för Q00(J) och Q01).

Dessa minnesceller L kan konfigureras för att bibehålla sin information både under ett nätspänningsavbrott och då RESET-omkopplaren ställs i läge RESET. Nollställning av status sker då LATCH CLEAR görs på följande sätt:

**QnA(S)**

Nyckelbrytaren LATCH CLEAR vrids mellan läge STOP och LATCH CLEAR fyra gånger.

**Q02 och större**

Omkopplaren för Reset och LATCH CLEAR förs till LATCH CLEAR-läge 4 gånger. Operationen kan även utföras via GX (IEC) Developer.

**Q00(J) och Q01**

LATCH CLEAR utförs via GX (IEC) Developer

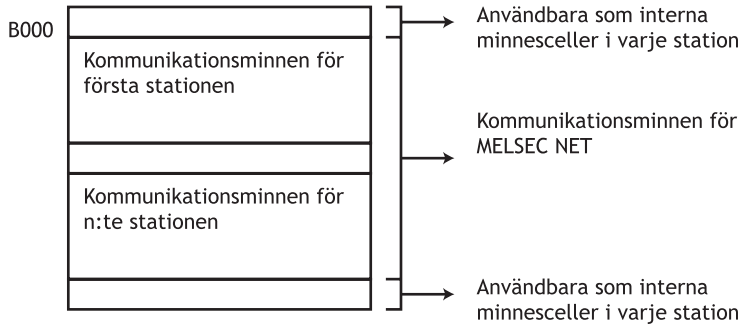
Minnescellerna är decimalt numrerade.

**Observera!**
Fördelningen av M- och L-minnesceller är oberoende av varandra. Båda områdena börjar adresseras från noll. (M0 resp L0)

## 3.2.2 Kommunikationsminnesceller B

Kommunikationsminnen B är interna minnesceller och används för kommunikation via nätverket MELSEC NET. Det finns upp till 8 192 B-minnesceller i MELSEC Q och QnA(S) (B0-B1FFF) (2 048 för Q00(J) och Q01).
Minnescellerna är hexadecimalt numrerade.

Kommunikationsminnen B, som inte används för kommunikation via MELSEC NET, kan användas som vanliga interna minnesceller.




Det finns ingen begränsning beträffande antalet gånger ett och samma kommunikationsminne kan användas i programmet som slutande eller brytande kontakt.

## 3.2.3 Felminnesceller F

Antalet felminnesceller kan ställas in i PLC-parametrarna.
Grundinställning är 2 048 (1 024 för Q00(J) och Q01).

Felminnescellerna F kan användas för att indikera feltillstånd hos ansluten maskin eller anläggning. F-cellerna aktiveras på normalt sätt i PLC-programmet.

När en felminnescell blir aktiverad sker följande

| CPU | Lysdiodsindikering |
|---|---|
| Q2A(-S1), Q | USER-lysdioden på CPU-fronten lyser<br>(Q00(J), Q01: ERR-lysdioden på CPU-fronten lyser) |
| Q3A, Q4A(R) | F-adressen visas med hjälp av teckendisplayen på CPU-fronten.<br>USER-lysdioden på CPU-fronten lyser |

F-adresser lagras i den tidsordning de aktiveras i specialregisterna SD64 - SD79 Dessa register kan lagra upp till 16 F-adresser samtidigt. Aktiveras en 17:e eller fler, registreras de ej i tidsordning.

Återställning av aktiverade F-adresser görs enligt följande

| CPU | Återställning av aktiverade F-adresser |
|---|---|
| Q2A(-S1), Q | Utförs i sekvensprogrammet med instruktionen:<br>RST F____ el LEDR<br>(Q00(J) och Q01: ej LEDR) |
| Q3AR, Q4A(R) | Utförs i sekvensprogrammet med instruktionen:<br>RST F____ el LEDR<br>Med hjälp av tryckknappen "INDICATOR RESET" på CPU:ns front |

# 3.3 Tidskretsar T

## Arbetssätt och noggrannhet hos tidskretsar

Tidskretsarna aktiveras i PLC-programmet med en OUT- eller OUTH-instruktion och deras ärvärde och utgångskontakt [TS_] uppdateras vid OUT- instruktionen.

När tidkretsen når sitt förinställda värde aktiveras utgångskontakten [TS_]. Då villkoret framför OUT-instruktionen faller, dvs då tidskretsen deaktiveras, återställs ärvärdet till 0 och kontakten [TS_] bryts.

Tidskretsens noggrannhet beror på programcykeltiden, eftersom ärvärde och utgångskontakt uppdateras en gång per arbetscykel.

## Onoggrannhet p g a att aktiveringen av tidkretsen sker osynkroniserat med klockpulserna

Aktiveras tidskretsen strax efter en klockpuls erhålls rätt tid. Aktiveras däremot tidskretsen strax innan en klockpuls räknas tidskretsens ärvärde upp med 1 direkt. Det innebär att en tidkrets med upplösningen 0,1 sekund ger ett tidssvar som är maximalt 0,1 sekund för kort medan för en tidskrets med upplösningen 0,01 sekund blir motsvarande fel 0,01 sekund.

## Onoggrannhet p g a placeringen av tidskretsens avkänningskontakt i PLC-programmet

Tidskrets med upplösningen 0,1 sekund: -0,1 sekund - 0 sekunder
Tidskrets med upplösningen 0,01 sekund: -0,01 sek - 0 sekunder

Ligger avkänningen efter OUT-instruktionen är tidsfelet 0 medan om den ligger före är felet + 1 cykelvarv.

Aktiveras tidskretsen genom en yttre signal ansluten till en ingång X tillkommer ett tidsfel på 0 - 1 cykeltid*. Observera att felet för de tidskretsar som är definierade som ackumulerande tidskretsar (håll-funktion) ökar med antalet gånger tiden fryses.

* Detta kan undvikas genom att adressera ingången DXnnn, vilket ger en direkt uppdatering av ingångens status.

## 3.3.1 High speed-/low speed-tidskretsar

Antalet tidskretsar kan ställas in i PLC-parametrarna. Grundinställning är 2 048 (512 för Q00(J) och Q01).

De tidskretsar T som står till förfogande är av fyra olika typer:
High speed- och low speed-tidskretsar samt ackumulerande high speed- och low speed-tidskretsar.

Tidbasen för high speed-tidskrets är ställbar inom området 1-100 ms (QnA(S)) och 0,1-100 ms (Q) (default = 10 ms).

Tidbasen för low speed-tidskrets är ställbar inom området 10-1 000 ms (QnA(S)) och 1-1 000 ms (Q) (default =100 ms).

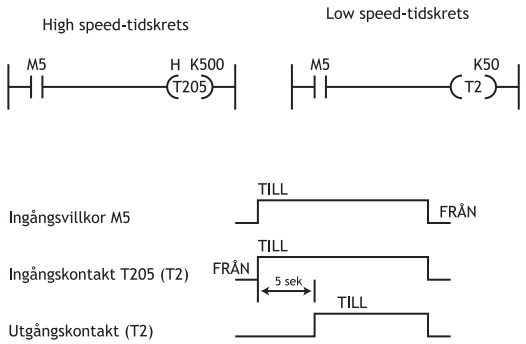
Instruktionen OUT startar tidskretsen som low speed och instruktionen OUTH startar tidskretsen som high speed oberoende av timeradress.

### Programexempel GX Developer

När ingångsvillkoret uppfylls (M5 i exemplet blir aktiverad) börjar tidskretsen räkna. När angiven tid har förflutit blir kontakten aktiverad (i exemplet T205 respektive T2).

Om ingångsvillkoret faller kommer det aktuella värdet hos tidskretsen att återgå till noll och tidskretsens kontakt deaktiveras (återgår till deaktiverat läge).

Tilldelningen av tidskretsens värden görs i form av decimaltal. Värden kan också anges med hjälp av dataregister.




| Tidsintervall 10 ms-tidskrets | Tidsintervall 100 ms-tidskrets |
|---|---|
| 1 till 32 767<br>(0.01 till 327.67 sek)* | 1 till 32 767<br>(0.1 till 3276.7 sek)* |

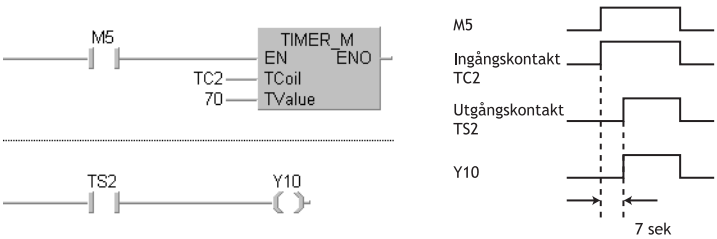
** Förutsatt att tidbaserna har grundinställning*

## Programexempel GX IEC Developer

När ingångsvillkoret uppfylls (M5 i exemplet blir aktiverad) börjar tidskretsen räkna. När angiven tid har förflutit blir kontakten aktiverad (i exemplet Y10, Y11).

Om ingångsvillkoret faller kommer det aktuella värdet hos tidskretsen att återgå till noll och tidskretsens kontakt deaktiveras (återgår till deaktiverat läge).

Tilldelningen av tidskretsens värden görs i form av decimaltal. Värden kan också anges med hjälp av dataregister.

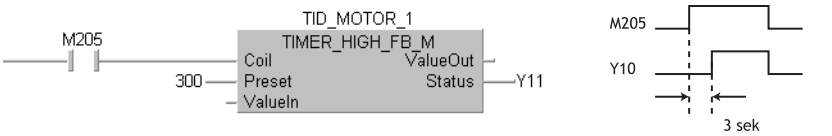



Funktionen TIMER_M ger en OUT-instruktion, vilket innebär att tidskretsen blir low speed. I GX IEC Developer har en timer tre adresser.

TC_= Ingångskontakt

TS_= Utgångskontakt

TN_= Aktuellt värde




Funktionsblocket TIMER_HIGH_FB_M ger en OUTH-instruktion, vilket innebär att tidskretsen blir high speed. Med denna typ av funktionsblock väljer GX IEC Developer själv en timer ur systemvariablerna.

# 3.3.2 Ackumulerande tidskrets ST

Antalet ackumulerande tidskretsar kan ställas in i PLC-parametrarna. Grundinställning är 0.

## Programexempel GX Developer

När ingångsvillkoret uppfylls (M5 i exemplet blir aktiverad) börjar tidskretsen att räkna upp till förinställd tid. När angiven tid har förflutit blir utgångskontakten aktiverad (i exemplet T254).

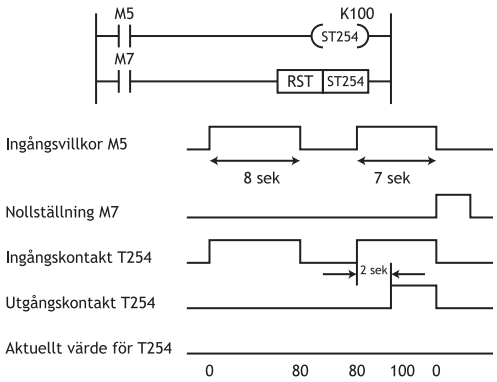
Om ingångsvillkoret faller kommer den ackumulerande tidskretsen att stanna på det aktuella tidvärdet.

Man kan endast återställa de ackumulerande tidskretsarna till noll med hjälp av RST-instruktionen.

Tilldelning av tidskretsens värden görs med decimaltal. Värden kan också ges med hjälp av dataregister.

| Tidsintervall |
|---|
| 1 till 32 767<br>(0.1 till 3276.7 sek) |

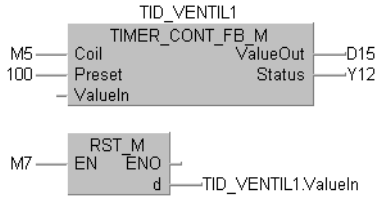
*Exempel*




## Programexempel GX IEC Developer

När ingångsvillkoret uppfylls (M5 i exemplet blir aktiverad) börjar tidskretsen räkna. När angiven tid har förflutit blir kontakten aktiverad (i exemplet TID_VENTIL1).

Om ingångsvillkoret faller kommer det aktuella värdet hos tidskretsen att bevaras. Återaktiveringen av ingångsvillkoret gör att tidskretsen fortsätter. Återställning av tidskretsen sker med RST-instruktionen. Tilldelningen av tidskretsens värden görs i form av decimaltal. Värden kan också anges med hjälp av dataregister.



# 3.4 Räknare

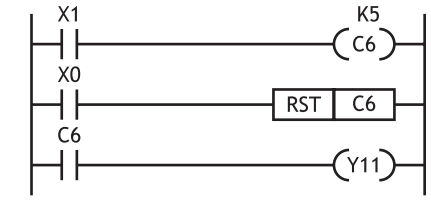
Antalet räknare kan ställas in i PLC-parametrarna. Grundinställning är 1 024 (512 för Q00(J) och Q01). Det finns två typer av räknare C, räknare för de normala PLC-programmen och interrupträknare för interruptprogrammen. Båda typerna utför endast räkning när ingångsvillkoret uppfylls (dvs på positiv flank).

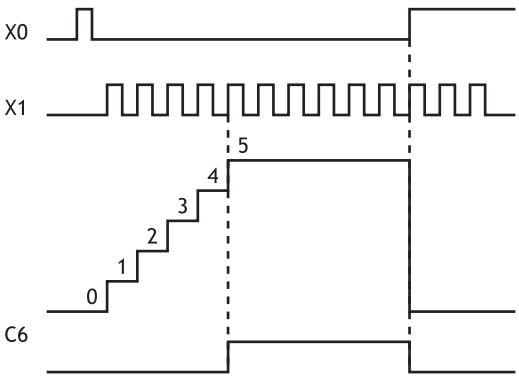
Räknarens förval anges med decimaltal mellan 1 och 32767. Även dataregister kan användas som förval. Räknarens aktuella status återställs med RST-instruktionen.

## 3.4.1 Räknare C

C är en normalt arbetande uppräknare för PLC-grundprogram och PLC-subrutinprogram. Den kan inte användas i PLC-interruptprogram.

### Programexempel GX Developer



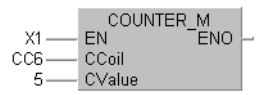

### Programexempel GX IEC Developer

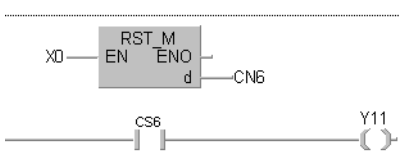
I GX IEC Developer har en räknare tre adresser.

CC_= Ingångskontakt

CS_= Utgångskontakt

CN_= Aktuellt värde



## 3.4.2 Interrupträknare C

Interrupträknare C används för att räkna antalet inträffade interrupt. Vilka räknare som skall användas anges i parameterinställningen (48 räknare går åt för interrupträknare).

**Observera!**
Interruptprogram kan inte användas om interrupträknare används.

### Maximal räknehastighet för räknare

Beräkningen av maxfrekvensen skiljer för räknare i grundprogrammet och räknare i ett interruptprogram.

Räknaren räknar upp 1 steg när villkoret framför OUT C går från ej uppfyllt tillstånd (positiv flankavkänning).

Den maximala räknehastigheten beror på cykeltiden. Pulstiden för räknevillkoret måste vara längre än cykeltiden för att erhålla en säker funktion.

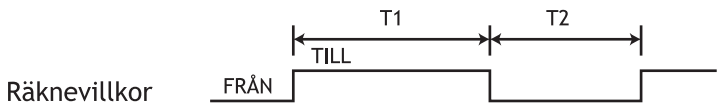
*Räknehastighet för räknare*

$C_{max} = n/100 \times 1/Ts$ (pulser/sekund)

Ts - Cykeltid (sek)
n - Förhållande mellan FRÅN och TILL hos räknevillkor (%) (se nedan)

$$n = \frac{T1}{T1 + T2} \times 100 \quad (\%)\ T1 \leq T2$$

$$n = \frac{T2}{T1 + T2} \times 100 \quad (\%)\ T1 > T2$$




### Maximal räknehastighet för interrupträknare

För att en räknare ska kunna användas i ett interruptprogram måste den definieras som interrupträknare. Detta görs i parameterinställningarna.

Den maximala räknefrekvensen är beroende av hur lång tid en instruktion i PLC programmet tar och antalet interrupträknare som är aktiva.

*Interrupträknarens maximala räknefrekvens beräknas enligt formeln*

$$C_{max} = \frac{1}{\text{längsta exekveringstid (S) för en instruktion eller END-instruktionen} + \text{0,0005 x antal interrupträknare}}$$

# 3.5 Register

## 3.5.1 Dataregister D

Q och QnA(S)-systemen har default 12 288 st dataregister (D0-D12287) (11 136 för Q00(J) och Q01). Dataregistret utgör ett minne på 16 bitar (enkelord) där mellanresultat efter beräkningar kan lagras. Man kan också hantera data på 32 bitar (dubbelord) om man adresserar två intilliggande dataregister. I GX Developer görs detta genom att bokstaven D läggs till datainstruktionens namn.

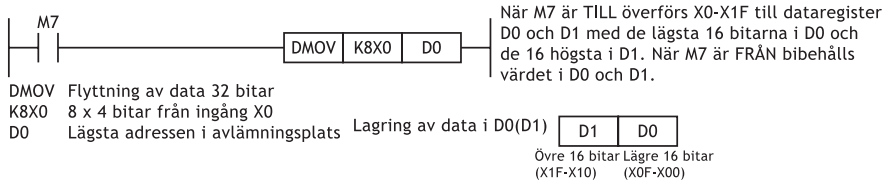
I GX IEC Developer deklareras register som INT (enkelord) eller som DINT (dubbelord) i globala eller lokala variabellistan.

### Programexempel GX Developer

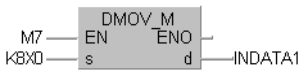
MOV flyttar 16 bitar medan DMOV flyttar 32 bitar.

Adresseringen av en dubbelordsoperand görs genom att ange det dataregister som har lägst adress.

*Exempel på en dubbelordsinstruktion*




### Programexempel GX IEC Developer




Variabeln INDATA1 deklareras t ex som global variabel av typen DINT (= double integer) med adressen D0.

Nollställning av dataregister:

– Vid nätspänningsavbrott*1

– Aktivering av RESET-omkopplaren till RESET-läge*1

– Vid LATCH CLEAR*2 operation

*1 Gäller inte om D har tilldelats batteribackup
*2 Gäller inte om D har tilldelats batteribackup med latch clear disable

Följande värden kan lagras i ett dataregister (enkelord):

– Decimala värden mellan -32 768 och 32 767

– Hexadecimala värden mellan 0 och FFFF

I dubbelord dvs i två seriekopplade register kan följande värden lagras:

– Decimala värden mellan -2 147 483 648 och 2 147 483 647

– Hexadecimala värden mellan 0 och FFFF FFFF

Följande operander kan också användas för lagring av data (om de inte använts på annat sätt):

– Tidskretsar T eller ST och räknare C

– Filregister R. Antalet R fastläggs i parametermode

– Kommunikationsregister W som inte används via nätverket MELSEC NET/ 10(H)

– Indexregister Z

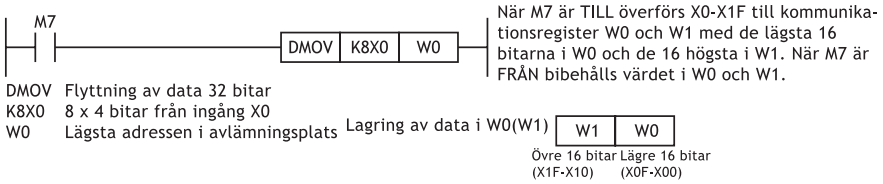
**Observera!**
Om ett dataregister ska behålla sin information vid spänningsbortfall måste det vara batterimatat. Detta väljs i parametermode.

## 3.5.2 Kommunikationsregister W

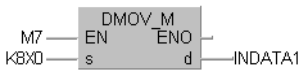
Default finns 8 192 st kommunikationsdataregister (W0-W1FFF) i Q och QnA(S)-systemen (2 048 för Q00(J) och Q01). Registren är hexadecimalt numrerade. Dessa dataregister används främst vid kommunikation via nätverket MELSEC NET.

Genom inställning i parametermode avgörs vilka kommunikationsregister som respektive system får skriva i. Om sådan kommunikation inte förekommer eller om alla kommunikationsregister inte används kan övriga register utnyttjas precis som vanliga dataregister D. Kommunikationsdataregisterna utgör ett minne om 16 bitar (enkelord) men genom att adressera två register fås ett ord om 32 bitar (dubbelord). Adressering av ett dubbelord görs genom att addera bokstaven D (D=dubbel) till den vanliga instruktionen samt ange det register som har den lägsta adressen.

### Programexempel GX Developer




### Programexempel GX IEC Developer




Variabeln INDATA1 deklareras t ex som global variabel av typen DINT (= double integer) med adressen W0.

Nollställning av kommunikationsregisterna:

– Vid nätspänningsavbrott[*1]

– Aktivering av RESET-omkopplaren till RESET-läge[*1]

– Vid LATCH CLEAR-operation[*2]

[*1]Gäller inte om W har tilldelats batteribackup

[*2] Gäller inte om W har tilldelats batteribackup med latch clear disabled

Följande värden kan lagras i ett kommunikationsregister (enkelord):

– Decimaltal mellan -32 768 och 32 767

– Hexadecimala tal mellan 0 och FFFF

I dubbelord dvs i två seriekopplade register kan följande värden lagras:

– Decimala värden mellan -2 147 483 648 och 2 147 483 647

– Hexadecimala värden mellan 0 och FFFF FFFF

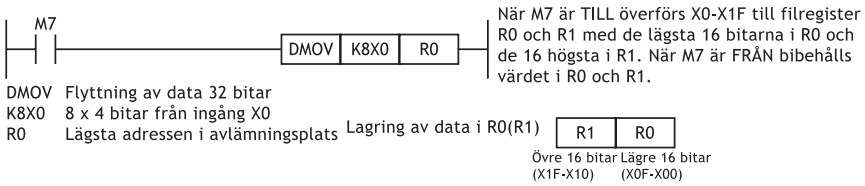
**Observera!**
Om ett kommunikationsregister ska bibehålla sin information vid spänningsbortfall måste detta anges i parametermode.

## 3.5.3 Filregister R

Filregisterna används som utökning av dataregisterna. Antalet register fastställs i parametermode. Filregisterna används på precis samma sätt som dataregisterna.

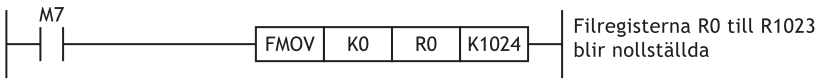
### Programexempel GX Developer

*Exempel på en dubbelordsinstruktion med R*

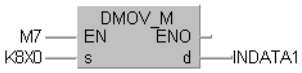



Filregisterna kan inte nollställas med centralenhetens omkopplare på samma sätt som D och W, utan man måste skriva in värdet 0. Ska många filregister nollställas bör man använda instruktionen FMOV.

*Exempel*

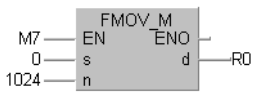



### Programexempel GX IEC Developer




Variabeln INDATA1 deklareras t ex som global variabel av typen DINT (= double integer) med adressen R0.

Filregisterna kan inte nollställas med centralenhetens omkopplare på samma sätt som D och W, utan man måste skriva in värdet 0. Ska många filregister nollställas bör man använda instruktionen FMOV.




Följande värden kan lagras i ett filregister (enkelord):

– Decimaltal mellan -32 768 och 32 767

– Hexadecimala tal mellan 0 och FFFF

I dubbelord dvs i två seriekopplade register kan följande värden lagras:

– Decimala värden mellan -2 147 483 648 och 2 147 483 647

– Hexadecimala värden mellan 0 och FFFF FFFF

**Observera!**
Antalet filregister fastställs i parametermode. Alla filregister är batterimatade PC-kort med RAM-minne måste installeras för att kunna använda filregister i Q, QnA(S).

## 3.5.4 Indexregister Z

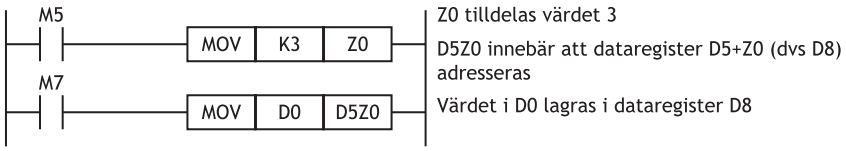
Det finns 16 st indexregister (Z0-Z15) i Q och QnA(S)-systemen (10 st för Q00(J) och Q01).

Indexregisterna kan användas vid t ex indirekt adressering med de olika operanderna: X, Y, M, B, L, F, T, C, D, W, R, K, H, P.

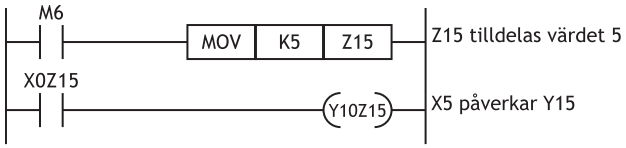
MELSEC Q och QnA(S) kan ej indexera konstanter K, H när det gäller dubbelords-instruktioner. Undantag: DROR, DRCR, DROL, DRCL (P)

### Programexempel GX Developer

*Indexering i datainstruktionen*




*Indexering av bitoperander*




Indexregisterna kan användas som vanliga dataregister.

Z består av 16 bitar men vid sammankoppling av två intilliggande adresser kan 32 bitar hanteras. Adressering sker då genom att ange t ex Z0 som lagrar de 16 lägsta bitarna och Z1 får de 16 högsta bitarna.

Nollställning av indexregisterna:

– Frånkoppling av nätspänning
– Aktivering av RESET-omkopplaren till RESET-läge
– Vid LATCH CLEAR operation

Följande värden kan lagras i indexregisterna (enkelord):

– Decimaltal mellan -32 768 och 32 767
– Hexadecimala tal mellan 0 och FFFF

I dubbelord dvs i två seriekopplade register kan följande värden lagras:

– Decimala värden mellan -2 147 483 648 och 2 147 483 647
– Hexadecimala värden mellan 0 och FFFF FFFF

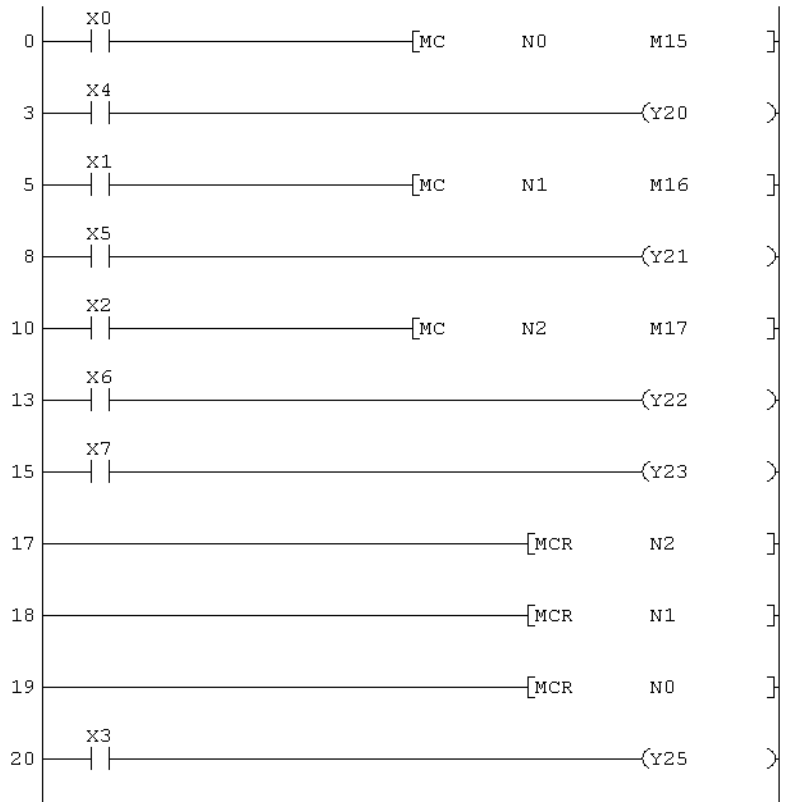
# 3.6 Master Control-markörer N

Master Control-funktionen innebär att ett programblock kan kopplas bort med hjälp av ett övergripande villkor. Blocket börjar med instruktionen MC med markören N och avslutas med instruktionen MCR med samma markör. Inuti ett MC-block kan ett nytt MC-block inrymmas. Markören N kan anta värden mellan 0 och 7 vilket innebär att upp till 8 nivåer är tillåtna.

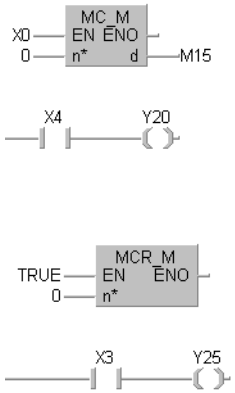
**Observera!**
Börja alltid med lägst nummer.

## Exempel GX Developer

*Nedanstående exempel visar hur N används.*



## Exempel GX IEC Developer




När Master Control-funktionen har kopplat bort ett programblock sker följande inom blocket:

– Tidkretsar: ärvärdet nollställs
– Ackumulerande tidskretsar: ärvärdet bibehålles
– Räknare: ärvärdet bibehålls
– Alla utgångar som styrs med OUT-instruktionen nollställs.

# 3.7 Pekare

## 3.7.1 Pekare P

Pekaren P används vid hopp som destinationsadress. P markerar till vilken plats i programmet hoppet ska ske. En och samma pekare kan inte användas mer än en gång inom ett program men däremot kan man hoppa till samma pekare från flera ställen i programmet. Det finns 4 096 pekare (P0-P4095) (300 för Q00(J) och Q01).

I GX Developer kan pekarna delas upp i lokala och gemensamma pekare. I PLC-parametrarna anger man vilken pekare som är den första gemensamma (gäller ej för Q00(J) och Q01).

Lokala pekare gäller enbart inom en programfil. Man kan då använda samma pekarnummer i flera programfiler. T ex P0-P99 i programfil A och P0-P49 i programfil B. För att bestämma var dom gemensamma pekarna ska börja summerar man antalet pekare i varje programfil.

Se exempel i tabellen nedan.

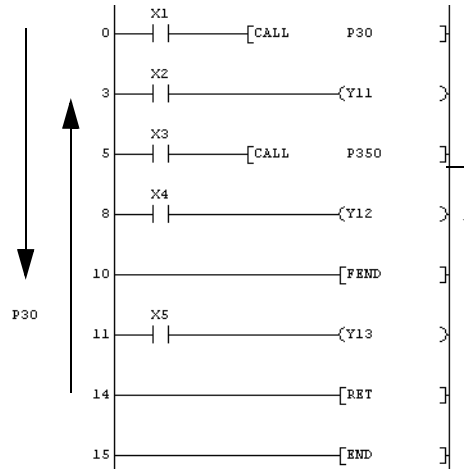
| Programfil | Antalet lokala pekare | Antalet använda |
|---|---|---|
| A | P20-P99 | 100 |
| B | P0-P49 | 50 |
| C | P100-P149 | 150 |
| | Summa | 300 |

Observera att även om dom första pekarna från P0 inte finns i programfil A, ska dom ändå räknas.

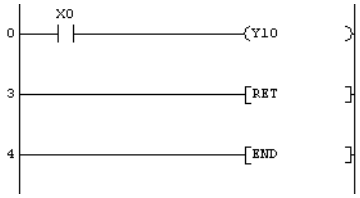
Eftersom summan av lokala pekare blev 300 kan den första gemensamma pekaren vara P300 eller högre. Till en gemensam pekare kan man hoppa från alla programfiler.

Nedanstående exempel visar hopp inom ett program och till ett annat program. P30 är en lokal pekare och P350 är en gemensam pekare. Program A är ett Scan-program och program D ett Wait-program.

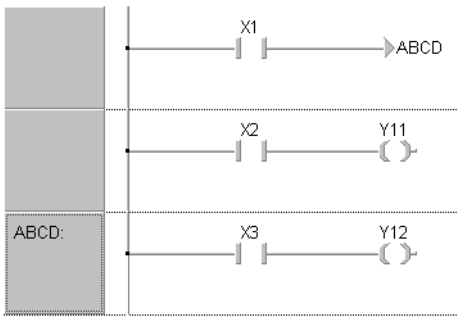
Program A




Program D




I GX IEC Developer anger man normalt ett “Label”-namn istället för att använda pekare vid hoppinstruktioner. GX IEC Developer väljer vid kompilering lediga pekare för hoppen.



## 3.7.2 Interruptpekare I

Interrupt betyder avbrott och används då en händelse måste behandlas direkt. Händelsen kan vara en yttre signal ansluten till funktionsmodulen QI61, AI61 eller A1SI61, en intern klockpuls, ett fel eller en signal från någon funktionsmodul.

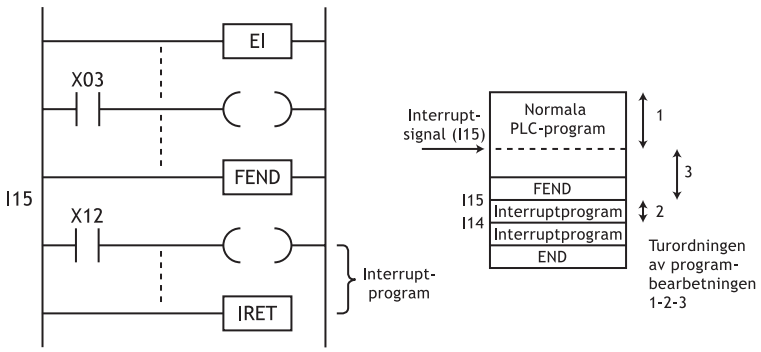
Interruptsignalen avbryter direkt bearbetningen av det normala PLC-programmet och startar ett interruptprogram. Pekarna I00-I47 anger var de olika interruptprogrammen börjar. När interruptprogrammet har bearbetats färdigt återupptas exekveringen av det normala PLC-programmet på det steg där det avbröts. Interruptprogrammet avslutas med instruktionen IRET. För att ett interruptprogram skall bearbetas måste instruktionerna EI och IMASK vara aktiverade.

### GX IEC Developer

När man gör ett interrupt program i GX IEC Developer skall man skapa en separat Task för varje interrupt. I taskattributen skriver man in en interruptpekare i fältet "Event". Till denna task knyter man det PLC-program som skall köras vid interrupt. GX IEC Developer lägger automatiskt till IRET och FEND vid kompilering. Instruktionerna EI (enable interrupt) och IMASK måste programmeras separat.

### GX Developer

Placeringen av interruptprogrammet i minnet måste vara efter instruktionen FEND men före END. Interrupt används för att få en snabb respons. Dessutom undviks den typiska variationen på responstiden (0 - 1 programcykel) som erhålls vid användning av ett PLC-system.



## Interruptpekare i QnA, QnAS

| Interruptpekare | Interruptfaktorer | | Prioritet |
|---|---|---|---|
| I0 | Interruptmodul AI61, A1SI61 | Ingång 1 | 29 |
| I1 | | Ingång 2 | 30 |
| I2 | | Ingång 3 | 31 |
| I3 | | Ingång 4 | 32 |
| I4 | | Ingång 5 | 33 |
| I5 | | Ingång 6 | 34 |
| I6 | | Ingång 7 | 35 |
| I7 | | Ingång 8 | 36 |
| I8 | | Ingång 9 | 37 |
| I9 | | Ingång 10 | 38 |
| I10 | | Ingång 11 | 39 |
| I11 | | Ingång 12 | 40 |
| I12 | | Ingång 13 | 41 |
| I13 | | Ingång 14 | 42 |
| I14 | | Ingång 15 | 43 |
| I15 | | Ingång 16 | 44 |
| I16 | Funktionsmodul med Interruptgenerator [1] | Modul 1 | 17 |
| I17 | | Modul 2 | 18 |
| I18 | | Modul 3 | 19 |
| I19 | | Modul 4 | 20 |
| I20 | | Modul 5 | 21 |
| I21 | | Modul 6 | 22 |
| I22 | | Modul 7 | 23 |
| I23 | | Modul 8 | 24 |
| I24 | | Modul 9 | 25 |
| I25 | | Modul 10 | 26 |
| I26 | | Modul 11 | 27 |
| I27 | | Modul 12 | 28 |
| I28 | Tidsinterrupt [2] | 100 ms | 48 |
| I29 | | 40 ms | 47 |
| I30 | | 20 ms | 46 |
| I31 | | 10 ms | 45 |

| Interruptpekare | Interruptfaktorer | | Prioritet |
|---|---|---|---|
| I32 | Felfaktorer [3] | Fel som stoppar programbearbetning | 1 |
| I33 | | Spänningsbortfall | 2 |
| I34 | | Kom.fel med I/O-modul. Trasig säkring. Fel i funktionsmodul. | 3 |
| I35 | | Exekveringsfel SFC programkörningsfel SFC exekveringsfel | 4 |
| I36 | | Fel i minneskort.<br>Fel i filsystemet | 5 |
| I37 | | Fel i utökad instruktion | 6 |
| I38 | | Konstant scantid överskriden | 7 |
| I39 | | CHK-instruktionen har utförts | 8 |
| I40 | Lediga | Lediga | 9 |
| I41 | | | 10 |
| I42 | | | 11 |
| I43 | | | 12 |
| I44 | | | 13 |
| I45 | | | 14 |
| I46 | | | 15 |
| I47 | | | 16 |

1) Första till tolfte pekare allokeras i ordning, med början på modulen närmast centralenheten.
2) De interna tiderna är grundinställningar. Tiderna kan ändras i steg om 5 ms i området 5 ms -1000 ms (ställs in i PLC-parametrarna).
3) När ett fel med interrupt "I32 (fel som stoppar programbearbetningen)" kommer programbearbetningen stanna först då programmet vid interruptpekare I32 har genomförts.

## Interruptpekare i Q00(J), Q01

| Interruptpekare | Interruptfaktorer | | Prioritet |
|---|---|---|---|
| I0 | Interruptmodul QI60 | Ingång 1 | 1 |
| I1 | | Ingång 2 | 2 |
| I2 | | Ingång 3 | 3 |
| I3 | | Ingång 4 | 4 |
| I4 | | Ingång 5 | 5 |
| I5 | | Ingång 6 | 6 |
| I6 | | Ingång 7 | 7 |
| I7 | | Ingång 8 | 8 |
| I8 | | Ingång 9 | 9 |
| I9 | | Ingång 10 | 10 |
| I10 | | Ingång 11 | 11 |
| I11 | | Ingång 12 | 12 |
| I12 | | Ingång 13 | 13 |
| I13 | | Ingång 14 | 14 |
| I14 | | Ingång 15 | 15 |
| I15 | | Ingång 16 | 16 |
| I16 - I27 | Används ej | — | — |
| I28 | Tidsinterrupt [1)] | 100 ms | 20 |
| I29 | | 40 ms | 19 |
| I30 | | 20 ms | 18 |
| I31 | | 10 ms | 17 |
| I32 - I127 | Används ej | — | — |

[1)] De interna tiderna är grundinställningar. Tiderna kan ändras i steg om 1 ms i området 2 ms -1000 ms (ställs in i PLC-parametrarna).

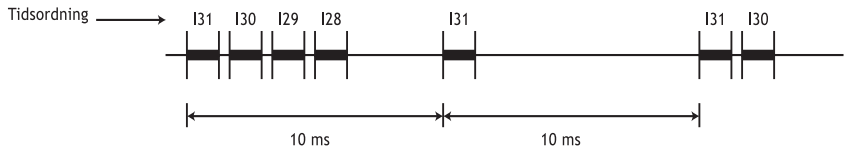
## Interruptpekare i Q02(H) - Q25H

| Interruptpekare | Interruptfaktorer | | Prioritet |
|---|---|---|---|
| I0 | Interruptmodul QI60 | Ingång 1 | 237 |
| I1 | | Ingång 2 | 238 |
| I2 | | Ingång 3 | 239 |
| I3 | | Ingång 4 | 240 |
| I4 | | Ingång 5 | 241 |
| I5 | | Ingång 6 | 242 |
| I6 | | Ingång 7 | 243 |
| I7 | | Ingång 8 | 244 |
| I8 | | Ingång 9 | 245 |
| I9 | | Ingång 10 | 246 |
| I10 | | Ingång 11 | 247 |
| I11 | | Ingång 12 | 248 |
| I12 | | Ingång 13 | 249 |
| I13 | | Ingång 14 | 250 |
| I14 | | Ingång 15 | 251 |
| I15 | | Ingång 16 | 252 |
| I16 | Funktionsmodul med interruptgenerator [1)] | Modul 1 | 224 |
| I17 | | Modul 2 | 225 |
| I18 | | Modul 3 | 226 |
| I19 | | Modul 4 | 227 |
| I20 | | Modul 5 | 228 |
| I21 | | Modul 6 | 229 |
| I22 | | Modul 7 | 230 |
| I23 | | Modul 8 | 231 |
| I24 | | Modul 9 | 232 |
| I25 | | Modul 10 | 234 |
| I26 | | Modul 11 | 235 |
| I27 | | Modul 12 | 236 |
| I28 | Tidsinterrupt [2)] | 100 ms | 256 |
| I29 | | 40 ms | 255 |
| I30 | | 20 ms | 254 |
| I31 | | 10 ms | 253 |

| Interruptpekare | Interruptfaktorer | | Prioritet |
|---|---|---|---|
| I32 | Felfaktorer 3) | Fel som stoppar program-bearbetning | 1 |
| I33 | | Ledig | - |
| I34 | | Kom.fel med I/O-modul. Trasig säkring. Fel i funktionsmodul. | 2 |
| I35 | | Exekveringsfel SFC programkörningsfel SFC exekveringsfel | 3 |
| I36 | | Fel i minneskort eller filsystemet | 4 |
| I37 | | Ledig | - |
| I38 | | Konstant scantid överskriden | 5 |
| I39 | | CHK-instruktionen har utförts | 6 |
| I40-I49 | Lediga | Lediga | |
| I50- I255 | Funktionsmoduler | Interrupt från funktionsmoduler med parametrerbar interruptfunktion t ex snabbräknare och kommunikationsmodul. | 18-223 |

1) Första till tolfte pekare allokeras i ordning, med början på modulen närmast centralenheten.
2) De interna tiderna är grundinställningar. Tiderna kan ändras i steg om 0,5 ms i området 0,5 ms -1000 ms (ställs in i PLC-parametrarna).
3) När ett fel med interrupt "I32 (fel som stoppar programbearbetningen)" kommer programbearbetningen stanna först då programmet vid interruptpekare I32 har genomförts.

När mer än ett tidsinterrupt (I28-I31) används kommer de att bearbetas i följande tidsordning:




**Observera!**
Start av interruptprogram kan blockeras med DI-instruktionen i grundprogrammet. Med EI-instruktionen upphävs denna blockering. Skulle en interruptsignal aktiveras under blockeringen, lagras denna händelse och interruptprogrammet startas direkt efter att EI-instruktionen har exekverats.

# 3.8 Konstanter

Vid programmering av MELSEC Q och QnA(S) kan både decimala konstanter K och hexadecimala konstanter H användas.

Decimala konstanter kan användas för följande:

– Förvalsvärden för tidskretsar T och räknare C, K1 - K32 767

– Antal grupper om 4 bitar då bitoperander (t ex X, Y, M, L) används i ordinstruktioner, K1 - K8

– Numeriska värden (konstanter) i ordinstruktioner

Enkelordsinstruktion, K-32 768 - K 32 767
Dubbelordsinstruktion, K-2 147 483 648 - K2 147 483 647

Hexadecimala konstanter kan användas för följande:

– Adressering av funktionsmoduler H0 - HFE0

– Numeriska värden (konstanter) i ordinstruktioner

Enkelordsinstruktion, H0 - HFFFF
Dubbelordsinstruktion, H0 - HFFFF FFFF

Konstanterna lagras i PLC-systemets minne i binär form.

I GX IEC Developer editorerna LD, FBD och IL skrivs konstanter in enligt följande:

– Decimala konstanter kan skrivas med eller utan K som prefix
Enkelordsinstruktion, -32 768 - 32 767
Dubbelordsinstruktion, -2 147 483 648 - 2 147 483 647

– Hexadecimala konstanter har prefixet 16#
Enkelordsinstruktion, 16#0 - 16#FFFF
Dubbelordsinstruktion, 16#0 - 16#FFFFFFFF

– Oktala konstanter har prefixet 8#
Enkelordsinstruktion, 8#0 - 8#17 7777
Dubbelordsinstruktion, 8#0 - 8#377 7777 7777

– Binära konstanter har prefixet 2#
Enkelordsinstruktion, 2#0 - 2#1111 1111 1111 1111
Dubbelordsinstruktion, 2#0 - 2#1111 1111 1111 1111 1111 1111 1111 1111

– Flyttalskonstanter
Positiva tal, 1.1754 E -38 - 3.4028 E 38
Negativa tal, -1.1754 E -38 - (-3.4028 E 38)

– ASCII sträng
T ex “ABCDE”

# 4 I/O-adressering och specialfunktioner

## 4.1 Regler för in-/ utgångsadressering

### Vid in-/utgångsadressering gäller följande för QnA(S)

– Grund- och expansionsplattorna har 5 eller 8 modulplatser. För QnA(S)-systemen finns även grundplattor med 3 modulplatser och expansionsplattor med 2 modulplatser.

– Expansionsplattor numreras (1-7 eller 1-3) med hjälp av byglar vid den vänstra expansionskontakten.

– Adresseringen börjar på grundplattan med modulplatsen längst till vänster (närmast CPU).

– Adresseringen av expansionsplattan sker på samma sätt som grundplattan, (från vänster till höger) med början på expansionsplattan med lägst nummer.

– En bottenplatta med färre platser än 8, anses innehålla 8 kortplatser.

– En tom modulplats, en plats som inte finns eller AG60 respektive A1SG60, belägger 16 adresser.

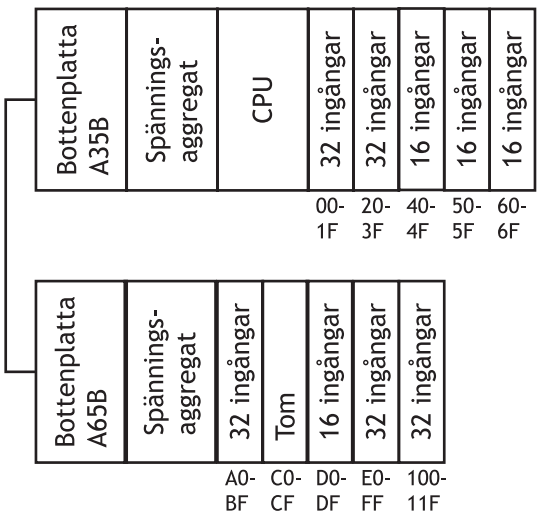
– När tommodulen AG62 eller A1SG62 används kan antalet belagda kanaler ställas in med en omkopplare.

– Vid parameterinställning kan 16, 32, 48 eller 64 adresser reserveras för tomma platser.

– Om en modul med 32 adresser monteras på en plats som är parametrerad som 16, fungerar inte de sista 16 kanalerna på modulen. Samtidigt fås ett felmeddelande.

– I exemplet nedan används en A35 grundplatta. Gör CPU-enheten själv in-/utgångsadresseringen kommer modulplatserna 5, 6 och 7 (vilka inte finns på A35B) att belägga vardera 16 kanaler (70-7F, 80-8F, och 90-9F).

Genom parameterinställning med en programmeringsenhet kan alla adresser utnyttjas.



## Vid in-/utgångsadressering gäller följande för Q

– Grund- och expansionsplattorna har 5, 8 eller 12 modulplatser.
Q00J har alltid grundplatta med 5 modulplatser.

– Expansionsplattor numreras (1-7 ) med hjälp av byglar vid den vänstra expansionskontakten.

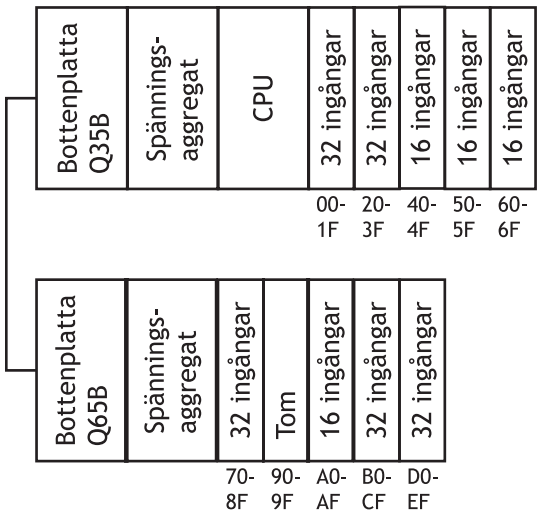
– Adresseringen börjar på grundplattan med modulplatsen längst till vänster (närmast CPU).

– Adresseringen av expansionsplattan sker på samma sätt som grundplattan, (från vänster till höger) med början på expansionsplattan med lägst nummer.

– En tom modulplats, QG60 belägger 16 adresser.

– Vid parameterinställning kan 16, 32, 48 eller 64 adresser reserveras för tomma platser.

– Om en modul med 32 adresser monteras på en plats som är tilldelad för 16, fungerar inte de sista 16 kanalerna på modulen. Samtidigt fås ett felmeddelande..



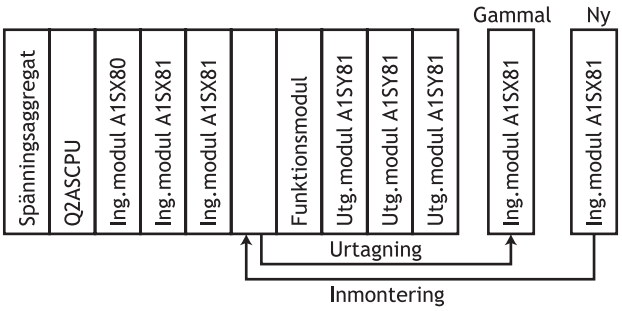
## 4.1.1 Utbyggnadsmöjligheter

| Centralenhet | Antal expansionsplattor | Antal I/O-moduler | Antal I/O i rack | Antal Ethernetmoduler | AntalMELSECNET 10(H) | Antal CC-Limk-moduler | Antal C24-moduler |
|---|---|---|---|---|---|---|---|
| Q2AS(H) | 3 | 32 | 512 | 4 | 4 | 8 | 6 |
| Q2AS(H)-S1 | 3 | 32 | 1 024 | 4 | 4 | 8 | 6 |
| Q2A | 3 | 32 | 512 | 4 | 4 | 8 | 6 |
| Q2A-S1 | 7 | 64 | 1 024 | 4 | 4 | 8 | 6 |
| Q3A | 7 | 64 | 2 048 | 4 | 4 | 8 | 6 |
| Q4A | 7 | 64 | 4 096 | 4 | 4 | 8 | 6 |
| Q4AR[1] | 7 | 64 | 4 096 | 4 | 4 | 8 | 6 |
| Q00J | 2 | 16 | 256 | 1 | 1 | 2 | —[2] |
| Q00, Q01 | 4 | 24 | 1 024 | 1 | 1 | 2 | —[2] |
| Q02(H), Q06H, Q12H, Q25H | 7 | 64 | 4 096 | 4 | 4 | 4 | —[2] |

[1] Max 63 I/O-moduler vid redundant spänningsmatning.
Max 56 I/O-moduler vid redundant spänningsmatning och redundant CPU.
[2] Max antal I/O-moduler begränsar antalet C24-moduler i MELSEC Q

# 4.2 Utbyte av in-/utgångsmodul under drift

Denna funktion gör det möjligt att byta ut en in- eller utgångsmodul i MELSEC QnA(S) under tiden PLC-programmet bearbetas. I MELSEC Q kan byte av moduler under drift endast göras med centralenheterna Q12PH och Q25PH.



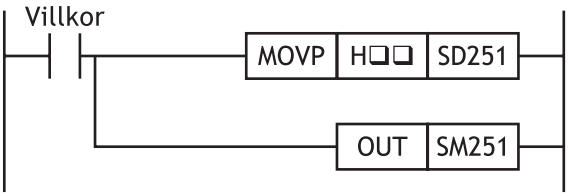

Utbytet kan utföras utan att centralenheten indikerar ett fel.

Utbytet kan ske på följande två sätt:

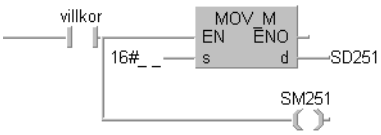
### Med hjälp av PLC-programmet

Aktivera nedanstående programavsnitt

### GX Developer




### GX IEC Developer




H❑❑ anger de två mest signifikanta siffrorna i den aktuella modulens startadress. (I exemplet ovan blir det talet H06)

Utbytesanvisning

– Aktivera ingångsvillkoret så att modulens två mest signifikanta siffror i startadressen hamnar i SD251 och att SM251 ettställs
– Byt ut modulen
– Återställ ingångsvillkoret så att SM251 nollställs

---

**Observera!**
Om en funktionsmodul ska bytas måste även FROM- och TO-instruktionerna avbrytas

---

## Med hjälp av en programmeringsenhet

Anslut programmeringsenheten till centralenheten och utför följande

– Skriv in de två mest signifikanta siffrorna i modulens startadress i specialdataregistret SD251 i Entry Data Monitor och tvångsställ sedan SM251 till 1.

*Exempel*

Modulens startadress = H070. Indata till SD251 = H07

– Tvångsställ SD251 till H7

– Ettställ SM251 i Entry Data Monitor

– Byt ut modulen

– Nollställ SM251 i Entry Data Monitor

## Funktion vid utbyte av in-/utgångsmodul

Följande inträffar med de olika modulerna i ett system beroende på status hos specialminnet SM251.

| SM251 | Modul | Funktion då CPU är i RUN-mode |
|---|---|---|
| Ettställd | CPU | 1. Lagrar aktuell status hos utpekad moduls ingångar eller utgångar i CPU interna minne när första END-instruktionen exekveras efter att SM251 har blivit ettställd. Därefter adresseras inte den utpekade modulen<br>2. Efter att SM251 har blivit ettställd, kommer status för de aktuella ingångarna att avläsas i det interna minnet<br>3. Efter att SM251 har blivit ettställd, kommer nya beräknade status för de aktuella utgångarna att lagras i det interna minnet<br>4. Efter att SM251 har blivit ettställd, kontrolleras inte samtliga in-/utgångsmoduler vad gäller korrekt typ eller utlöst säkring |
| | Övriga in-/utgångsmoduler | 1. Adresserar samtliga moduler utom den utpekade på normalt sätt. Den utpekade adresseras inte |
| Nollställd | CPU | 1. Efter att SM251 har blivit nollställd och END-instruktionen har exekverats, avläses status från ingångsmodulen<br>2. Efter att SM251 har blivit nollställd och END-instruktionen har exekverats, överförs status i det interna minnet till utgångsmodulen<br>3. Efter att SM251 har blivit nollställd, kontrolleras åter samtliga in-/utgångsmoduler vad gäller korrekt typ eller utlöst säkring |
| | Övriga in-/utgångsmoduler | 1. Samtliga moduler adresseras normalt |

# 4.3 Förberedelser vid programmering

Detta kapitel beskriver en del funktioner som bör uppmärksammas i samband med programinskrivning.

## 4.3.1 Batterimatade minnesceller

I MELSEC Q och QnA(S) finns default 8192 batterimatade minnesceller L0-L8191 (antalet kan ändras i parametermode). Dessa minnesceller behåller informationen vid ett nätspänningsavbrott och även när RESET-omkopplaren på CPU-enheten ställs i RESET-läge. Alla övriga operander nollställs (vissa undantag finns vad gäller special-minne och specialregister). Om tidskretsar, räknare, dataregister etc ska ha batteri-matning väljs detta i parametermode med en programmeringsenhet.

## 4.3.2 Meddelande vid fel

Inträffar ett fel i programbearbetningen eller i en I/O-modul aktiveras speciella min-nesceller och dataregister i enlighet med nedanstående tabell.

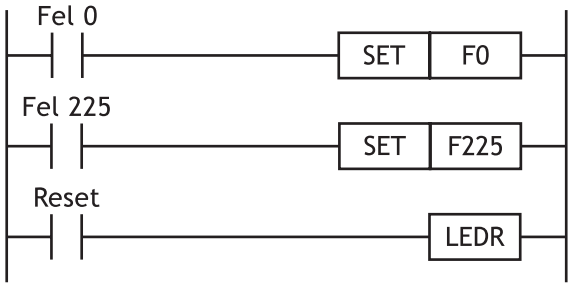
Vid fel påverkas CPU-enhetens RUN-lysdiod på olika sätt vilket också kan utläsas i tabellen nedan.

| FEL | | CPU-status | | | | | |
|---|---|---|---|---|---|---|---|
| | | Program-bearbet-ning | RUN LED | Special-minne som ett-ställs | Data i spe-cial-regis-ter | Själv-diag-nostik-fel (SD0) | ERROR LED |
| Beräk-ningsfel | T ex har en BCD-omvandling utförts som överstiger 9999 | Stannar* | Släcks | SM56 | - | 4100 | Blinkar |
| Fel typ av I/O-modul | Felaktig typ av I/O-modul detekterad | Stannar* | Släcks | SM61 | SD61 | 2000 | Blinkar |
| Säk-ringsfel | Säkring utlöst på utgångsmodul (kortslutnings-säkra kort) | Stannar* | Släcks | SM60 | SD60 | 1300 | Blinkar |
| Fel i funk-tions-modul | FROM/TO-instruktion har utförts på adress där funk-tionsmodul ej finns | Stannar* | Släcks | - | - | 2110 | Blinkar |

* Vid något av ovanstående fel kan man välja om CPU-enheten ska stanna eller fort-sätta. Detta val görs i parameterinställningen.

## 4.3.3 Felmeddelande med felminne F

På Q och QnA(S)-systemen tänds USER-lysdioden när ett felminne F aktiveras. För Q3A och Q4AR-systemen gäller att teckendisplayen på fronten visar numret på aktiverad felminnescell.




*Programexempel för att hantera felminne.*

**Observera!**
Q00(J) och Q01 kan inte hantera LEDR-instruktioner. Använd RST istället.

## 4.3.4 STOP till RUN utgångsstatus

När RUN-omkopplaren på centralenheten ändras från RUN till STOP eller om ingången för fjärrstyrning av RUN ändras från 0 till 1, nollställs alla utgångar. När sedan RUN-omkopplaren respektive fjärrkontakten återställs så att centralenheten hamnar i RUN-läge får alla utgångar samma status som innan centralenheten stoppades.

**Observera!**
Om man ej önskar funktionen kan detta ändras i parametermode.

## 4.3.5 Användning av interruptprogram

Följande punkter bör observeras i samband med användning av interruptprogram.

1. Ett interrupt kan inträffa under bearbetning av en instruktion. T ex kan en dubbelordsinstruktion (DMOV) avbrytas när endast förflyttningen av de första 16 bitarna skett. Efter avslutat interruptprogram fortsätter bearbetningen av instruktionen på vanligt vis.
2. Operander (Y, M, L, etc) som blivit påverkade i ett interruptprogram behåller sitt värde då bearbetningen av huvudprogrammet startar igen.
3. När interruptprogrammet bearbetas aktiveras DI (Disable Interrupt) automatiskt. Använd därför inte instruktionen EI i interruptprogrammet.

# 4.4 Specialfunktioner

PLC-systemen i MELSEC Q och QnA(S)-serierna innehåller många specialfunktioner vilka kan vara till stor nytta vid idrifttagning och felsökning.

## 4.4.1 Behörighetskod

För att skydda all information i centralenheten finns möjlighet att lägga in en kod. Innan några ändringar med någon programmeringsenhet kan utföras måste koden slås in. Om koden förkommer kan den inte läsas ut utan hela minnet måste raderas.

MELSEC QnA(S):
Behörighetskoden ska innehålla sex numeriska tecken

MELSEC Q:
Behörighetskoden ska innehålla fyra alfanumeriska tecken. När GX Developer används kan varje fil tilldelas en egen behörighetskod
Endast filer som lagras i RAM-minne kan tilldelas lösenord.

## 4.4.2 Realtidsklocka

Realtidsklockan är en standardfunktion i alla Q och QnA(S) centralenheter. Med den kan styrfunktioner sättas vid ett förutbestämt datum och tidpunkt. Vid spänningsbortfall reservmatas klockan från samma batteri som matar programminnet.

Realtidsklockan anger år, månad, dag, timme, minut, sekund och veckodag enligt följande:

År: Anger de 2 minst signifika siffrorna i QnA(S)-system. I MELSEC Q anges det med fyra siffror.
Skottår: Automatisk kompensering
Tid: 24-timmars angivelse
Klockan kan avläsas och tvångsställas med hjälp av specialminnen och specialdataregister. Klockans tidsnoggrannhet beror på omgivningstemperaturen.

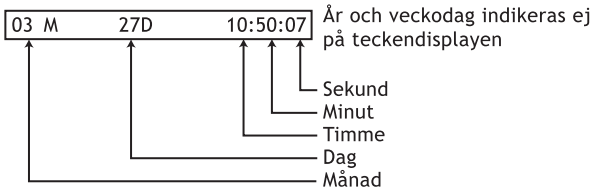
QnA(S)-system

| Omgivningstemp (˚C) | Avvikelse per dygn (sek) |
|---|---|
| 0˚ | -2,3 till +4,4 |
| 25˚ | -1,1 till +4,4 |
| 55˚ | -9,6 till +2,4 |

MELSEC Q

| Omgivningstemp (˚C) | Avvikelse per dygn (sek) |
|---|---|
| 0˚ | -3,18 till +5,25 |
| 25˚ | -3,93 till +5,25 |
| 55˚ | -14,69 till +3,53 |

Hos centralenheten Q3A och Q4A(R) kan klockan avläsas på teckendisplayen genom att aktivera specialminnet SM212. En felindikering har högre prioritet och kommer att skrivas över tidsangivelsen. År och veckodag indikeras ej på teckendisplayen.

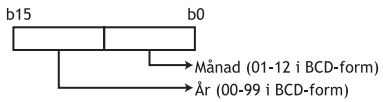
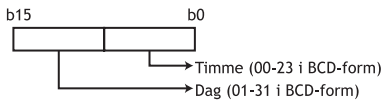
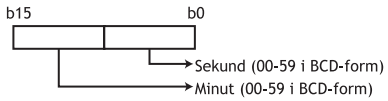
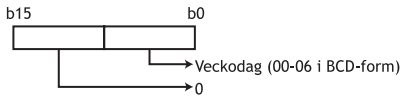


## Avläsning och tvångsställning av realtidklockan

Specialminne SM

| Kanal-nummer | Funktion | Förklaring |
|---|---|---|
| SM210 | Tvångsställning av klockan | Tvångsställer klockan till det värde som lagrats i specialdataregistren SD210-SD213. Tvångsställningen sker efter END-instruktionen efter att specialminnet SM210 har ettställts |
| SM211 | Fel på indata vid tvångsställning | Ettställs då indata i specialdataregistren SD210-SD213 inte är i BCD-form |
| SM212 | Utläsning på centralenhetens teckendisplay | Skriver ut datum och tid på Q3A och Q4A(R) teckendisplay då specialminnet SM212 ettställs |
| SM213 | Avläsning av klocka | Klockan avläses och aktuellt datum och tid lagras i specialdataregister SD210-SD213. Avläsningen sker efter END-instruktionen efter att specialminnet SM213 har ettställts |

Specialdataregister SD

| Dataregister | Funktion | Förklaring |
|---|---|---|
| SD210 | Klockdata (år, månad)[1)] |  |
| SD211 | Klockdata (dag, timme) |  |
| SD212 | Klockdata (minut, sekund) |  |
| SD213 | Klockdata (veckodag)[2)] |  <br>Veckodag / Data<br>Söndag 0<br>Måndag 1<br>Tisdag 2<br>Onsdag 3<br>Torsdag 4<br>Fredag 5<br>Lördag 6 |

Tvångsställning av klockan:

– Lagra det nya datumet och tiden i specialdataregistren SD210-SD213 i BCD-form

– Ettställ specialminnet SM210

Avläsning av klockan:

– Ettställ specialminnet SM213

– Datumet och tiden kan avläsas i specialdataregistren SD210 - SD213

**Observera!**
Realtidsklockan måste ställas minst en gång om funktionen ska utnyttjas.
Samtliga specialfunktioner SD210-SD213 måste innehålla relevant information vid tvångsställning, även om endast delar av klockan ska användas.
Klockan fungerar ej korrekt om felaktig data anges vid tvångsställningen.
Vid spänningsbortfall slutar klockan att gå om även anslutningen av batteriet bryts upp.
Det finns även PLC-instruktioner för att skriva till (DATEWR) respektive läsa av (DATERD) realtidsklockan.

1. De två lägsta siffrorna i året skrivs här i Q och QnA(S).
2. De två högsta siffrorna i året skrivs här i MELSEC Q.
   Alltid 0 i QnA(S)

# 4.5 Parametrar

I parameterarean kan funktioner och kapacitet hos centralenheterna i MELSEC Q och QnA(S) anpassas optimalt för de olika typer av applikationer.

Typiska parametrar är t ex:

– Antal register, minnesceller, timers osv.

– Hur många benämningar som ska kunna lagras i centralenheten

– Vilka operander som ska vara batterimatade

Parametrarna lagras, liksom PLC-programmet, i centralenhetens minne. Inställning av parametrar görs med programmeringsenhet.

## 4.5.1 Inställning av parametrar

I detta avsnitt beskrivs de oftast förekommande parameterinställningarna. Ändringar av parametrar kan göras exempelvis med GX Developer.

### Filregister i QnA(S)

Filregisterna lagras i PC-kort RAM-minne. Antalet filregister kan väljas i steg om 1k register (2k bytes). Minneskortets storlek begränsar antalet filregister.

### Filregister i Q

Filregister lagras i internt RAM-minne (standard RAM) eller i ett litet PC-kort med RAM-minne (memory card RAM). Antalet filregister kan väljas i steg om 1k register (2 k bytes). Minneskortets storlek begränsar antalet filregister.

### Inställning av operander med batteribackup

Inställning av de operander som ska bibehålla sin information under ett spänningsavbrott eller då RESET-omkopplaren sätts i läge RESET. Däremot nollställs informationen hos dessa operander om LATCH CLEAR-operation utförs.

### Inställning av operander för nätverket MELSEC NET

Här anges de operander som ska användas för kommunikation mellan de olika systemen via nätverket MELSEC NET.

In-/utgångar (X/Y):

Adressområden för masterstationen, lokala stationerna och decentraliserade I/O-enheterna

Kommunikationsminnesceller B:

Adressområden för kommunikationsminnesceller för masterstationen och de lokala stationerna

Kommunikationsdataregister W:

Adressområden för kommunikationsdataregister för masterstationen och de lokala stationerna

### Inställning av tidskretsarna

Tidbasen för tidskretsar är inställningsbar. Grundinställningen är 100 ms för low-speed-timer och 10 ms för highspeed timer. Alla tidskretsar kan vara low speed- eller high speed-tidskretsar. Instruktionen avgör vilket det blir. OUT-instruktionen ger low speed-tidskrets och OUTH-instruktionen ger en high speed-tidskrets. I GX IEC Developer finns även olika funktionsblock för de olika tidskretsarna.

### Inställning av I/O-adresser

Här kan I/O-adresserna reserveras för varje modulplats på grund- och expansionsplattorna. Antalet I/O-adresser per modulplats kan anges i steg om 16. Om ingen I/O-inställning görs så numrerar PLC-systemet automatiskt efter den hårdvara som finns.

### Externa (fjärr) RUN/PAUSE-signaler

Här kan valfria ingångar X tilldelas funktionen RUN respektive PAUSE. Det går inte att enbart utnyttja den externa REMOTE PAUSE-funktionen. Med dessa ingångar kan centralenheten ställas i RUN/STOP/PAUSE-tillstånd.

### Funktion vid fel

Här kan anges om centralenheten ska fortsätta programbearbetningen eller stoppa då ett fel uppträder i PLC-systemet.

### Val av felindikering (endast för Q3A och Q4A(R))

Om felminnescellerna F har försetts med benämningar, kan dessa erhållas på teckendisplayen på CPU-enhetens front, då F-cellerna aktiveras med en OUT- eller SET-instruktion.

### Utgångarnas status vid STOP - RUN

Här bestäms hur utgångarna ska påverkas då PLC-systemet går från STOP till RUN.

Prior to calc/Previous state: Utgångarnas aktuella status vid STOP-tillfället överförs till utgångarna igen.

After one scan/Recalculate: Utgångarna nollställs och påverkas först efter exekveringen av ett programvarv.

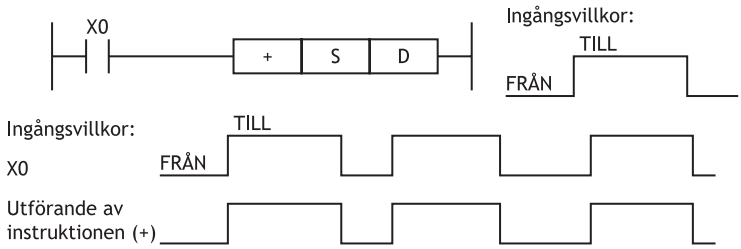
# 5 Grundläggande instruktionsbeskrivning

Innan alla de olika instruktionerna beskrivs i detalj, skall gemensamma grundläggande begrepp om instruktionshandhavandet förklaras.

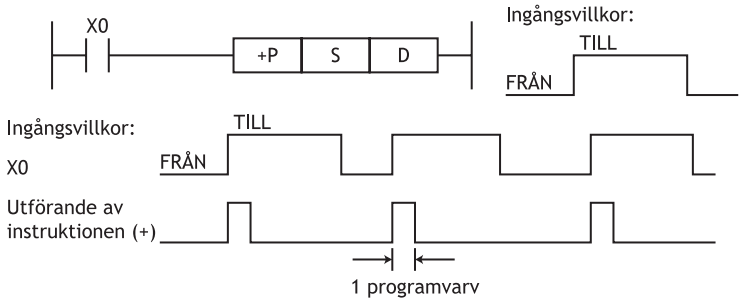
## 5.1 Symbolförklaringar

Här beskrivs de olika symboler som anger hur instruktionerna bearbetas och storleken på tillhörande operander. Beskrivningen gäller symbolerna P, D, S, S1, S2, och D som sätts in i kombination med de egentliga instruktionerna.

Först beskrivs de olika möjligheter som föreligger för att instruktionen ska utföras, dvs hur ingångsvillkoret behandlas. Det normala är att instruktionen utförs då ingångsvillkoret är uppfyllt, dvs en gång per programvarv så länge villkoret är uppfyllt.



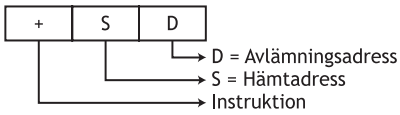

Den andra möjligheten är att instruktionen utförs endast vid ingångsvillkorets positiva flank dvs endast ett programvarv. Till instruktionen adderas då ett P.




Vid hantering av dubbelordsoperander placeras ett D framför instruktionen.

| | Möjliga operander | Belagda adresser |
|---|---|---|
| 16 bitar | X, Y, M, L, F, B | Max 16 adresser (16 bitar) |
| | T, C, D, W, R, A, Z | 1 adress (16 bitar) |
| 32 bitar | X, Y, M, L, F, B | Max 32 adresser (32 bitar) |
| | T, C, D, W, R, A, Z | 2 adresser (2*16 bitar) |

## Instruktion med två operander

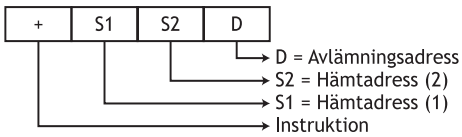



Enkelordsaddition (D) + (S) → (D)

Dubbeordsaddition (D+1, D) + (S+1, S) → (D+1, D)

## Instruktion med tre operander

I följande exempel adderas (S1) med (S2) och summan placeras i (D)




Enkelordsaddition (S1) + (S2) → (D)

Dubbelordsaddition (S1+1, S1) + (S2+1, S2) → (D+1, D)

## Exempel på en instruktions olika kombinationsmöjligheter

| | | |
|---|---|---|
| Enkelord | + S D | + S1 S2 D |
| | +P S D | +P S1 S2 D |
| Dubbelord | D+ S D | D+ S1 S2 D |
| | D+P S D | D+P S1 S2 D |

**Observera!**
Tänk på att de flesta data- och applikationsinstruktionerna upptar mer än ett programsteg.

# 5.2 Hantering av numeriska värden

I MELSEC Q och QnA(S) finns instruktioner för hantering av både 16- och 32-bitarsord. Den mest signifikanta biten i enkel- eller dubbelordet används som teckenbit.

Det numeriska området blir som följer:

16 bitar: -32 768 till 32 767
32 bitar: -2 147 483 648 till 2 147 483 647

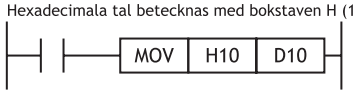
## Användning av konstanter

Decimaltal betecknas med bokstaven K

Det decimala talet 10 lagras i dataregistret D10 i binär form

Det decimala talet -10 lagras i dataregistret D10 i binär form

Hexadecimala tal betecknas med bokstaven H (16#)

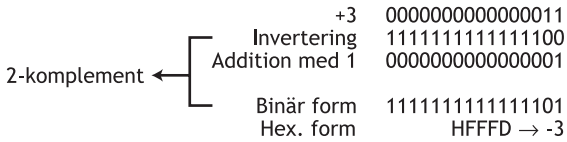



Det hexadecimala talet 10 (motsvarande det decimala talet 16) lagras i dataregistret D10 i binär form

## Det hexadecimala talet HFFFE ska divideras med 2

Var försiktig vid användningen av hexadecimala tal tillsammans med beräkningsinstruktioner. Nedan visas hur slutresultatet kan variera beroende på om det hexadecimala talet betraktas som ett enkel- eller dubbelord.

Först visas hur talet -3 repesenteras i ett enkelord i (2-komplement).




I nedanstående exempel divideras talet HFFFE med 2. I ena fallet med en enkelordsinstruktion och i det andra fallet med en dubbelordsinstruktion.

Enkelordsinstruktion

HFFFE som enkelord motsvarar det decimala talet -2. Division med 2 ger slutresultatet -1, vilket motsvarar HFFFF. Dvs i dataregistret D = lagras värdet HFFFF

Dubbelordsinstruktion

HFFFE som dubbelord motsvarar det decimala talet 65534. Division med 2 ger slutresultatet -32767 vilket motsvarar H7FFF. Dvs i dataregistret D = lagras värdet H7FFF

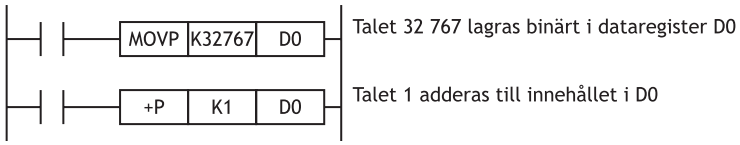
## Numeriska värden vid bearbetning med datainstruktioner

Bearbetning av numeriska data med enkel- och dubbelordsinstruktioner måste göras inom tillåtna talområden. Var uppmärksam med omslagen från positiva till negativa tal och tvärtom.

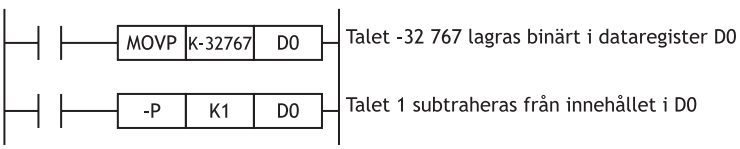
| | Enkelordsbearbetning | | Dubbelordsbearbetning | |
|---|---|---|---|---|
| | decimala tal | hexadecimala tal | decimala tal | hexadecimala tal |
| OMSLAG | 32 764 | 7FFC | 2 147 483 644 | 7FFF FFFC |
| | 32 765 | 7FFD | 2 147 483 645 | 7FFF FFFD |
| | 32 766 | 7FFE | 2 147 483 646 | 7FFF FFFE |
| | 32 767 | 7FFF | 2 147 483 647 | 7FFF FFFF |
| | -32 768 | 8000 | -2 147 483 648 | 8000 0000 |
| | -32 767 | 8001 | -2 147 483 647 | 8000 0001 |
| | -32 766 | 8002 | -2 147 483 646 | 8000 0002 |
| | -32 765 | 8003 | -2 147 483 645 | 8000 0003 |
| Omslag | - | - | - | - |
| | - | - | - | - |
| | - | - | - | - |

| | Enkelordsbearbetning | | Dubbelordsbearbetning | |
|---|---|---|---|---|
| | decimala tal | hexadecimala tal | decimala tal | hexadecimala tal |
| OMSLAG | -32 764 | 8004 | -2 147 483 644 | 8000 0004 |
| | -32 765 | 8003 | -2 147 483 645 | 8000 0003 |
| | -32 766 | 8002 | -2 147 483 646 | 8000 0002 |
| | -32 767 | 8001 | -2 147 483 647 | 8000 0001 |
| | -32 768 | 8000 | -2 147 483 648 | 8000 0000 |
| | 32 767 | 7FFF | 2 147 483 647 | 7FFF FFFF |
| | 32 766 | 7FFE | 2 147 483 646 | 7FFF FFFE |
| | 32 765 | 7FFD | 2 147 483 645 | 7FFF FFFD |
| | - | - | - | - |
| Neg. | - | - | - | - |
| Omslag | - | - | - | - |

**Observera!**
Varken Carry- eller Fel-flaggorna påverkas vid ovan nämnda omslag.




*Exempel på omslag (overflow)*




*Exempel på negativt omslag (underflow)*

## Bestämning av antal siffror

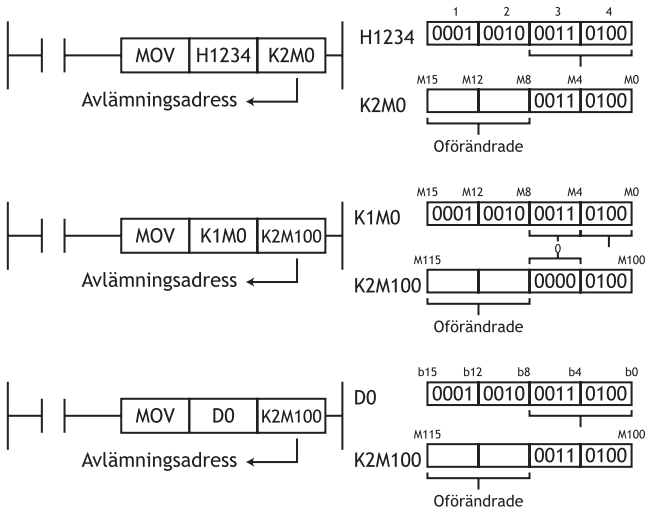
Vid användning av data- och applikationsinstruktioner är det i vissa fall nödvändigt att specificera antalet siffror i bearbetningen. Detta gäller då ordinstruktioner använder bitoperander (X, Y, M, L, B, F). Specificeringen görs genom att ange antalet grupper om 4 bitar. Är operanden lagrad i BCD-kod anger varje sådan 4 bitars grupp en siffra (eller en dekad).

| | * | Antal bitar | Talområde för enkelords-instruktioner | Talområde för dubbelordsinstruntioner |
|---|---|---|---|---|
| K1 | 1 | 4 | 0 till 15 | 0 till 15 |
| K2 | 2 | 8 | 0 till 255 | 0 till 255 |
| K3 | 3 | 12 | 0 till 4 095 | 0 till 4 095 |
| K4 | 4 | 16 | -32 768 till 32 767 | 0 till 65 535 |
| K5 | 5 | 20 | - | 0 till 1 048 575 |
| K6 | 6 | 24 | - | 0 till 16 777 215 |
| K7 | 7 | 28 | - | 0 till 268 435 455 |
| K8 | 8 | 32 | - | -2 147 483 648 till 2 147 483 647 |

* Antal siffror i BCD-form

*Exempel på en enkelordsinstruktion*

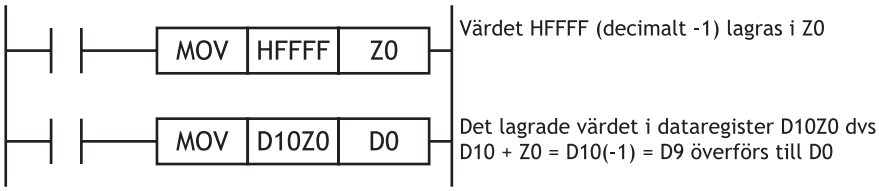
*Exempel på bitangivelse för avlämningsadressen*

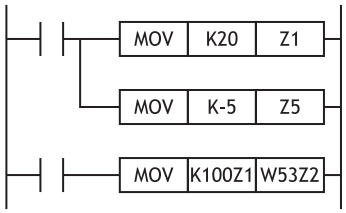
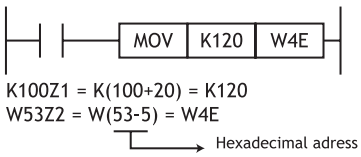
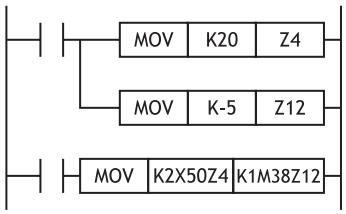
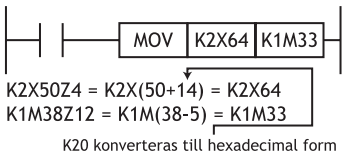
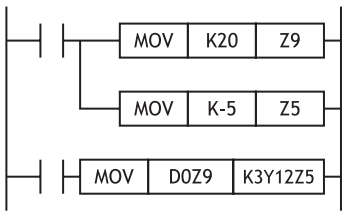
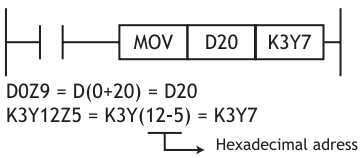
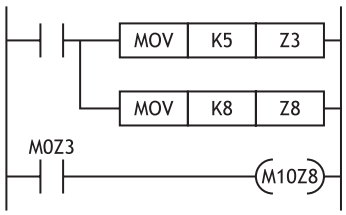
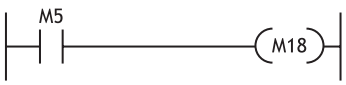


# 5.3 Indexering

Indexering kan användas tillsammans med de flesta operander (X, Y, M, L, B, F, T, C, D, R, W, K och H) där indexvärdet adderas till operandens adress. Index kan anta värde mellan -32 768 och 32 767.

*Exempel*




| Programexempel | Förklaring/resultat |
|---|---|
|  |  |
|  |  |
|  |  |
|  |  |

**Observera!**
QnA(S) och Q02-Q25H har 16 indexregister, Z0 - Z15.
Q00(J) och Q01 har 10 indexregister, Z0 - Z9

# 5.4 Felmeddelande

Om ett fel i programexekveringen uppstår lyser eller blinkar ERROR-dioden på centralenheten. På Q3A och Q4A(R) visas även en text på frontens display. Information om felet kan läsas av dels direkt i programmeringsverktyget och i dels felflaggor och felregister.

Vid parameterinställning kan väljas om ett fel ska stoppa exekveringen eller inte.

### GX IEC Developer

Välj meny: Debug…System Errors.

En felkod och en text presenteras. Om felet kommer av en instruktion i programmet kan hopp till platsen för felet göras (knappen Show).

### GX Developer

Välj meny: Diagnostics…PLC diagnostics.

En felkod och en text presenteras. Om felet kommer av en instruktion i programmet kan hopp till platsen för felet göras (knappen Error Jump).

### Felflaggor och felregister

Det finns specialminnsceller och specialregister som indikerar aktuell status och felmeddelanden. Se kapitel 10 och 11 i instruktionsdelen.

### Återställning av fel

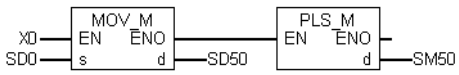
Återställning kan dels göras genom att vrida nyckelbrytaren på centralenhetens front till läge RESET och tillbaka till RUN. Vid reset stoppas programexekveringen. Utgångar, minnesceller mm nollställs.

För att återställa fel utan att stoppa programexekvering kan man, efter att orsaken till felet rättats, använda ett specialregister och en specialminnescell.

Skriv aktuell felkod (från SD0) i SD50.

Ettställ därefter SM50.

### Exempel:

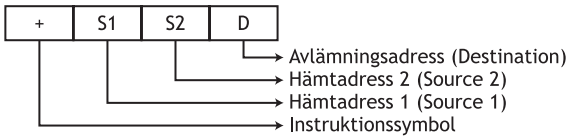


# 6 Instruktionsöversikt

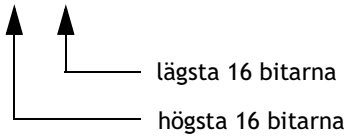
| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Addition och subtraktion av binär data | + | S, D | D+ S → D | → | 3 | *8.10.1* |
| | +P | | | ↑ P | | |
| | + | S1, S2, D1 | S1 + S2 → D1 | → | 4 | |
| | +P | | | ↑ P | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** |

Förklaring till kolumnerna:

1. Instruktionernas användningsområde
2. Visar instruktionen i reläschemasymbol




3. Visar hur många variabler som skall användas.
4. Beskriver instruktionens funktion.
   Enkelordsinstruktion (D) + (S) → (D)
   Dubbelordsinstruktion (D + 1, D) + (S+1, S) → (D +1, D)




5. →: Instruktionen kommer att utföras så länge ingångsvillkoret är uppfyllt.
   ↑ P: Instruktionen kommer att utföras en gång på positiv flank (0 till 1).
   ↓ P: Instruktionen kommer att utföras en gång på negativ pulsflank (1 till 0)
6. Antal steg som instruktionen behöver för uppgiften
7. Visar i vilket kapitel instruktionen beskrivs. Vissa instruktioner beskrivs endast i den engelska programmeringsmanualen MA00350. Detta indikeras med *.

# 6.1 Sekvensinstruktioner

## 6.1.1 Ingångsinstruktioner

| Användning | Namn | Variabler | Funktion | Exekvering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Ingångs-instruktioner | LD | | Startkontakt (Normalt öppen kontakt=NO) | | 1 | *8.2.1* |
| | LDI | | Startkontakt (Normalt stängd kontakt=NC) | | | |
| | AND | | Serieanslutning (NO) | | | |
| | ANI | | Serieanslutning (NC) | | | |
| | OR | | Parallellanslutning (NO) | | | |
| | ORI | | Parallellanslutning (NC) | | | |
| | LDP | | Startkontakt med pulsfunktion Positiv flank | | 2 | *8.2.2* |
| | LDF | | Startkontakt med pulsfunktion Negativ flank | | | |
| | ANDP | S | Serieanslutning med pulsfunktion Positiv flank | | | |
| | ANDF | S | Serieanslutning med pulsfunktion Negativ flank | | | |
| | ORP | S | Parallellanslutning med pulsfunktion Positiv flank | | | |
| | ORF | S | Parallellanslutning med pulsfunktion Negativ flank | | | |

## 6.1.2 Anslutningsinstruktioner

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Anslutnings-instruktioner | ANB | | Seriell block-anslutning | | 1 | *8.3.1* |
| | ORB | | Parallell block-anslutning | | | |
| | MPS | | Lagra resultat (memory push) | | 1 | *8.3.2* |
| | MRD | | Läsa resultat (memory read) | | | |
| | MPP | | Läs och radera resultat (memory pop) | | | |
| | INV | | Invertera resultat | | 1 | *8.3.3* |
| | MEP | | Pulsgenerering på positiv flank | | 1 | * |
| | MEF | | Pulsgenerering på negativ flank | | | |
| | EGP | D | Inställning av flankminnes-cell på positiv flank | | 1 | |
| | EGF | | Inställning av flankminnes-cell på negativ flank | | | |

## 6.1.3 Utgångsinstruktioner

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Utgångs-instruktioner | OUT | D | Utgångs-aktivering | | 1 | *8.4.1*<br>*8.4.2*<br>*8.4.3* |
| | SET | D | Ettställning av operander | → | 1 | *8.4.4* |
| | RST | D | Nollställning av operander | → | 2 | |
| | PLS | D | Utgång på positiv flank | ↑ | 2 | *8.4.5* |
| | PLF | | Utgång på negativ flank | ↓ | | |
| | FF | S | Invertera utgång | ↑ | 2 | *8.4.6* |
| | DELTA | D | Pulsgenerering på utgång | → | 2 | * |
| | DELTAP | | | ↑ | | |

### 6.1.4 Skiftinstruktioner

| Användning | Namn | Variabler | Funktion | Exekvering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Skiftinstruktioner | SFT | D | Skifta bitoperander | → | 2 | *8.5.1* |
| | SFTP | | | ↑ | | |

### 6.1.5 Master control instruktioner

| Användning | Namn | Variabler | Funktion | Exekvering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Master Control-instruktioner | MC | n, D | Aktivering av markerade programdelar | → | 2 | *8.6.1* |
| | MCR | n | Återställning av markerade programdelar | | 1 | |

### 6.1.6 Programavslutande instruktioner

| Användning | Namn | Variabler | Funktion | Exekvering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Avslutningsinstruktioner | FEND | | Avslutning av programgrenar | → | 1 | *8.7.1* |
| | END | | Avslutning av sekvensprogram | | | |

### 6.1.7 Diverse instruktioner

| Användning | Namn | Variabler | Funktion | Exekvering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Diverse instruktioner | STOP | | STOP-instruktion | → | 1 | *8.8.1* |
| | NOP | | Tomt programsteg | → | | *8.8.2* |

# 6.2 Tillämpade instruktioner del 1

## 6.2.1 Jämförelseinstruktioner

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Jämförelse, binära värden | LD= | S1, S2 | Villkor uppfyllt då (S1) = (S2) | | 3 | *8.9.1* |
| | AND= | | | | | |
| | OR= | | | | | |
| | LD<> | S1, S2 | Villkor uppfyllt då (S1) ≠ (S2) | | 3 | |
| | AND<> | | | | | |
| | OR<> | | | | | |
| | LD> | S1, S2 | Villkor uppfyllt då (S1) > (S2) | | 3 | |
| | AND> | | | | | |
| | OR> | | | | | |
| | LD<= | S1, S2 | Villkor uppfyllt då (S1) ≤ (S2) | | 3 | |
| | AND<= | | | | | |
| | OR<= | | | | | |
| | LD< | S1, S2 | Villkor uppfyllt då (S1) < (S2) | | 3 | |
| | AND< | | | | | |
| | OR< | | | | | |
| | LD>= | S1, S2 | Villkor uppfyllt då (S1) ≥ (S2) | | 3 | |
| | AND>= | | | | | |
| | OR>= | | | | | |
| | LDD= | S1, S2 | Villkor uppfyllt då (S1+1, S1) = (S2+1, S2) | | 3 | *8.9.2* |
| | ANDD= | | | | | |
| | ORD= | | | | | |
| | LDD<> | S1, S2 | Villkor uppfyllt då (S1+1, S1) ≠ (S2+1, S2) | | 3 | |
| | ANDD<> | | | | | |
| | ORD<> | | | | | |
| | LDD> | S1, S2 | Villkor uppfyllt då (S1+1, S1) > (S2+1, S2) | | 3 | |
| | ANDD> | | | | | |
| | ORD> | | | | | |
| | LDD<= | S1, S2 | Villkor uppfyllt då (S1+1, S1) ≤ (S2+1, S2) | | 3 | |
| | ANDD<= | | | | | |
| | ORD<= | | | | | |
| | LDD< | S1, S2 | Villkor uppfyllt då (S1+1, S1) < (S2+1, S2) | | 3 | |
| | ANDD< | | | | | |
| | ORD< | | | | | |
| | LDD>= | S1, S2 | Villkor uppfyllt då (S1+1, S1) ≥ (S2+1, S2) | | 3 | |
| | ANDD>= | | | | | |
| | ORD>= | | | | | |

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Jämförelse av flyttals-data | LDE= | S1, S2 | Villkor uppfyllt då (S1+1, S1) = (S2+1, S2) | | 3 | *8.9.3* |
| | ANDE= | | | | | |
| | ORE= | | | | | |
| | LDE<> | S1, S2 | Villkor uppfyllt då (S1+1, S1) ≠ (S2+1, S2) | | 3 | |
| | ANDE<> | | | | | |
| | ORED<> | | | | | |
| | LDE> | S1, S2 | Villkor uppfyllt då (S1+1, S1) > (S2+1, S2) | | 3 | |
| | ANDE> | | | | | |
| | ORE> | | | | | |
| | LDE<= | S1, S2 | Villkor uppfyllt då (S1+1, S1) ≤ (S2+1, S2) | | 3 | |
| | ANDE<= | | | | | |
| | ORE<= | | | | | |
| | LDE< | S1, S2 | Villkor uppfyllt då (S1+1, S1) < (S2+1, S2) | | 3 | |
| | ANDED< | | | | | |
| | ORE< | | | | | |
| | LDE>= | S1, S2 | Villkor uppfyllt då (S1+1, S1) ≥ (S2+1, S2) | | 3 | |
| | ANDE>= | | | | | |
| | ORE>= | | | | | |
| Jämförelse av tecken-strängar | LD$= | S1, S2 | Villkor uppfyllt då (S1) = (S2) | | 3 | *8.9.4* |
| | AND$= | | | | | |
| | OR$= | | | | | |
| | LD$<> | S1, S2 | Villkor uppfyllt då (S1) ≠ (S2) | | 3 | |
| | AND$<> | | | | | |
| | OR$<> | | | | | |
| | LD$> | S1, S2 | Villkor uppfyllt då (S1) > (S2) | | 3 | |
| | AND$> | | | | | |
| | OR$> | | | | | |
| | LD$<= | S1, S2 | Villkor uppfyllt då (S1) ≤ (S2) | | 3 | |
| | AND$<= | | | | | |
| | OR$<= | | | | | |
| | LD$< | S1, S2 | Villkor uppfyllt då (S1) < (S2) | | 3 | |
| | AND$< | | | | | |
| | OR$< | | | | | |
| | LD$>= | S1, S2 | Villkor uppfyllt då (S1) ≥ (S2) | | 3 | |
| | AND$>= | | | | | |
| | OR$>= | | | | | |

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Jämförelse av binär data i block | BKCMP= | S1, S2, n, D1 | Jämför det n:te binära 16-bitarsblocket i S1 med det n:te binära 16-bitarsblocket i S2, med början på det första registret. Resultatet av varje blockjämförelse lagras i D1 och framåt. | → | 5 | * |
| | BKCMP<> | | | | | |
| | BKCMP> | | | | | |
| | BKCMP<= | | | | | |
| | BKCMP< | | | | | |
| | BKCMP= | | | | | |
| | BKCMP=P | | | ↑ | | |
| | BKCMP<>P | | | | | |
| | BKCMP>P | | | | | |
| | BKCMP<=P | | | | | |
| | BKCMP<P | | | | | |
| | BKCMP=P | | | | | |

## 6.2.2 Aritmetiska instruktioner

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Addition och subtraktion av binära värden | + | S, D | (D) + (S) → (D) | → | 3 | *8.10.1* |
| | + P | | | ↑P | | |
| | + | S1, S2, D1 | (S1)+(S2) → (D1) | → | 4 | |
| | + P | | | ↑P | | |
| | - | S, D | (D) - (S) → (D) | → | | |
| | - P | | | ↑P | | |
| | - | S1, S2, D1 | (S1) - S2) → (D1) | → | 4 | |
| | - P | | | ↑P | | |
| | D+ | S, D | (D+1, D) + (S+1, S) → (D+1, D) | → | 3 | *8.10.2* |
| | D+P | | | ↑P | | |
| | D+ | S1, S2, D | (S1+1, S1) + (S2+1, S2) → (D+1, D) | → | 4 | |
| | D+P | | | ↑P | | |
| | D- | S, D | (D+1,D) - (S+1, S) → (D+1, D) | → | 3 | |
| | D-P | | | ↑P | | |
| | D- | S1, S2, D1 | (S1+1, S1) - (S2+1, S2) → (D1+1, D) | → | 4 | |
| | D-P | | | ↑P | | |
| Multiplikation och division av binära värden | x | S1, S2, D1 | (S1) x (S2) → (D+1,D1) | → | 4 | *8.10.3* |
| | xP | | | ↑P | | |
| | / | S1, S2, D1 | (S1) / (S2) → Kvot (D1), Rest (D1+1) | → | 4 | |
| | /P | | | ↑P | | |
| | Dx | S1, S2, D1 | (S1+1,S1) x (S2+1,S2) → (D1+3, D1+2, D1+1, D1) | → | 4 | *8.10.4* |
| | DxP | | | ↑P | | |
| | D/ | S1, S2, D1 | (S1+1,S1) / (S2+1,S2) → Kvot (D1+1,D1), Rest (D3+3, D1+2) | → | 4 | |
| | D/P | | | ↑P | | |
| 4-siffrig addition och subtraktion av BCD-värden | B+ | S, D | (D) + (S) → (D) | → | 3 | *8.10.5* |
| | B+P | | | ↑P | | |
| | B+ | S1, S2, D1 | (S1) + (S2) → (D1) | → | 4 | |
| | B+P | | | ↑P | | |
| | B- | S, D | (D) - (S) → (D) | → | 3 | |
| | B-P | | | ↑P | | |
| | B- | S1, S2, D1 | (S1) - (S2) → (D1) | → | 4 | |
| | B-P | | | ↑P | | |

| Användning | Namn | Variabler | Funktion | Exekvering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| 8-siffrig addition och subtraktion av BCD-värden | DB+ | S, D | (D+1,D) + (S+1,S) → (D+1,D) | → | 3 | *8.10.6* |
| | DB+P | | | ↑P | | |
| | DB+ | S1, S2, D1 | (S1+1,S1) + (S2+1,S2) → (D1+1,D1) | → | 4 | |
| | DB+P | | | ↑P | | |
| | DB- | S, D | (D+1,D) - (S+1,S) → (D+1,D)) | → | 3 | |
| | DB-P | | | ↑P | | |
| | DB- | S1, S2, D1 | (S1+1,S1) + (S2+1,S2) → (D1+1,D1) | → | 4 | |
| | DB-P | | | ↑P | | |
| 4-siffrig multiplikation och division av BCD-värden | Bx | S1, S2, D1 | (S1) x (S2) →(D1+1,D1) | → | 4 | *8.10.7* |
| | BxP | | | ↑P | | |
| | B/ | S1, S2, D1 | (S1) / (S2) → Kvot (D1), Rest (D1+1) | → | 4 | |
| | B/P | | | ↑P | | |
| 8-siffrig multiplikation och division av BCD-värden | DBx | S1, S2, D1 | (S1+1,S1) x (S2+1,S2 → (D1+3,D1+2, D1+1,D1) | → | 4 | *8.10.8* |
| | DBxP | | | ↑P | | |
| | DB/ | S1, S2, D1 | (S1+1,S1) / (S2+1,S2) → Kvot (D1+1,D1), Rest (D1+3, D1+2) | → | 4 | |
| | DB/P | | | ↑P | | |
| Addition och subtraktion av flyttal | E+ | S, D | (D+1,D)+(S+1,S) → (D+1,D) | → | 3 | *8.10.9* |
| | E+P | | | ↑P | | |
| | E+ | S1, S2, D1 | (S1 +1, S1) + (S2+1, S2) → (D1+1, D1) | → | 4 | |
| | E+P | | | ↑P | | |
| | E- | S, D | (D+1, D)-(S+1, S) → (D+1,D) | → | 3 | |
| | E-P | | | ↑P | | |
| | E- | S1, S2, D1 | (S1 +1, S1) - (S2+1, S2) → (D1+1, D1) | → | 4 | |
| | E-P | | | ↑P | | |
| Multiplikation och division av flyttal | Ex | S1, S2, D1 | (S1 +1, S1) x (S2+1, S2) → (D1+1, D1) | → | 4 | *8.10.10* |
| | ExP | | | ↑P | | |
| | E/ | S1, S2, D1 | (S1+1,S1) / (S2+1,S2) → Kvot (D1+1,D1) | → | 4 | |
| | E/P | | | ↑P | | |
| Addition och subtraktion av binära värden i block | BK+ | S1, S2, D, n | Adderar n:te 16-bitars-blocket i S1 till n:te 16-bitars-blocket i S2 | → | 5 | * |
| | BK+P | | | ↑P | | |
| | BK- | S1, S2, D, n | Subtraherar n:te 16-bitars-blocket i S2 från n:te 16-bitarsblocket i S1 | → | 5 | |
| | BK-P | | | ↑P | | |

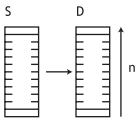
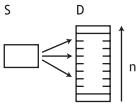
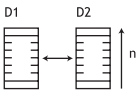
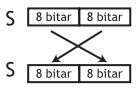
| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Länkning av tecken-strängar | $+ | S, D | Teckendata i S är utökad till teckendata i D. Den länkade teckendatan sparas i D | → | 3 | *8.10.11* |
| | $+P | | | ↑P | | |
| | $+ | S1, S2, D1 | Teckendata i S1 utökas med teckendata i S2. Den länkade teckendatan sparas i D | → | 4 | |
| | $+P | | | ↑P | | |
| Öka binärt värde med 1 | INC | D | (D) +1 → (D) | → | 2 | *8.10.12* |
| | INCP | | | ↑P | | |
| | DINC | D | (D+1,D) + 1 → (D+1,D) | → | 2(3) | *8.10.13* |
| | DINCP | | | ↑P | | |
| Minska binärt värde med 1 | DEC | D | (D) -1 → (D) | → | 2 | *8.10.12* |
| | DECP | | | ↑P | | |
| | DDEC | D | (D+1,D) - 1 → (D+1,D) | → | 2(3) | *8.10.13* |
| | DDECP | | | ↑P | | |

## 6.2.3 Omvandlingsinstruktioner

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Omvandling av binärdata till BCD-data | BCD | S, D | BIN(S) {0-9999} → BCD→ (D) | → | 3 | *8.11.1* |
| | BCDP | | | ↑P | | |
| | DBCD | S, D | BIN (S+1,S) {0-9999 9999} → BCD → (D+1, D) | → | 3 | |
| | DBCDP | | | ↑P | | |
| Omvandling av BCD-data till binärdata | BIN | S, D | BCD(S) {0-9999} → BIN → (D) | → | 3 | *8.11.2* |
| | BINP | | | ↑P | | |
| | DBIN | S, D | BCD (S+1,S) {0-9999 9999} → BIN → (D+1, D) | → | 3 | |
| | DPINP | | | ↑P | | |
| Omvandling av binärdata till flyttal | FLT | S, D | BIN (S+1,S) {-32 768- 32 767} → Flyttal→ (D+1, D) | → | 3 | *8.11.3* |
| | FLTP | | | ↑P | | |
| | DFLT | S, D | BIN (S+1,S) {-2 147 483 648 - 2 147 483 647} → Flyttal → (D+1, D) | → | 3 | |
| | DFLTP | | | ↑P | | |
| Omvandling av flyttal till binärdata | INT | S, D | Flyttal (S+1,S) {-32 768 - 32 767} → BIN→ (D+1, D) | → | 3 | *8.11.4* |
| | INTP | | | ↑P | | |
| | DINT | S, D | Flyttal (S+1,S) {-2 147 483 648 - 2 147 483 647} → BIN→ (D+1, D) | → | 3 | |
| | DINTP | | | ↑P | | |
| Omvandling från binärt enkelord till binärt dub-belord | DBL | S, D | BIN{-32 768 - 32 767} → (D+1, D) | → | 3 | *8.11.5* |
| | DBLP | | | ↑P | | |
| Omvandling från binärt dubbelord till binärt enkel-ord | WORD | S, D | BIN(S+1,S) {-32 768 - 32 767} → D | → | 3 | *8.11.6* |
| | WORDP | | | ↑P | | |

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Omvandling från binär-data till Graykod | GRY | S, D | BIN S {-32 768-32 767} → Graykod → D | → | 3 | * |
| | GRYP | | | ↑P | | |
| | DGRY | S, D | BIN (S+1,S) {-32 768 - 32 767}→Gray-kod→ (D+1,D) | → | 3 | |
| | DGRYP | | | ↑P | | |
| Omvandling från Graykod till binärdata | GBIN | S, D | Graykod S {-32 768-32 767} → BIN → D | → | 3 | |
| | GBINP | | | ↑P | | |
| | DGBIN | S, D | Graykod (S+1,S) {-2 147 483 648-2 147 483 647} → (D+1,D) | → | 3 | |
| | DGBINP | | | ↑P | | |
| Teckenbyte för binär-data (2-kom-plement) | NEG | D | BIN D → D | → | 2 | *8.11.7* |
| | NEGP | | | ↑P | | |
| | DNEG | D | BIN (D+1,D) → (D+1,D) | → | 2 | |
| | DNEGP | | | ↑P | | |
| Teckenbyte för flyttal | ENEG | D | (D+1,D) → (D+1,D) | → | 2 | *8.11.8* |
| | ENEGP | | | ↑P | | |
| Omvandling från binär blockdata till BCD-block-data | BKBCD | S, D, n | Omvandlar n:te BIN 16-bitars-block i S till n:te BCD 4-siff-rigt block. Kon-verterad data lagras i D | → | 4 | * |
| | BKBCDP | | | ↑P | | |
| Omvandling från BCD-blockdata till binär block-data | BKBIN | S, D, n | Omvandlar n:te BCD 4-siffrigt block i S till n:te BIN 16-bitarsblock Kon-verterad data lagras i D | → | 4 | |
| | BKBINP | | | ↑P | | |

## 6.2.4 Överföringsinstruktioner

| Användning | Namn | Variabler | Funktion | Exekvering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Dataöverföring | MOV | S, D | S → D | → | 3 | *8.12.1* |
| | MOVP | S, D | | ↑P | | |
| | DMOV | S, D | (S+1,S) → (D+1,D) | → | 3 | |
| | DMOVP | S, D | | ↑P | | |
| Överföring flyttalsdata | EMOVE | S, D | (S+1,S) → (D+1,D) | → | 3 | *8.12.2* |
| | EMOVEP | S, D | | ↑P | | |
| Överföring teckensträng | $MOV | S, D | S → D | → | 3 | *8.12.3* |
| | $MOVP | S, D | | ↑P | | |
| Invertering och överföring | CML | S, D | $\overline{S}$ → D | → | 3 | *8.12.4* |
| | CMLP | S, D | | ↑P | | |
| | DCML | S, D | ($\overline{S+1,S}$)→ (D+1,D) | → | 3 | |
| | DCMLP | S, D | | ↑P | | |
| Binär blocköverföring | BMOV | S, n, D |  | → | 4 | *8.12.5* |
| | BMOVP | S, n, D | | ↑P | | |
| Blocköverföring av identisk binärdata | FMOV | S, n, D |  | → | 4 | *8.12.6* |
| | FMOVP | S, n, D | | ↑P | | |
| Datautbyte | XCH | D1, D2 | D1↔D2 | → | 3 | *8.12.7* |
| | XCHP | D1, D2 | | ↑P | | |
| | DXCH | D1, D2 | (D1+1, D1) ↔ (D2+1, D2) | → | 3 | |
| | DXCHP | D1, D2 | | ↑P | | |
| Binär blockdatautbyte | BXCH | n, D1, D2 |  | | 4 | * |
| | BXCH | n, D1, D2 | | ↑P | | |
| Växling av högre och lägre byte | SWAP | S |  | → | 3 | *8.12.8* |
| | SWAPP | S | | ↑P | | |

## 6.2.5 Förgreningsinstruktioner

| Användning | Namn | Variabler | Funktion | Exekvering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Hoppinstruktioner | CJ | P | Villkorligt hopp till Pxx | → | 2 | *8.13.1* |
| | SCJ | P | Villkorligt hopp till Pxx nästa programvarv | → | | |
| | JMP | P | Hoppinstruktion | | 2 | |
| | GOEND | | Hopp till END av programmet | → | 1 | * |

## 6.2.6 Interruptinstruktioner

| Användning | Namn | Variabler | Funktion | Exekvering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Interrupt ej aktivt | DI | | Stänger av exekvering av interruptprogram | → | 2 | *8.14.1* |
| Interrupt aktivt | EI | | Aktiverar interruptprogram | → | | |
| Aktivera interupt med bitmönster | IMASK | S | Bitmönster styr aktivering av olika interrupt | | 2 | |
| Återgång till huvudprogrammet | IRET | | Slut på interruptprogrammet | → | 1 | *8.14.2* |

## 6.2.7 Uppdateringsinstruktioner

| Användning | Namn | Variabler | Funktion | Exekvering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Deluppdatering av I/O-kort | RFS | S, n | RFS-instruktionen uppdaterar in- och utgångarnas adressområde under ett programvarav | ↑P | 3 | *8.15.1* |

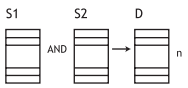
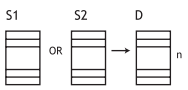
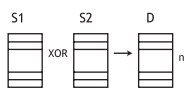
## 6.2.8 Andra behändiga instruktioner

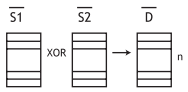
| Användning | Namn | Variabler | Funktion | Exekvering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| 1-fas ingång upp-/nedräknare | UDCNT1 | S, n, D | Instruktionen UDCNT1 registrerar pulser på ingången S och lagrar dem på räknare D, förvalsvärde n, s+1 definierar riktning. | → | 4 | * |
| 2-fas ingång upp-/nedräknare | UDCNT2 | S, n, D | Instruktionen UDCNT2 registrerar pulser från en 2-fas givare S(A-fas) och S+1(B-fas) och lagrar dem på räknare D, förvalsvärde n | → | 4 | * |
| Inspelningstidskrets | TTMR | D, n | D+1 mäter hur lång tid villkoret är uppfyllt för instruktionen TTMR. Resultatet multipliceras med en faktor (n) och lagras i D.<br>n=0; faktor=1<br>n=1; faktor=10<br>n=2; faktor=100 | → | 3 | * |
| Specialfunktion för tidsrets (låghastighet) | STMR | S, n, D | Instruktionen STMR kan utföra fyra olika timerfunktioner på utgångarna beroende på D+0 till D+3:<br>D+0: Frånslagsfördröjd timer<br>D+1: En puls på utgången. Pulsen startar vid en negativ flank.<br>D+2: En puls på utgången. Pulsen startar vid en positiv flank.<br>D+3: Tillslagsfördröjd timer | | 3 | * |
| Specialfunktion för tidskrets (höghastighet) | STMRH | S, n, D | Se ovan! | | 3 | * |

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Positions-instruktion för rote-rande bord | RTOC | S, n1, n2, D | Instruktionen ROTC roterar en sektor på ett bord, som bestäms av S+2, till en position som anges i S+1. Antalet sektorer på bor-det anges i n1. | → | 5 | * |
| Rampfunk-tion | RAMP | n1, n2, n3, D1, D2 | Instruktionen RAMP ändrar innehållet i (D1)+0 gradvis från det initi-ala värdet, n1, till slutvärdet, n2. | → | 6 | * |
| Hastghets-mätning | SPD | S, n, D | Instruktionen SPD räknar antalet pulser på ingången S under en tids-period som bestäms av n. Antalet pulser lagras i D. | → | 4 | * |
| Variabel puls-utgång | PLSY | S1, S2, D | Instruktionen PLSY skickar ut pulser på utgången som bestäms av D. Frekvensen på pulserna anges i S1 och antalet pulser anges i S2. | → | 4 | * |
| Pulsbredds modulering | PWM | n1, n2, D | Instruktionen PWM reglerar förhållandet mellan gång-och paustid i ett pulståg på utgången D. Gångtiden bestäms av n1 och hela peri-odtiden bestäms av n2. | → | 4 | * |
| Läsning av matris | MTR | S, n, D1, D2 | Instruktionen MTR läser från startadressen som anges i S. Antalet upprep-ningar (rader) bestäms av n. Datan lagras i D2 och framåt. | → | 5 | * |

# 6.3 Tillämpade instruktioner del 2

## 6.3.1 Logiska instruktioner

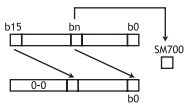
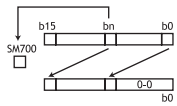
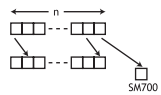
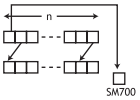
| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Logisk pro-dukt, AND | WAND | S, D | (D) AND (S) → (D) | → | 3 | *8.16.1* |
| | WANDP | | | ↑P | | |
| | WAND | S1, S2, D1 | (S1) AND (S2) → (D1) | → | 4 | |
| | WANDP | | | ↑P | | |
| | DAND | S, D | (D+1, D) AND (S+1, S) → (D+1, D) | | 3 | |
| | DANDP | | | ↑P | | |
| | DAND | S1, S2, D1 | (S1+1,S1) AND (S2+1,S2) → (D+1, D) | → | 4 | |
| | DANDP | | | ↑P | | |
| | BKAND | S1, S2, n, D |  | → | 5 | * |
| | BKANDP | | | ↑P | | |
| Logisk summa, OR | WOR | S, D | (D) OR (S)→(D) | → | 3 | *8.16.2* |
| | WORP | | | ↑P | | |
| | WOR | S1, S2, D1 | (S1) OR (S2)→(D1) | → | 4 | |
| | WORP | | | ↑P | | |
| | DOR | S, D | (D+1, D) OR (S+1,S) → (D+1, D) | → | 3 | |
| | DORP | | | ↑P | | |
| | DOR | S1, S2, D | (S1+1,S1) OR (S2+1,S2) → (D+1, D) | → | 4 | |
| | DORP | | | ↑P | | |
| | BKOR | S1, S2, n, D |  | → | 5 | * |
| | BKORP | | | ↑P | | |
| Logisk exklusiv OR | WXOR | S, D | S XOR D | → | 3 | *8.16.3* |
| | WXORP | | | ↑P | | |
| | WXOR | S1, S2, D1 | S1 XOR S2, D1 | → | 4 | |
| | WXORP | | | ↑P | | |
| | DXOR | S, D | (D+1, D) XOR (S+1,S) → (D+1, D) | | 3 | |
| | DXORP | | | ↑P | | |
| | DXOR | S1, S2, D | (S1+1,S1) XOR (S2+1,S2) → (D+1, D) | → | 4 | |
| | DXORP | | | ↑P | | |
| | BKXOR | S1, S2, n, D |  | → | 5 | * |
| | BKXORP | | | ↑P | | |

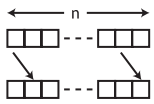
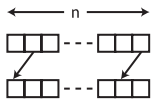
| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Logisk exklusiv NOR | WNXR | S, D | $\overline{D \text{ XOR } S} \rightarrow D$ | → | 3 | *8.16.4* |
| | WNXRP | | | ↑P | | |
| | WNXR | S1, S2, D1 | $\overline{S1 \text{ XOR } S2} \rightarrow D1$ | → | 4 | |
| | WNXRP | | | ↑P | | |
| | DNXR | S, D | $\overline{(D+1, D) \text{ XOR } (S+1, S)} \rightarrow (D+1, D)$ | → | 3 | |
| | DNXRP | | | ↑P | | |
| | DNXR | S1, S2, D | $\overline{(S1+1, S1) \text{ XOR } (S2+1, S2)} \rightarrow (D+1, D)$ | → | 4 | |
| | DNXRP | | | ↑P | | |
| | BKXNR | S1, S2, n, D |   | → | 5 | * |
| | BKXNRP | | | ↑P | | |

## 6.3.2 Rotationsinstruktioner

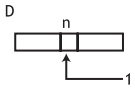
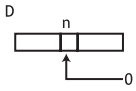
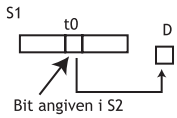
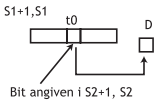
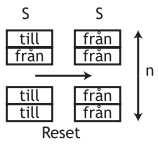
| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Rotation till höger | ROR | n, D | Roterar n bitar höger utan carryflagga | → | 3 | *8.17.1* |
| | RORP | | | ↑P | | |
| | RCR | n, D | Roterar n bitar höger med carryflagga | → | 3 | |
| | RCRP | | | ↑P | | |
| Rotation till vänster | ROL | n, D | Roterar n bitar vänster utan carryflagga | → | 3 | *8.17.2* |
| | ROLP | | | ↑P | | |
| | RCL | n, D | Roterar n bitar vänster med carryflagga | → | 3 | |
| | RCLP | | | ↑P | | |
| Rotation till höger | DROR | n, D | Roterar n bitar höger utan carryflagga | → | 3 | *8.17.3* |
| | DRORP | | | ↑P | | |
| | DRCR | n, D | Roterar n bitar höger med carryflagga | → | 3 | |
| | DRCRP | | | ↑P | | |
| Rotation till vänster | DROL | n, D | Roterar n bitar vänster utan carryflagga | → | 3 | *8.17.4* |
| | DROLP | | | ↑P | | |
| | DRCL | n, D | Roterar n bitar vänster med carryflagga | → | 3 | |
| | DRCLP | | | ↑P | | |

## 6.3.3 Skiftinstruktioner

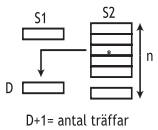
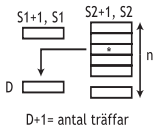
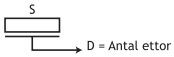
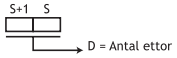
| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Skifta en ordoperand, n bitar åt höger | SFR | n, D |  | → | 3 | *8.18.1* |
| | SFRP | | | ↑P | | |
| Skifta en ordoperand, n bitar åt vänster | SFL | n, D |  | → | 3 | |
| | SFLP | | | ↑P | | |
| Skifta n bit-operander en bit åt höger | BSFR | n, D |  | → | 3 | *8.18.2* |
| | BSFRP | | | ↑P | | |
| Skifta n bit-operander en bit åt vänster | BSFL | n, D |  | → | 3 | |
| | BSFLP | | | ↑P | | |

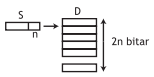
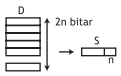
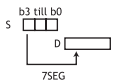
| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Skifta n ord-operander ett ord åt höger | DSFR<br>DSFRP | n, D |  | →<br>↑P | 3 | *8.18.3* |
| Skifta n ord-operander ett ord åt vänster | DSFL<br>DSFLP | n, D |  | →<br>↑P | 3 | |

## 6.3.4 Bitinstruktioner

| Användning | Namn | Variabler | Funktion | Exekvering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Ettställ/nollställ enskild bit i ordoperand | BSET<br>BSETP | n, D |  | →<br>↑P | 3 | *8.19.1* |
| | BRST<br>BRSTP | n, D |  | →<br>↑P | 3 | |
| Kontrollera status på enskild bit i ordoperand | TEST<br>TESTP | S1, S2, D |  | →<br>↑P | 4 | *8.19.2* |
| | DTEST<br>DTESTP | S1, S2, D |  | →<br>↑P | 4 | |
| Nollställ flera bitar | BKRST<br>BKRSTP | S, n |  | →<br>↑P | 3 | * |

## 6.3.5 Datainstruktioner

| Användning | Namn | Variabler | Funktion | Exekvering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Sökning i dataområde | SER<br>SERP | S1, S2, n, D |  | →<br>↑P | 5 | *8.20.1* |
| | DSER<br>DSERP | S1, S2, n, D |  | →<br>↑P | 5 | |
| Summering av antalet ettställda bitar i ordoperand | SUM<br>SUMP | S, D |  | →<br>↑P | 3 | *8.20.2* |
| | DSUM<br>DSUMP | S, D |  | →<br>↑P | 3 | |

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Avkodning av 1-8 bitar till 256 bitar | DECO<br>DECOP | S, D, n |  | →<br>↑P | 4 | *8.20.3* |
| Kodning av 1-256 bitar till 8 bitar | ENCO<br>ENCOP | S, D, n |  | →<br>↑P | 4 | *8.20.4* |
| Avkodning av 7-segments display | SEG<br>SEGP | S, D |  | →<br>↑P | 3 | * |
| Upp-/ned-packning av ordoperander | DIS<br>DISP | S, n, D | Instruktionen packar upp enkelord till grupper om 4 bitar. Datavärden packas upp i S och antalet grupper anges i n. Den första adressen anges i D. | →<br>↑P | 4 | *8.20.5* |
| | UNI<br>UNIP | S, n, D | Instruktionen packar ner de 4 lägsta bitarna i upp till fyra ord till ett enkelord | →<br>↑P | 4 | *8.20.6* |

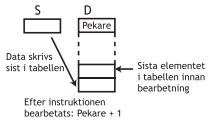
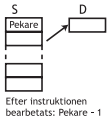
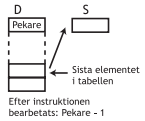
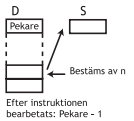
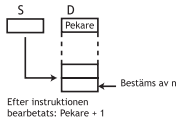
| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Upp-/ned-packning av ordoperander | NDIS<br>NDISP | S1, D, S2 | Instruktionen NDIS splittar datan i registren, som anges i S1, i grupper. Antalet bitar i varje grupp anges i S2. Grupperna lagras separat i registrerna som anges i D och framåt. | →<br>↑P | 4 | * |
| | NUNI<br>NUNIP | S1, D, S2 | Instruktionen NUNI slår samman grupper av bitar i registrerna, som anges i S1, till ett datavärde. Antalet bitar i varje grupp anges i S2. Grupperna lagras i registrerna som anges i D och framåt. | →<br>↑P | 4 | * |
| | WTOB<br>WTOBP | S, n, D | WTOB delar upp n ord med start på S i byte och lagrar resultatet i D. | →<br>↑P | 4 | * |
| | BTOW<br>BTOWP | S, n, D | BTOW kopplar ihop de låga byten i n ord med start på S och och lagrar resultatet i D. | →<br>↑P | 4 | |
| Sökning av maximum-värden i ordoperander | MAX<br>MAXP | S, n, D | MAX-instruktionen söker efter maxvärdet i datablock med start på S och längden n. Det största värdet lagras i D. | →<br>↑P | 4 | * |
| | DMAX<br>DMAXP | S, n, D | DMAX-instruktionen söker efter maxvärdet i datablock med start på S och längden 2 x n. Det största värdet lagras i D | →<br>↑P | 4 | |

## 6.3.6 Strukturerade programinstruktioner

| Användning | Namn | Variabler | Funktion | Exekvering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Repetionsinstruktioner | FOR | n | En FOR/NEXT-loop upprepar en programsekvens utan att ingångarnas tillstånd sätts. Programsekvensen som omges av FOR och NEXT upprepas n gånger. | | 2 | *8.21.1* |
| | NEXT | | | | 1 | |
| | BREAK | p, D | Instruktionen BREAK avbryter en FOR/NEXT-loop och hoppar till pekaren som anges av p. | → | 3 | *8.21.2* |
| | BREAKP | | | ↑P | | |
| Subrutin programanrop | CALL | p | Instruktionen CALL startar den subrutin som anges av pekaren Pxx. | → | 2 | *8.21.3* |
| | CALLP | | | ↑P | | |
| | RET | | Instruktionen RET markerar slutet på en subrutin | | 1 | *8.21.4* |
| | FCALL | p | På fallande flank på exekveringsvillkoret till instruktionen FCALL återgår utgångarna i subrutinen, som anges av pekaren p, till samma värde som om instruktionens exekverings villkor inte har satts. | → | 2 | *8.21.5* |
| | FCALLP | | | ↑P | | |

| Användning | Namn | Variabler | Funktion | Exekvering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Anrop av subrutinprogram mellan programfiler | ECALL | Filnamn, p | Instruktionen ECALL startar en subrutin som anges av pekaradressen i en programfil med namnet som anges i Filnamn. | → | 3 | * |
| | ECALLP | | | ↑P | | |
| Anrop av subrutinprogram mellan programfiler | EFCALL | Filnamn, p | På fallande flank på exekveringsvillkoret till instruktionen EFCALL återgår utgångarna i subrutinen, som anges av pekaren p, till samma värde som om instruktionens exekverings villkor inte har satts. | → | 3 | * |
| | EFCALLP | | | ↑P | | |
| Uppdatering av kommunikations och gränssnittsdata | COM | S | Om SM775 inte är 0 uppdateras kommunikations och gränssnittsdata (END exekveras) | | 4 | *8.21.6* |
| Stänger av kommunikations uppdataringen | DI | | Stänger av kommunikations uppdataringen tills EI exekveras | | 1 | *8.14.1* |
| Aktiverar kommunikations uppdataringen | EI | | Aktiverar kommunikations uppdataringen efter EI instruktionen | | 1 | |
| Indexering av operander mellan IX och IXEND | IX | S | Instruktionerna IX och IXEND indexerar de register i programdelen mellan IX och IXEND. | | 1 | * |
| | IXEND | | | | 2 | |
| Benämningar på lämpliga värden i index | IXDEV | | | ↑P | 1 | |
| | IXSET | p, D | | | 1 | |

## 6.3.7 Tabellinstruktioner

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Skriva data till tabell (FIFO) | FIFW<br>FIFWP | S, D |  | →<br>↑P | 3 | *8.22.1* |
| Läs från början av tabell (FIFO) | FIFR<br>FIFRP | S, D |  | →<br>↑P | 3 | *8.22.2* |
| Läs från slutet av tabell (FIFO) | FPOP<br>FPOPP | S, D |  | →<br>↑P | 3 | *8.22.3* |
| Radera definierade datablock från tabell (FIFO) | FDEL<br>FDELP | S, n, D |  | →<br>↑P | 4 | * |
| Infogar datablock till tabell (FIFO) | FINS<br>FINSP | |  | →<br>↑P | | * |

## 6.3.8 Buffertminnesinstruktioner

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Läsa data från en funktionsmodul | FROM<br>FROMP | n1, n2, n3, D | Instruktionen FROM läser data med storleken ett ord från buffertminnet i en funktions-modul. | →<br>↑P | 5 | *8.23.1* |
| | DFRO<br>DFROP | | Instruktionen DFRO läser data med storleken två ord från buffertminnet i en funktionsmo-dul | →<br>↑P | | |

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Skriva data till en funktionsmodul | TO | S, n1, n2, n3, | Instruktionen TO skriver data med storleken ett ord från minnet i CPU till buffertminnet i en funktionsmodul. | → | 5 | *8.23.2* |
| | TOP | | | ↑P | | |
| | DTO | | Instruktionen DTO skriver data med storleken två ord från minnet i CPU till buffertminnet i en funktionsmodul. | → | | |
| | DTOP | | | ↑P | | |

## 6.3.9 Displayinstruktioner

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| ASCII-tecken utgång | PR | S, D | SM701 aktiverad (1): Instruktionen PR skickar ut en ASCII-teckensträng på 16 tecken till en utgångsmodul. Teckensträngen delas i två delar med 8 tecken per del. Strängen läses från adressarean S och skickas till utgångarna som anges i D. SM701 nollställd (0): Instruktionen PR skickar ut en ASCII-teckensträng fram till tecknet med koden "00H" i hexadecimalt format. Strängen läses från adressarean S och skickas till utgångarna som anges i D. | ↑P | 3 | * |
| | PRC | S, D | Instruktionen PRC skickar ut en kommentar till ett register (ASCII-kodat) till en utgångsmodul. Om SM701 (MELSEC Q) är aktiverad (1), skickas 16 tecken ut. Om SM701 (MELSEC Q) är nollställd (0), skickas 32 tecken ut | ↑P | 3 | * |

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| ASCII-tecken display, kommentarer | LED | S | Instruktionen LED läser ASCII-data (16 tecken) från en angiven adressarea och visar det på en passande CPU-display. | → | 2 | * |
| | LEDC | S | Instruktionen LEDC läser kommentarer (16 tecken) från en angiven adressarea och visar det på en passande CPU-display. | → | 2 | |
| | LEDR | | Nollställer displayer för felkoder. | → | 1 | |

## 6.3.10 Debug- och feldiagnosinstruktioner

| Användning | Namn | Variabler | Funktion | Exekvering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Felkontroll | CHKST | | Instruktionen CHKST startar exekveringen av instruktionen CHK. Om exekveringsvillkoret för instruktionen CHKST är nollställt (0), kommer instruktionen efter CHK att exekveras. | | 1 | * |
| | CHK | | Instruktionen CHK stödjer felsökningsfunktioner för kontaktanslutningar. När ett fel upptäcks kommer SM80 aktiveras och en felkod presenteras i SD80. | | 1 | |
| | CHKCIR | | Startinstruktion för program som skall kontrolleras av CHK-instruktionen. | | 1 | * |
| | CHKEND | | Avslutningsinstruktion för program som skall kontrolleras av CHK-instruktionen. | | 1 | * |
| Aktivera / återställ Status- Latch | SLT | | Aktivera Status Latch | ↑P | 1 | * |
| | SLTR | | Nollställer resultat av Status Latch | | | |
| Aktivera / återställ Sampling Trace | STRA | | Aktivera Sampling Trace | ↑P | 1 | |
| | STRAR | | Återställ Sampling Trace | | | |
| Exekvera/ återställ resetprogram trace | PTRA | | Aktiverar program trace | ↑P | 1 | |
| | PTRAR | | Återställ program trace | | | |
| | PTRAEXE | | Exekvera program trace | ↑P | | |
| | PTRAEXEP | | | ↑P | | |

## 6.3.11 Teckensträngsinstruktioner

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Omvandling av binära tal till decimal-värden i ASCII kod | BINDA | S, D | Instruktionen BINDA omvand-lar ett tal som anges i S till ett femsiffrigt decimalt tal i ASCII-kod och lagrar det i registret som anges i D. | → | 3 | * |
| | BINDAP | | | ↑P | | |
| | DBINDA | | Instruktionen DBINDA omvandlar tal som anges i S till ett tiosiff-rigt decimalt tal i ASCII-kod och lagrar det i registret som anges i D. | → | | |
| | DBINDAP | | | ↑P | | |
| Omvandling av binära tal hexadecimal till ASCII kod | BINHA | S, D | Instruktionen BINHA omvand-lar ett tal som anges i S till ett fyrsiffrigt hexa-decimalt tal i ASCII-kod och lagrar det i registret som anges i D. | → | 3 | * |
| | BINHAP | | | ↑P | | |
| | DBINHA | | Instruktionen DBINHA omvandlar ett tal som anges i S till ett åtta-siffrigt hexade-cimalt tal i ASCII-kod och lagrar det i registret som anges i D. | → | | |
| | DBINHAP | | | ↑P | | |

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Omvandling av BCD data till ASCII kod | BCDDA | S, D | Instruktionen BCDDA omvandlar ett fyrsiffrigt BCD-kodat tal som anges i S till ett tal i ASCII-kod och lagrar det i registret som anges i D. | → | 3 | * |
| | BCDDAP | | | ↑P | | |
| | DBCDDA | | Instruktionen DBCDDA omvandlar ett åttasiffrigt BCD-kodat tal som anges i S till ett tal i ASCII-kod och lagrar det i registret som anges i D. | → | | |
| | DBCDDAP | | | ↑P | | |
| Omvandling av decimal ASCII till binära tal | DABIN | S, D | Instruktionen DABIN omvandlar ett femsiffrigt decimalt tal i ASCII-kod som anges i S till ett tal i binär form och lagrar det i registret som anges i D | → | 3 | * |
| | DABINP | | | ↑P | | |
| | DDABIN | | Instruktionen DDABIN omvandlar ett tiosiffrigt decimalt tal i ASCII-kod som anges i S till ett tal i binär form och lagrar det i registret som anges i D. | → | | |
| | DDABINP | | | ↑P | | |

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Omvandling av hexadecimal ASCII till binär data | HABIN | S, D | Instruktionen HABIN omvandlar ett fyrsiffrigt hexadecimalt tal i ASCII-kod som anges i S till ett tal i binär form och lagrar det i registret som anges i D. | → | 3 | * |
| | HABINP | | | ↑P | | |
| | DHABIN | | Instruktionen DHABIN omvandlar ett åttasiffrigt hexadecimalt tal i ASCII-kod som anges i S till ett tal i binär form och lagrar det i registret som anges i D. | → | | |
| | DHABINP | | | ↑P | | |
| Omvandling av ASCII till BCD-data | DABCD | S, D | Instruktionen DABCD omvandlar ett decimalt tal i ASCII-kod som anges i S till ett fyrsiffrigt tal i BCD-kod och lagrar det i registret som anges i D. | → | 3 | * |
| | DABCDP | | | ↑P | | |
| | DDABCD | | Instruktionen DDABCD omvandlar ett decimalt tal i ASCII-kod som anges i S till ett åttasiffrigt tal i BCD-kod och lagrar det i registret som anges i D. | → | | |
| | DDABCDP | | | ↑P | | |

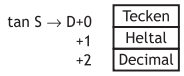
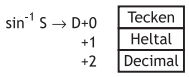
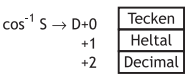
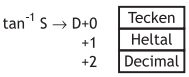
| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Utmatning av kommentars-data | COMRD | S, D | Instruktionen COMRD läser benämningar från registret som anges i S och lagrar det i ASCII-kod i arean D | → | 3 | * |
| | COMRDP | | | ↑P | | |
| Mät tecken-längd | LEN | | Instruktionen LEN mäter längden på teckensträngen som anges i S och lagrar resultatet i registret som anges i D | → | 3 | |
| | LENP | | | ↑P | | |
| Omvandling av binärt tal till tecken-sträng | STR | S1, S2, D | Instruktionen gör om ett binärvärde som anges i S2 till ett decimaltal i ASCII-format. Antal tecken och decimaler anges S1 och S1+1. Resultatet lagras i D och framåt | → | 4 | * |
| | STRP | | | ↑P | | |
| | DSTR | | | → | | |
| | DSTRP | | | ↑P | | |
| Omvandling av tecken-sträng till binär data | VAL | S, D1, D2 | Instruktionen VAL omvandlar teckensträngen i arean S till tal i binär form. Antalet tecken, decimaler och det binära värdet lagras i D1 och D2. | → | 4 | * |
| | VALP | | | ↑P | | |
| | DVAL | | Instruktionen DVAL omvandlar teckensträngen i arean S till tal i binär form. Antalet tecken, decimaler och det binära värdet lagras i D1 och D2. | → | | |
| | DVALP | | | ↑P | | |
| Omvandling av flyttal till teckendata | ESTR | S1, S2, D | Samma funktion som STR fast för flyttalsvärden. | → | 4 | * |
| | ESTRP | | | ↑P | | |
| Omvandling av tecken-sträng till decimalt flyttal | EVAL | S, D | Samma funktion som VAL fast för flyttalsvärden. | → | 3 | |
| | EVALP | | | ↑P | | |

| Användning | Namn | Variabler | Funktion | Exekvering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Omvandling av 16 bit data till ASCII tecken | ASC<br>ASCP | S, n, D | Instruktionen ASC omvandlar talet lagrat i S och framåt till ett hexadecimalt tal i ASCII-kod. Antalet tecken som skall omvandlas anges i n. Resultatet lagras i D och framåt. | →<br>↑P | 4 | *8.24.1* |
| Omvandling av hexadecimal ASCII värden till binära värden | HEX<br>HEXP | S, n, D | Instruktionen HEX omvandlar tecken i hexadecimal ASCII-kod i S och framåt till binärt värde. Antalet tecken som ska omvandlas anges i n. Resultatet lagras i D och framåt. | →<br>↑P | 4 | *8.24.2* |
| Uppackning av teckensträng (högra delen av teckensträngen) | RIGHT<br>RIGHTP | S, n, D | Instruktionen RIGHT lagrar n tecken från höger i teckensträngen (slutet av teckensträngen) i S. Tecknen lagras i D. | →<br>↑P | 4 | * |
| Uppackning av teckensträng (vänstra delen av teckensträngen) | LEFT<br>LEFTP | S, n, D | Instruktionen LEFT lagrar n tecken från vänster i teckensträngen (början av teckensträngen) i S. Tecknen lagras i D. | →<br>↑P | | |
| Flytta delar från en teckensträng till en annan teckensträng | MIDR<br>MIDRP | S1, S2, D | MIDR-instruktionen lagrar S2+1 tecken i D ur strängen S1 med start på S2. | →<br>↑P | | |
| | MIDW<br>MIDWP | S1, S2, D | MIDW-instruktionen lagrar S2+1 tecken ur strängen i S1 med start på S2 i D och framåt. | →<br>↑P | | |

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Sökning efter teckensträng | INSTR<br>INSTRP | S1, S2, n, D | Intruktionen INSTR söker ett tecken definerad i S1, inuti en teckensträng definierad i S2. Sökningen börjar med ett tecken definierat i n. | →<br>↑P | 5 | * |
| Flyttalsformat omvandling med BCD-representation | EMOD<br>EMODP | S1, S2, D1 | EMOD-instruktionen beräknar BCD formatet ur ett flyttal i S1 som tar hänsyn till decimalkommat definierat i S2. Resultatet lagras i D1. | →<br>↑P | 4 | * |
| BCD-dataomvandling med decimalt flyttalsformat | EREXP<br>EREXPP | S1, S2, D1 | EREXP-instruktionen beräknar decimalerna ur ett flyttal i BCD format i S1 som tar hänsyn till decimalkommat placering definierat i S2.Resultatet lagras i D1 | →<br>↑P | 3 | * |

# 6.3.12 Specialfunktionsinstruktioner

| Användning | Namn | Variabler | Funktion | Exekvering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Sinusberäkning | SIN<br>SINP | S, D | SIN(S+1, S) → (D+1, D) | →<br>↑P | 3 | *8.25.1* |
| Cosinusberäkning | COS<br>COSP | S, D | COS(S+1, S) → (D+1, D) | →<br>↑P | 3 | *8.25.2* |
| Tangentberäkning | TAN<br>TANP | S, D | TAN(S+1, S) → (D+1, D) | →<br>↑P | 3 | *8.25.3* |
| Arcussinusberäkning | ASIN<br>ASINP | S, D | $ASIN^{-1}$(S+1, S) → (D+1, D) | →<br>↑P | 3 | *8.25.4* |
| Arcuscosinusberäkning | ACOS<br>ACOSP | S, D | $ACOS^{-1}$(S+1, S) → (D+1, D) | →<br>↑P | 3 | *8.25.5* |
| Arcustangens beräkning | ATAN<br>ATANP | S, D | $ATAN^{-1}$(S+1, S) → (D+1, D) | →<br>↑P | 3 | *8.25.6* |
| Omvandling från grader till radianer | RAD<br>RADP | S, D | (S+1, S) → (D+1, D) | →<br>↑P | 3 | *8.25.7* |
| Omvandling från radianer till grader | DEG<br>DEGP | S, D | (S+1, S) → (D+1, D) | →<br>↑P | 3 | *8.25.8* |
| Kvadratrotsutdragning | SQR<br>SQRP | S, D | $\sqrt{(S+1, S)} \rightarrow (D+1, D)$ | →<br>↑P | 3 | *8.25.9* |
| Flyttalsvärde som exponent till e | EXP<br>EXPP | S, D | $e^{(S+1, S)} \rightarrow (D+1, D)$ | →<br>↑P | 3 | *8.25.10* |
| Logaritm | LOG<br>LOGP | S, D | LOG e (S+1, S) → (D+1, D) | →<br>↑P | 3 | *8.25.11* |
| Slumpvärde | RND<br>RNDP | D | Lagrar slumptal i D | →<br>↑P | 3 | *8.25.12* |
| Uppdaterar slumpvärden | SRND<br>SRNDP | S | Uppdaterar slumptal i S | →<br>↑P | 3 | |
| Kvadratrots beräkning ur 4-siffrigt BCD-värde | BSQR<br>BSQRP | S, D | $\sqrt{S} \rightarrow$ D+0 (Heltal), +1 (Decimal) | →<br>↑P | 3 | * |
| Kvadratrotsberäkning ur 8-siffrigt BCD- värde | BDSQR<br>BDSQRP | S, D | $\sqrt{S+1, S} \rightarrow$ D+0 (Heltal), +1 (Decimal) | →<br>↑P | 3 | * |
| Sinusberäkning ur BCD- värde | BSIN<br>BSINP | S, D | sin S → D+0 (Tecken), +1 (Heltal), +2 (Decimal) | →<br>↑P | 3 | * |
| Cosinusberäkning ur BCD- värde | BCOS<br>BCOSP | S, D | cos S → D+0 (Tecken), +1 (Heltal), +2 (Decimal) | →<br>↑P | 3 | * |

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Tangens-beräkning ur BCD-värde | BTANP<br>BTANP | S, D |  | →<br>↑P | 3 | * |
| Arcussinus beräkning ur BCD-värde | BASIN<br>BASINP | S,D |  | ↑P<br>→ | 3 | * |
| Arcuscosinus-beräkning ur BCD-värde | BACOS<br>BACOSP | S,D |  | ↑P<br>→ | 3 | * |
| Arcustangens beräkning ur BCD-värde | BATAN<br>BATANP | S,D |  | ↑P | 3 | * |

## 6.3.13 Datakontrollinstruktioner

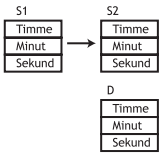
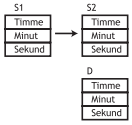
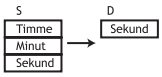
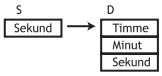
| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Övre och lägre gräns-kontroll för binära vär-den | LIMIT<br>LIMITP | S1, S2, S3, D | Begränsning av utgångsvärden för 16/32-bitarsbinära värden | →<br>↑P | 5 | *8.26.1* |
| | DLIMT<br>DLIMITP | S1, S2, S3, D | | →<br>↑P | | |
| Dödbands-kontroll för binära vär-den | BAND<br>BANDP | S1, S2, S3, D | Instruktionen skapar en död-bandskontroll av 16/32-bitars-binära värden | →<br>↑P | 5 | *8.26.2* |
| | DBAND<br>DBANDP | S1, S2, S3, D | | →<br>↑P | | |
| Zonkontroll för binära värden | ZONE<br>ZONEP | S1, S2, S3, D | Instruktionen gör en zonkon-troll av 16/32-bitarsbinära värden | →<br>↑P | 5 | *8.26.3* |
| | DZONE<br>DZONEP | S1, S2, S3, D | | →<br>↑P | | |

## 6.3.14 Växlingsinstruktioner för filregister

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Växlings-instruktion för filregis-terblock | RSET<br>RSETP | S | RSET-instruktio-nen växlar från ett filregister-block till ett annat filregis-terblock med adress definie-rat i S | →<br>↑P | 2 | *8.27.1* |

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Växlings-instruktion för filregis-terfiler | QDRSET<br>QDRSETP | S | QDRSET-instruk-tionen växlar från en filregis-terfil som används av ett program till en definierad av S | →<br>↑P | 2 | * |
| Växlings-instruktion för kommen-tarfiler | QCDSET<br>QCDSETP | S | QCDSET-instruk-tionen växlar från kommen-tarfilen som används till en kommentarsfil definierad i S. | →<br>↑P | | * |

## 6.3.15 Klockinstruktioner

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Läs realtids-klockan | DATERD<br>DATERDP | D | | →<br>↑P | 2 | * |
| Skriva till realtids-klockan | DATEWR<br>DATEWRP | S | | →<br>↑P | 2 | * |
| Addera klockdata | DATE+<br>DATE+P | S1, S2, D |  | →<br>↑P | 4 | * |
| Subtrahera klockdata | DATE-<br>DATE-P | S1, S2, D |  | →<br>↑P | 4 | * |
| Ändra klock-dataformatet från hh:mm:ss till sekunder | SECOND<br>SECONDP | S,D |  | →<br>↑P | 3 | * |
| Ändra klock-dataformatet från sekunder till hh:mm:ss | HOUR<br>HOURP | S,D |  | →<br>↑P | | |

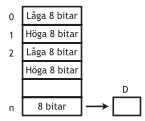
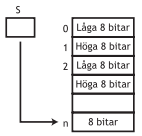
## 6.3.16 Yttre operandinstruktioner

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Utgångsmeddelande till yttre enhet | MSG | S | MSG-instruktionen lägger ut en teckensträng lagrad i S till en terminal eller en display. | → | 2 | * |
| Ingångsmeddelande från yttre enhet | PKEY | D | PKEY instruktionen läser tecken i en terminal och skriver detta I ASCII-format till D | → | 2 | * |

## 6.3.17 Programinstruktioner

| Användning | Namn | Variabler | Funktion | Exekv-ering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Växla programdel till Stand-by | PSTOP<br>PSTOPP | S | PSTOP instruktionen sätter programdelen, specificerad i S, i Stand-by läge. | →<br>↑P | 2 | * |
| Växla programdel till Stand-by och återställ utgångarna | POFF<br>POFFP | S | POFF instruktionen sätter programdelen, specificerad i S, i Stand-by läge och återställer utgångarna. | →<br>↑P | 2 | * |
| Växlar programdel till exekveringsläge | PSCAN<br>PSCANP | S | PSCAN instruktionen sätter programdelen specificerad i S i scanläge, varvid programdelen utörs varje programvarv. | →<br>↑P | 3 | * |
| Växla programdel till låghastighets exekveringläge | PLOW<br>PLOWP | S | PLOW instruktionen sätter programdelen specificerad i S i låghastighet exekveringsläge | →<br>↑P | 3 | * |

## 6.3.18 Andra instruktioner

| Användning | Namn | Variabler | Funktion | Exekvering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Reset watchdog timer | WDT<br>WDTP | | WDT instruktionen återställer WDT under exekveringen av ett program | →<br>↑P | 1 | *8.28.1* |
| Egendefinierat pulståg | DUTY | n1, n2, D | n1 och n2 styr hur många programvarv specialminnescellen D skall vara till resp. från | ↑P | 4 | *8.28.2* |
| Läs en byte från filregister | ZRRDB<br>ZRRDBP | n, D |  | →<br>↑P | 3 | * |
| Skriv en byte till filregister | ZRWRB<br>ZRWRBP | n, S |  | →<br>↑P | 3 | * |
| Numerisk inmatning från tangentbord | KEY | S, n D1, D2 | Tangentbordsinstruktionen KEY stöder inmatningen av 8 st ASCII tecken i S (X). Det inmatade värdet hexadecimalkodas och lagras i operanden D1. | → | 5 | * |
| Lagring av indexregister | ZPUSH<br>ZPUSHP | D | ZPUSH instruktionen spara innehållet i indexregistren Z0 till Z15 i D. | →<br>↑P | 3 | * |
| Återställning av indexregister | ZPOP<br>ZPOPP | D | ZPOP instruktionen återställer innehållet i till indexregistren Z0 Z15. | →<br>↑P | 3 | * |
| Skriver till EEPROM | EROMWR<br>EROMWRP | S, n D1, D2 | EROMWR instruktionen skriver ett antal dataord definierat av n lagrat i S till ett EEPROM filregister definierat av D1. | →<br>↑P | 6 | * |

# 6.4 Datalänkinstruktioner

## 6.4.1 Nätverksinstruktioner

| Användning | Namn | Variabler | Funktion | Exekvering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Nätverkinstruktioner | ZCOM | Jn<br>Un | Datauppdatering i nätverksmodulerna | → | 6 | * |

## 6.4.2 Dedicerade datalänk instruktioner

| Användning | Namn | Variabler | Funktion | Exekvering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Dedicerade länkinstruktioner | READ | Jn, S1, S2, D1, D2<br>Un, S1, S2, D1, D2 | Läser ordoperander från en nätverksstation. | → | 11 | * |
| | SREAD | Jn, S1, S2, D1, D2, D3<br>Un, S1, S2, D1, D2, D3 | | | 13 | |
| | WRITE | Jn, S1, S2, D1, D2<br>Un, S1, S2, D1, D2 | Skriver ordoperander till en nätverksstation. | → | 12 | |
| | SWRITE | Jn, S1, S2, D1, D2, D3<br>Un, S1, S2, D1, D2, D3 | | | 13 | |
| | SEND | Jn, S1, S2, D<br>Un, S1, S2, D | Sänder data till en nätverksstation | → | 10 | |
| | RECV | Jn, S, D1, D2<br>Un, S, D1, D2 | Mottager data från andra nätverksstationer. | → | 9 | |
| | REQ | Jn, S1, S2, D1, D2<br>Un, S1, S2, D1, D2 | Begär data från funktionsmoduler i fjärrnätverksstationer. | → | 10 | |

### 6.4.3 A-serien kompatibla datalänkinstruktioner

| Användning | Namn | Variabler | Funktion | Exekvering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Dedicerade länkinstruktioner | ZNFR | Jn, S1, S2, D<br>Un, S1, S2, D | Läser data från funktionsmoduler i nätverksstationer | → | 9 | * |
| | ZNTO | Jn, S1, S2, D<br>Un, S1, S2, D | Skriver data till funktionsmoduler i nätverksstationer | → | 9 | |
| A-serien, kompatibla länkinstruktioner | J.ZNRD | Jn, n1, S, n2, D1, D2 | Läser QnA data från stationer i nätverk. | →<br>↑P | 12 | |
| | JP.ZNRD | | | | | |
| | J.ZNWR | Jn, n1, S, n2, D1, D2 | Skriver QnA data till stationer i nätverk | →<br>↑P | 12 | |
| | JP.ZNWR | | | | | |
| | RFRP | n1, n2, n3, D | Läser data från funktionsmodul i en fjärrstation | → | 9 | |
| | G.RFRP | Un, n1, n2, D1, D2 | | | | |
| | RTOP | S, n1, n2, n3 | Skriver data till funktionsmodul i en fjärrstation | → | 9 | |
| | G.RTOP | Un, n1, S, n2, D1 | | | | |

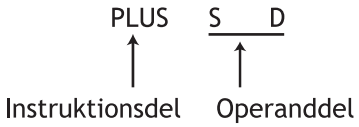
### 6.4.4 Läs/Skriv routing information

| Användning | Namn | Variabler | Funktion | Exekvering | Antal steg | Ref. |
|---|---|---|---|---|---|---|
| Läs/Skriv routing information | Z.RTREAD | n, D | RTREAD instruktionen läser routing information från destinationsnätverket definierat i n. Readrouting information är lagrad i D | →<br>↑P | 7 | * |
| | ZP.RTREAD | | | | | |
| | Z.RTWRITE | S, n | RTWRITE instruktionen skriver routing information till destinationsnätverket definierat i n. Readrouting information är lagrad i S. | →<br>↑P | 8 | |
| | ZP.RTWRITE | | | | | |

# 7 Instruktionernas uppbyggnad

## 7.1 Instruktionsstruktur

De flesta instruktionerna består av en instruktionsdel och en operanddel. Övriga instruktioner kräver ingen operanddel och består därmed endast av en instruktionsdel.




### Instruktionsdel

Instruktionsdelen beskriver funktionen.

PLUS = Addition

### Operanddel

Operanddelen beskriver konstanterna eller variablerna som måste anges. Operanddelen kan bestå av tre delar, källa (S), destination (D) och antal (n).

## 7.1.1 Källa (S)

– Källdatan anger operanderna som skall bearbetas av instruktionen.
  För enkelordsinstruktioner är källdatan S (16 bitar).
  För dubbelordsinstruktioner är källdatan S+1 och S (32 bitar).

– Som källdata kan konstanter eller variabler anges.

### Konstanter

Konstanter specificerar ett numeriskt värde som instruktionen skall bearbeta. Värdet anges av programmeraren i programmet och kan inte ändras under programbearbetningen.

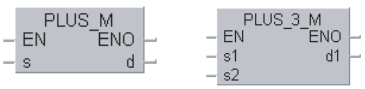
### Variabler

Variabler specificerar en operand som innehåller data som skall bearbetas av instruktionen (*se kapitel 7.4*)
Innan en instruktion bearbetas, måste data lagras i operanden. Datan kan ändras under programbearbetning.

## 7.1.2 Destination (D)

– Destinationsdatan anger operanderna där data skall lagras efter instruktionen bearbetats.
För enkelordsinstruktioner är destinationen D (16 bitar).
För dubbelordsinstruktioner är destinationen D+1 och D (32 bitar).
I vissa fall, i instruktioner med endast två operander, kan destinationen ingå som en del av källan. Då kommer resultatet att lagras i samma variabel.

Exempel:
Binär addition, enkelordsinstruktion. D innehåller källdata för additionen i före programbearbetning, efteråt lagras resultatet i D.




S + D = D S1 + S2 = D

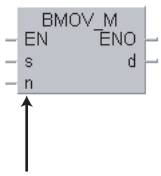
– Operander för datalagring (destination) måste anges som en variabel.

## 7.1.3 Antal (n)

– Antal (n) specificerar hur många operander som skall användas eller hur många gånger instruktionen skall exekveras.

Exempel:

BMOV-instruktion för blockförflyttning




Specificerar antalet register som
skall överföras med BMOV-instruktionen

Värdet n skall vara inom området 0 till 32 767. Om n är 0 kommer instruktionen ej utföras.

# 7.2 Instruktionernas skrivsätt

Instruktionernas funktioner kan utläsas av dess skrivsätt:

## 7.2.1 Enkel- eller dubbelord och pulsfunktion

| | |
|---|---|
| SORT | Enkelordsbearbetning |
| SORTP | Enkelordsbearbetning med pulsfunktion |
| DSORT | Dubbelordsbearbetning |
| DSORTP | Dubbelordsbearbetning med pulsfunktion |

## 7.2.2 MELSEC och IEC

GX IEC Developer innehåller flera olika programeditorer.

### Texteditorer

– MELSEC instruktionslista (MELSEC IL)

– IEC instruktionslista (IL)

– Strukturerad text (ST)

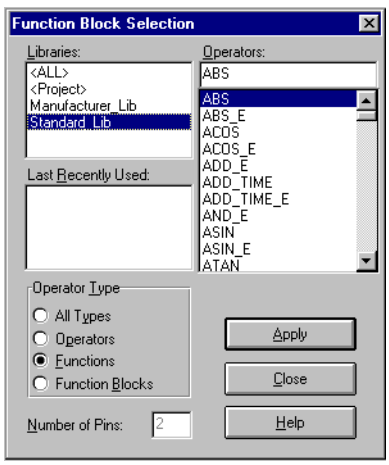
### Grafiska editorer

– Reläschema (IEC) (LD)

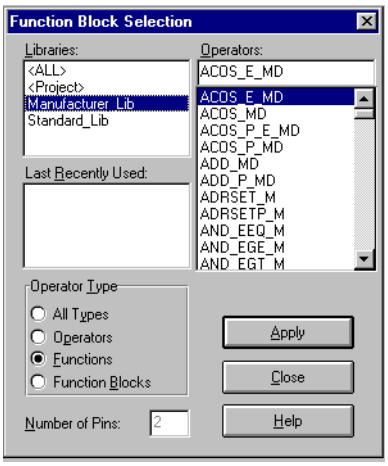
– Funktionsblocksdiagram (IEC) (FBD)

– Sekvensdiagram (IEC) (SFC)

I respektive editor representeras instruktionerna med olika notation.




*När en instruktion skall väljas i GX IEC Developer, visas denna dialogruta där Standard_Lib är valt. Listan innehåller de standardiserade IEC-instruktionerna.*




*När en instruktion skall väljas i GX IEC Developer, visas denna dialogruta där Manufacturer_Lib är valt. Listan innehåller de "rena" MELSEC-instruktionerna.*

Hanteringen av standardinstruktionerna och MELSEC-instruktionerna är den samma, endast skrivsättet skiljer sig.

## Förklaring av tillägg i IEC-editorerna:

| Tillägg i IEC-editor | Betydelse |
|---|---|
| _M | MELSEC-instruktion |
| _P_M | MELSEC-instruktion med pulsfunktion |
| _MD | Utökad MELSEC-instruktion |
| _P_MD | Utökad MELSEC-instruktion med pulsfunktion |
| _K_MD | Utökad MELSEC-instruktion med konstant |
| _K_P_MD | Utökad MELSEC-instruktion med konstant och pulsfunktion |

## 7.2.3 Ytterligare beskrivning av skrivsättet

Tabellen nedan innehåller symboler vilka representerar ett flertal funktioner i en MELSEC-editor. Kolumnen till höger visar motsvarande benämning i en IEC-editor.

Exempel:

| MELSEC-editor | IEC-editor |
|---|---|
| LD$> | LD_STRING_GT_M |

| MELSEC-editor | IEC-editor |
|---|---|
| $ | STRING |
| = | EQ |
| <> | NE |
| <= | LE |
| < | LT |
| >= | GE |
| > | GT |
| + | PLUS |
| - | MINUS |
| x | MULTI |
| / | DIVID |

## 7.2.4 Specifikation av skrivsättet

Kapitel 5 till 8 ger en detaljerad beskrivning av instruktionerna, och illustrerar båda editorerna, d v s båda skrivsätten. Rubriken innehåller den rena MELSEC-instruktionen som den uppträder i MELSEC-instruktionslistan.

**Observera!**
Tabellöversikten i början av varje kapitel beskriver båda skrivsätten.

# 7.3 Programmering av variabler

De flesta instruktionerna behöver, förutom en instruktionsdel, också en operanddel med specificerade variabler. Variablerna innehåller värden som används vid bearbetningen av instruktionen.

Beroende på vilken editor som valts i GX IEC Developer, programmeras variablerna på olika sätt.

### I en MELSEC-editor:

Dataregister kan anges direkt som hårdvaruadresser t ex D100 och D10. Det anslutna PLC-systemet känner själv av att följande operander skall användas:

D100 = D100 och D101 (vid en dubbelordsinstruktion med dataförflyttning)
D10 = D10, D11, D12, D13 (vid t ex en dubbelordsmultiplikation)

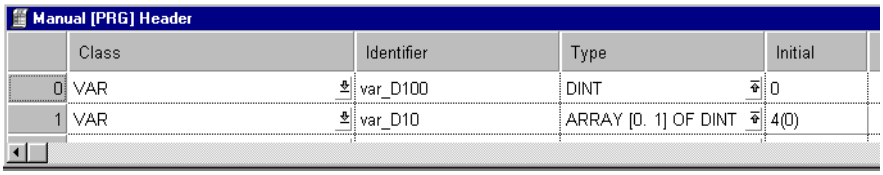
### I en IEC-editor:

I en IEC-editor kan hårdvaruadresser endast anges om enbart den angivna operanden skall användas.

Exempel: AND D10

Innan t ex instruktionen DDIVIDP_M kan bearbetas, måste variablerna som skall användas definieras i programdelens (POU) header.

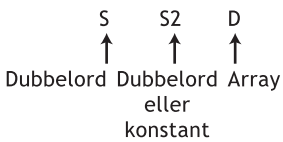
Exempel:




var_D100 och var_D10 är definierade som benämningar. PLC-systemet kommer inte använda D100 och D10 utan allokerar internt lediga dataregister till variablerna.

Exempel:

```
DDIVIDP_M var_D100, 4, var_D10
```




Variabeln var_D100 är definierad som en DINT (dubbelord), variabeln var_D10 är av typen Array (vektor). Vektorn innehåller två stycken register av typen DINT (dubbelord).

## Specifikation av skrivsättet

Definieringen av var_D100 och var_D10 i skärmdumparna indikerar att hårdvaruadresser inte kan användas i intruktionen. I dessa fall är variabeldefinition obligatorisk! Om en instruktion endast kan programmeras med variabler kommer manualen upplysa om detta.

**Observera!**
En benämning kan anta vilket namn som helt (t ex Motor_1, Indikering). Namnen var_D100 och var_D10 valdes för att ge en klar referens till programmeringen i MELSEC-editorn.

Tabellen över variabler i början av varje instruktionsbeskrivning ger en översikt på de datatyper operanderna kan anta för respektive instruktion *(exemplet visar DDIVID-instruktionen, se kapitel 8.10.4).*

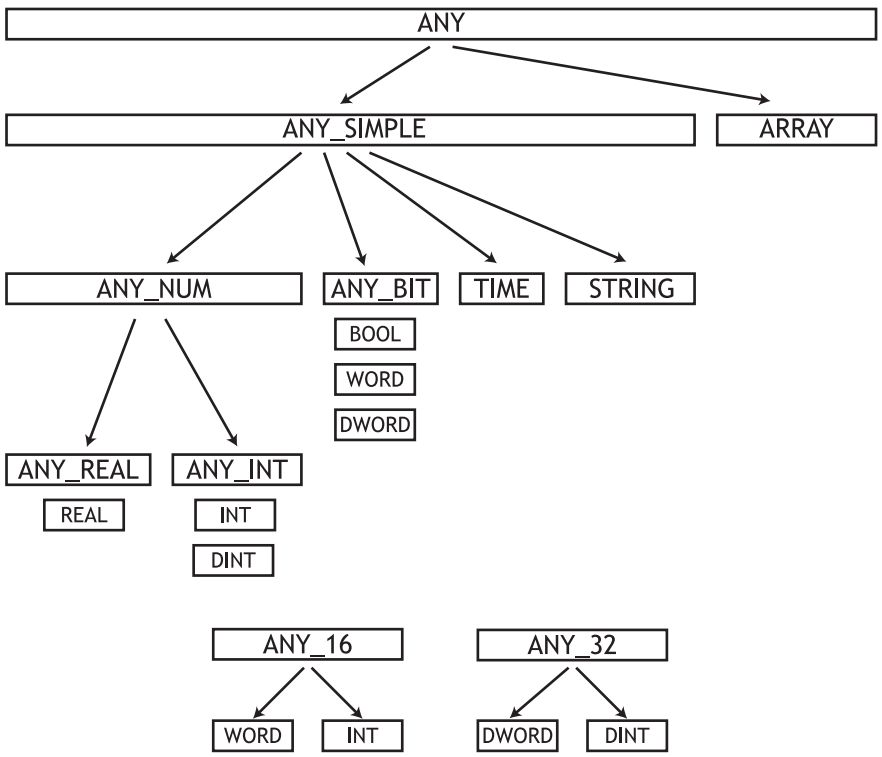
| Operander | Beskrivning | Datatyp | |
|---|---|---|---|
| | | MELSEC | IEC |
| S1 | Första operanden med data för täljaren | BIN 32-bit | ANY 32-bit |
| S2 | Första operanden med data för nämnaren | BIN 32-bit | ANY 32-bit |
| D1 | Första operanden där lagring sker | BIN 64-bit | Array [1..2] of ANY 32-bit |

# 7.4 Datatyper

Datatypen bestämmer adressen och behandlingen av bitar och variabelområden. Följande datatyper finns:

| Datatyp | | Område | Antal bitar |
|---|---|---|---|
| BOOL | Boolean | 0 (FALSE), 1 (TRUE) | 1 bit |
| INT | Heltal (Integer) | -32 768 - 32 767 | 16 bitar, med teckenbit |
| DINT | Dubbelt heltal (Integer) | -2 147 483 648 - 2 147 483 647 | 32 bitar, med teckenbit |
| WORD | 16 bitar | 0 - 65 535 | 16 bitar, utan teckenbit |
| DWORD | 32 bitar | 0 - 4 294 967 295 | 32 bitar, utan teckenbit |
| REAL | Flyttal | 3,4 +/- 38 (7 tecken) | 32 bitar |
| TIME | Tidvärde | T#24d00h31m23s648,00ms - T#24d20h31m23s647ms | 32 bitar |
| STRING | Teckensträng | max 50 tecken | |

## Datatypernas hierarki



| Datatyp | Beskrivning |
|---|---|
| ANY | Alla datatyper |
| ANY_SIMPLE | Alla enkla datatyper |
| ANY_NUM | Alla numeriska datatyper |
| ANY_REAL | Alla flyttalsdatatyper |
| ANY_INT | Alla heltalsdatayper |
| ANY_BIT | Alla bitar |
| ANY_16 | Alla enkelord |
| ANY_32 | Alla dubbelord |
| TIME | Tid |
| STRING | Teckensträng |
| REAL | Flyttal |
| INT | Enkelt heltal (16 bitar, med teckenbit) |
| DINT | Dubbelt heltal (32 bitar, med teckenbit) |
| BOOL | Bit |
| WORD | Enkelord (16 bitar, utan teckenbit) |
| DWORD | Dubbelord (32 bitar, utan teckenbit) |
| ARRAY | Vektor |

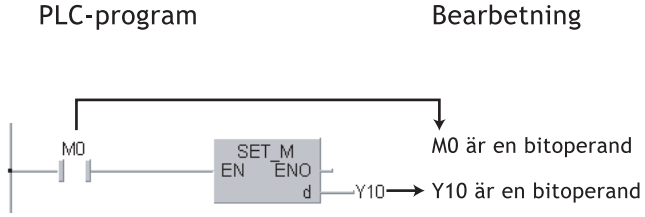
## 7.4.1 Databehandling

# Behandling av bitar

En bitoperand (X, Y, M, K, S, B eller F) kan anta två tillstånd (TILL=1 eller FRÅN=0). Dess status kan därför representeras av en bit (1 eller 0). Bitexekvering utförs alltid om en bitoperand är adresserad i programmet. För att exekvera enkel- eller dubbelordsinstruktioner måste flera bitoperander grupperas i block om 16 eller 32 adresser.

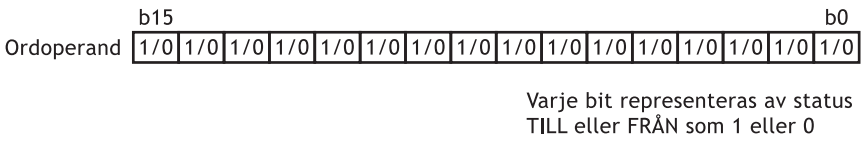
### Att använda bitoperander

En bitoperand (t ex ingångar, utgångar, minnesceller) består av en bit.

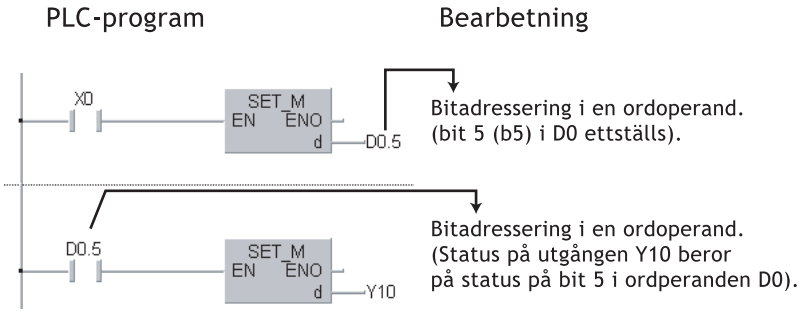



### Att använda ordoperander

Man kan adressera de enskilda bitarna i en ordoperand.



Bitarna måste adresseras i hexadecimalt format. T ex, bit 5 (b5) i D0 adresseras som D0.5. Bit 10 (b10) i D0 adresseras som D0.A. Bitar i tidskretsar, räknare och ackumulerande tidskretsar kan inte adresseras.




### Att använda bitar i block

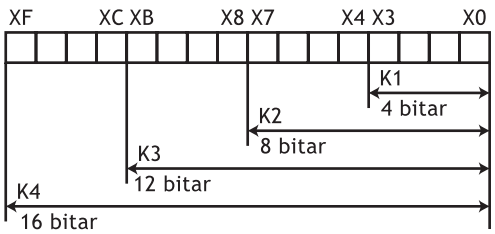
Enstaka bitar kan grupperas i block om fyra, och på så sätt exekveras som ordoperander. En detaljerad beskrivning av detta finns i följande kapitel.

## Behandling av orddata (enkelord)

### Att använda bitoperander

Bitoperander kan exekveras som ordoperander under förutsättning att antalet bitadresser som skall användas, definieras. Upp till 16 bitar kan exekveras i block om 4. Längden på varje block bestäms av konstanterna K1 till K4.

K1X0 4 adresser från X0 till X3
K2X0 8 adresser från X0 till X7
K3X0 12 adresser från X0 till XB
K4X0 16 adresser från X0 till XF

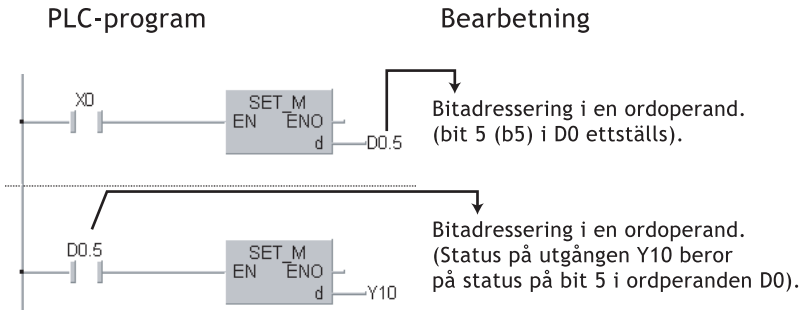


## Definiering av bitblock för S

Tabellen nedan visar värdena som kan användas som källdata för bitadressering av källdatan (S).

| Bitadressering | 16-bitarsinstruktion |
|---|---|
| K1 (4 bitar) | 0 - 15 |
| K2 (8 bitar) | 0 - 255 |
| K3 (12 bitar) | 0 - 4095 |
| K4 (16 bitar) | -32 768 - 32 767 |

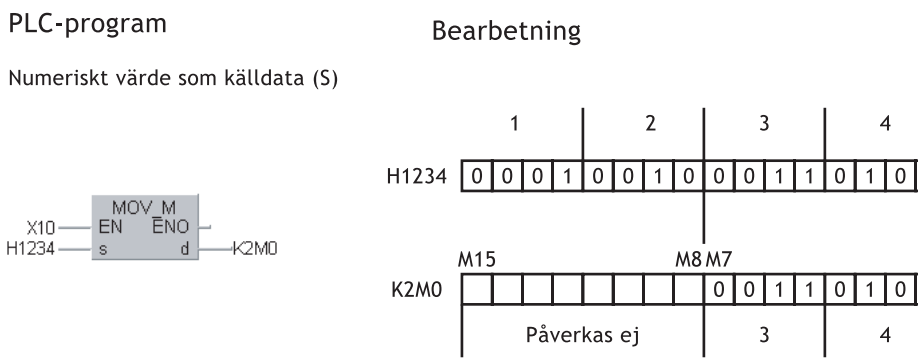
Bitadresserna som ej används sätts till 0.

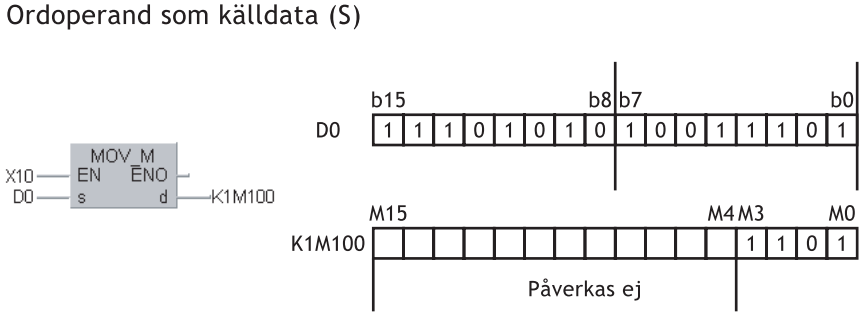



**Observera!**
Adressen till första biten i ett block kan sättas till valfri adress.

## Definiering av bitblock för D

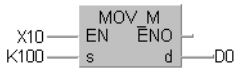
Bitadresseringen för destinationsdatan (D) bestämmer hur många adresser man skall skriva till. Bitadresserna utanför området påverkas ej.



### Att använda ordoperander

Ordoperander bestäms med en adress. Adressen består av 16 bitar.

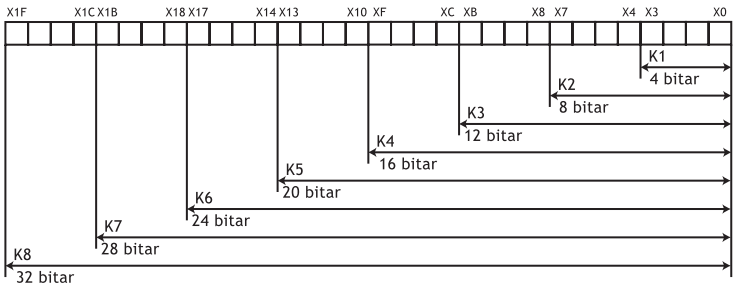



## Behandling av orddata (dubbelord)

### Att använda bitoperander

Bitoperander kan exekveras som ordoperander under förutsättning att antalet bitadresser som skall användas, definieras. Upp till 32 bitar kan exekveras i block om 4. Längden på varje block bestäms av konstanterna K1 till K8.

K1X0 4 adresser från X0 till X3
K2X0 8 adresser från X0 till X7
K3X0 12 adresser från X0 till XB
K4X0 16 adresser från X0 till XF
K5X0 20 adresser från X0 till X13
K6X0 24 adresser från X0 till X17
K7X0 28 adresser från X0 till X1B
K8X0 32 adresser från X0 till X1F

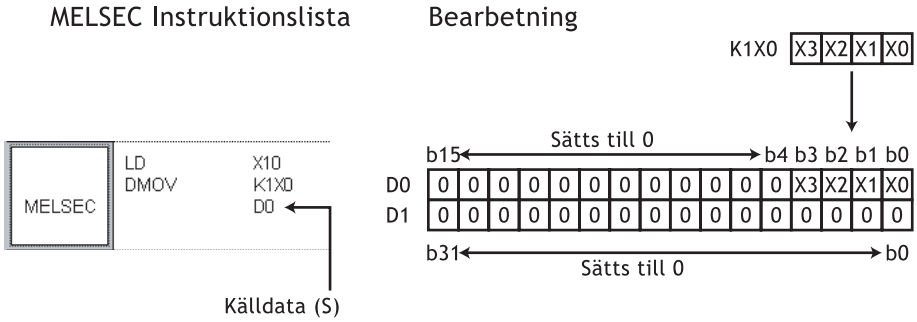



### Definiering av bitblock för S

Tabellen nedan visar värdena som kan användas som källdata för bitadressering av källdatan (S).

| Bitadressering | 16-bitarsinstruktion |
|---|---|
| K1 (4 bitar) | 0 - 15 |
| K2 (8 bitar) | 0 - 255 |
| K3 (12 bitar) | 0 - 4 095 |
| K4 (16 bitar) | 0 - 65 535 |
| K5 (20 bitar) | 0 - 1 048 575 |
| K6 (24 bitar) | 0 - 16 777 215 |
| K7 (28 bitar) | 0 - 33 554 432 |
| K8 (32 bitar) | -2 147 483 648 - 2 147 483 647 |

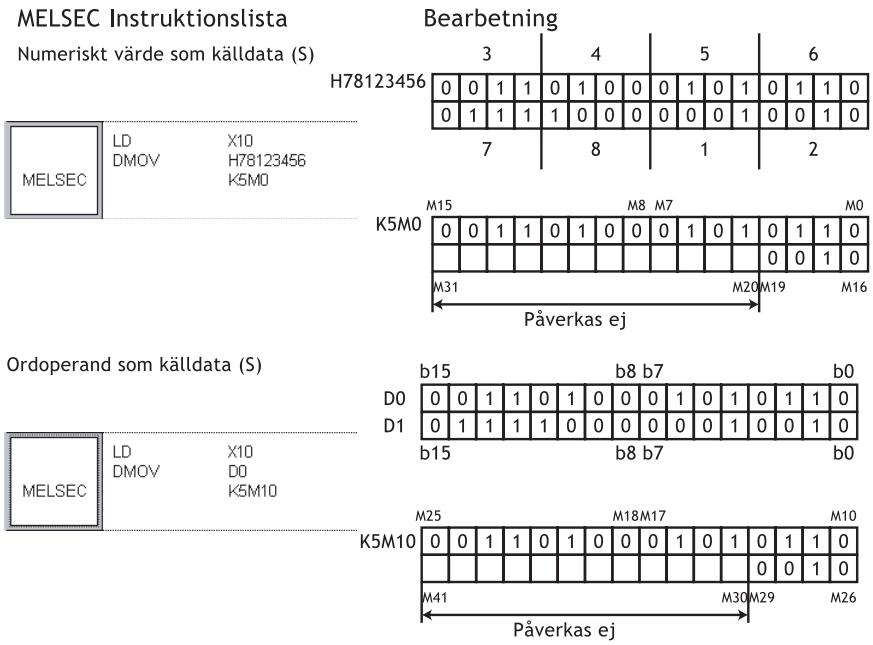
Bitadresserna som ej används sätts till 0.




**Observera!**
Adressen till första biten i ett block kan sättas till valfri adress.

## Definiering av bitblock för D

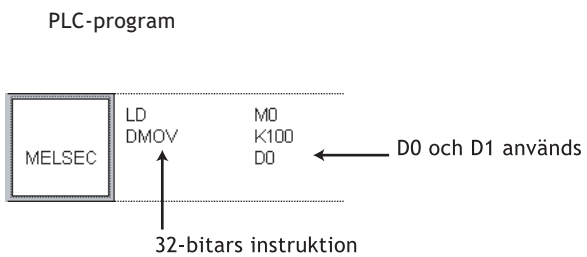
Bitadresseringen för destinationsdatan (D) bestämmer hur många adresser man skall skriva till. Bitadresserna utanför området påverkas ej.




## Att använda ordoperander

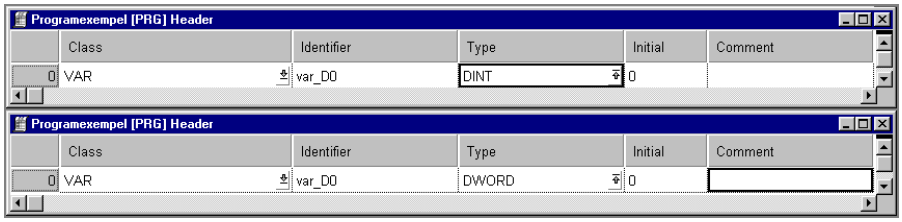
Dubbla ordoperander består av två 16-bitarsoperander, dessa programmeras olika beroende på vilken programmeringsmjukvara och programmeringseditor som används.

– GX Developer och MELSEC-editorn i GX IEC Developer

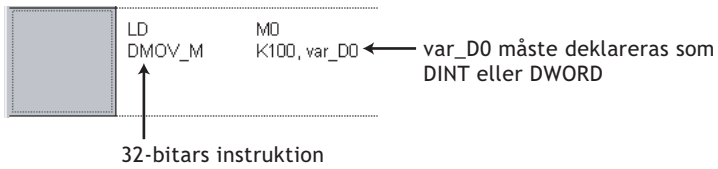


– I en IEC-editor i GX IEC Developer

Innan en dubbelordsoperand kan användas i en IEC-editor i GX IEC Developer måste en variabel definieras i programdelens header. Datatyperna DINT och DWORD är dubbelord (32 bitar).




PLC-program

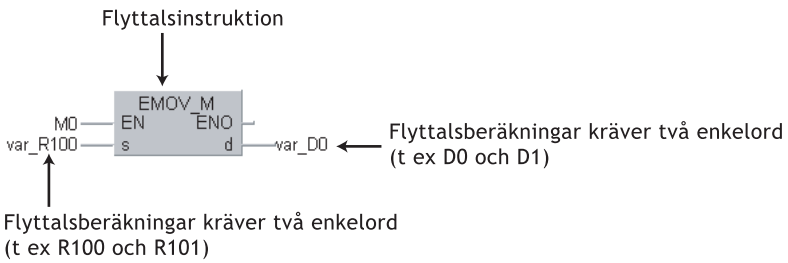



## 7.4.2 Flyttal

Genom att använda flyttalsinstruktioner kan beräkningar ge noggrannare resultat. Resultatet blir alltid ett dubbelord i flyttalsformat. Vid division erhålls inget restvärde. Med flyttalsberäkningar kan man enkelt använda ingenjörsstorheter direkt i programmet. Även trigonometriska beräkningar kan utföras.

### Behandling av flyttalsdata REAL

Operander av typen REAL är flyttalsvärden. Endast ordoperander kan lagra flyttalsvärden. Flyttalet lagras i två enkelordsregister och adresseras till registret med de lägsta 16 bitarna.

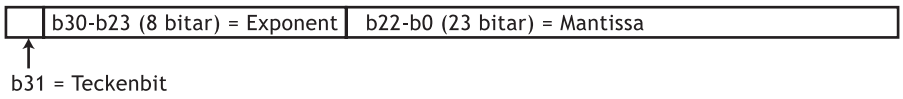



**Observera!**
Flyttalsinstruktioner börjar med ett E, t ex EMOV.

Två ordoperander behövs för att lagra ett flyttalsvärde. Därför delas flyttalet upp i följande delar:

Teckenbit ; $2^{[Exponent]}$ ; [Mantissa]

Bitkonfigurationen visas i bilden nedan:




Teckenbit: Teckenbiten lagras i b31.
0 = Positivt
1 = Negativt

Exponent:Värdet n i $2^n$ lagras binärt i b23 till b30. Betydelsen av n visas i bilden nedan.

| b30-b23 | HFF | HFE | HFD | H81 | H80 | H7F | H7E | H02 | H01 | H00 |
|---|---|---|---|---|---|---|---|---|---|---|
| n | Används ej | 127 | 126 | 2 | 1 | 0 | -1 | -125 | -126 | Används ej |

Exempel: Om binärkoden H81 lagras i b23 till b30 är n=2

Mantissa: Med 23 bitar från b0 till b22 kan ett flyttal visas med 6 siffrors noggrannhet (XXXXXX eller 1,XXXXXX).

**Observera!**
Q00(J) och Q01 revision A hanterar inte flyttal.
Q02 och större samt Q4AR utför flyttalsberäkningar snabbt medan övriga QnA(S)-system har 50-100 ggr längre exekveringstid då operanden är ett flyttal.
Flyttal avrundas till sju signifikanta siffror plus eventuell exponent. Det innebär att beräkningarna blir noggrannare, men ej exakta. Undvik därför:
- att göra jämförelser med tal med många decimaler.
- att addera/subtrahera ett mycket litet tal med ett mycket stort tal.

# 7.4.3 Vektorer och register i GX IEC Developer

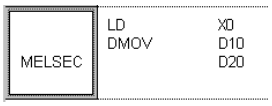
## Adressering av dubbelord

Vid adressering av dubbelord (DINT, DWORD) måste en variabel definieras i headern till programdelen (POU).

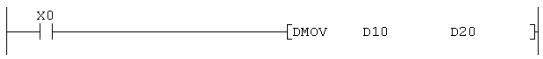
I följande exempel behövs två stycken enkelord till instruktionen DMOV, som flyttar ett dubbelord. För att adressera detta i GX Developer och i en MELSEC-editor i GX IEC Developer behöver endast första registeradressen anges (i exemplet D10 och D20). Varje register efter detta som behövs (D11 och D21) adresseras automatiskt av kompilatorn.

I en IEC-editor i GX IEC Developer måste en variabel definieras i headern till programdelen, (i exemplet var_D10, var_D20) av en datatyp beroende på instruktionen (här DINT (32 bitar)). För nedanstående variabler väljer kompilatorn själv vilka adresser som skall användas internt.

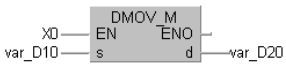
MELSEC instruktionslista




Reläschema




Funktionsblocksdiagram




DMOV-instruktionens header

| | Class | Identifier | Type | Initial | Comment |
|---|---|---|---|---|---|
| 0 | VAR_INPUT | s | ANY32 | 0 | |
| 1 | VAR_OUTPUT | d | ANY32 | 0 | |

Programdelens header

| | Class | Identifier | Type | Initial | Comment |
|---|---|---|---|---|---|
| 0 | VAR | var_D10 | DINT | 0 | |
| 1 | VAR | var_D20 | DINT | 0 | |

## Adressering av vektorer

Programmering av instruktioner som använder vektorer med vektorelement som ingångar eller utgångar (16 bitar) kräver att variablerna definieras i programdelens header, enligt de krav som ställs för respektive instruktion.

De individuella vektorelementen adresseras genom att anropa vektorn med det önskade elementet inom hakparenteser (var_xx[x]).

Bilden nedan visar adressering med vektorer för instruktionen ROTC (positionering av roterande bord).

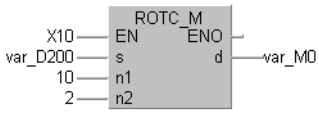
**MELSEC instruktionslista**

MELSEC

| | |
|---|---|
| LD | X0 |
| OUT | M0 |
| LD | X1 |
| OUT | M1 |
| LD | X2 |
| OUT | M2 |
| LD | X10 |
| ROTC | D200 |
| | K10 |
| | K2 |
| | M0 |

**Reläschema**

**Funktiosnblocksdiagram**




**ROTC-instruktionens header**

| | Class | Identifier | Type | Initial | Comment |
|---|---|---|---|---|---|
| 0 | VAR_INPUT | s | ARRAY [ 1.. 3 ] OF AN | 0,0,0 | |
| 1 | VAR_INPUT | n1 | ANY16 | 0 | |
| 2 | VAR_INPUT | n2 | ANY16 | 0 | |
| 3 | VAR_OUTPUT | d | ARRAY [ 1.. 8 ] OF BO | 8(FALSE) | |

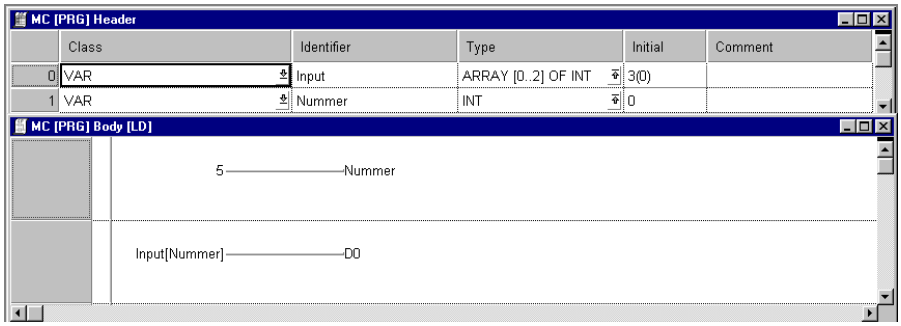
**Programdelens header**

| | Class | Identifier | Type | Initial | Comment |
|---|---|---|---|---|---|
| 0 | VAR | var_D200 | ARRAY [0..2] OF INT | 3(0) | |
| 1 | VAR | var_M0 | ARRAY [0..7] OF BOOL | 8(FALSE) | |

Från headern för ROTC-instruktionen kan utläsas att ingång S består av tre vektorelement av typen ANY16 och att utgången består av 8 vektorelement av typen BOOL.

I en MELSEC-editor i GX IEC Developer och i MELSEC MEDOC och GX Developer behöver endast D200 och M0 anges för operanderna S och D. Kompilatorn adresserar själv D200 till D202 för S och M0 till M7 för D.

I en IEC-editor måste vektorer definieras för S och D. Vektorn för ingången S är definierad som var_D200 och består av tre vektorelement (var_D200[0] - var_D200[2]) av typen INT (enkelord med tecken). Vektorn för utgången D är definierad som var_M0. Den består av 8 vektorelement (var_M0[0] - var_M0[7]) av typen BOOL (bit). Kompilatorn tilldelar interna adresser för varje vektorelement.



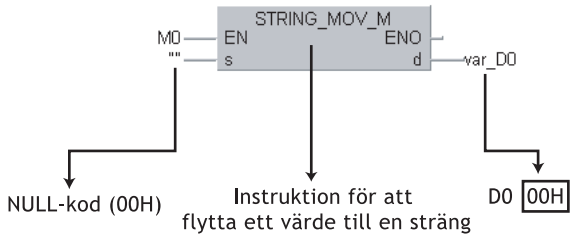

**Observera!**
Vektorer kan också adresseras variabelt. Istället för numret på vektorelementet anges en benämning inom hakparenteserna, t ex [Nummer]. Benämningen "Nummer" måste vara deklarerad som ANY16 (INT eller WORD) i programdelens header. Värdet i "Nummer" bestämmer sedan vilket vektorelement som anropas.

## 7.4.4 Att använda strängar (STRING)

Datatypen STRING ($) hanterar strängar. En sträng består av upp till 50 tecken eller fram till en s k NULL-kod (00H)

### Strängen är NULL (00H)

För att lagra NULL-koden behövs endast ett ord (dataregister).

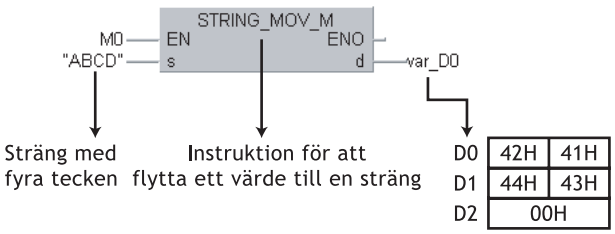



### Strängen innehåller ett jämnt antal tecken

Det antalet ord (dataregister) som behövs för att lagra ett jämnt antal tecken i en sträng beräknas enligt följande formel:

*(Antal tecken / 2) + 1*

Om t ex strängen "ABCD" flyttas till D0, kommer D0 och D1 användas för strängen och D2 för NULL-koden som indikerar att strängen är slut.

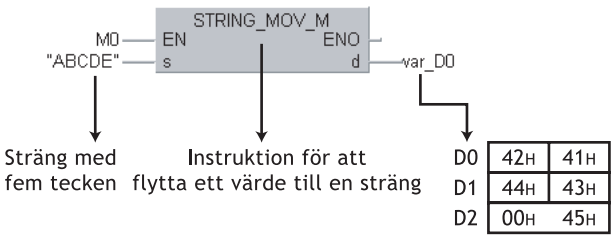



### Strängen innehåller ett udda antal tecken

Det antalet ord (dataregister) som behövs för att lagra ett jämnt antal tecken i en sträng beräknas enligt följande formel:

*(Antal tecken + 1 ) / 2*

Om t ex strängen "ABCDE" flyttas till D0, kommer D0 till D2 användas för strängen och NULL-koden som indikerar att strängen är slut lagras i den höga byten i D2.

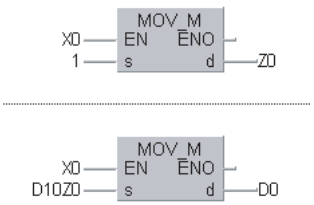


# 7.5 Indexering

Indexering är ett sätt att indirekt adressera en operand med hjälp av ett indexregister. Vid indexering läggs värdet i indexregistret till adressen på den angivna operanden. Det finns 16 indexregister Z0 - Z15. Varje register kan innehålla värden från -32 768 - 32 767. Indexering kan användas på alla ingångar och utgångar samt de flesta typer av operander.

## Att använda indexering

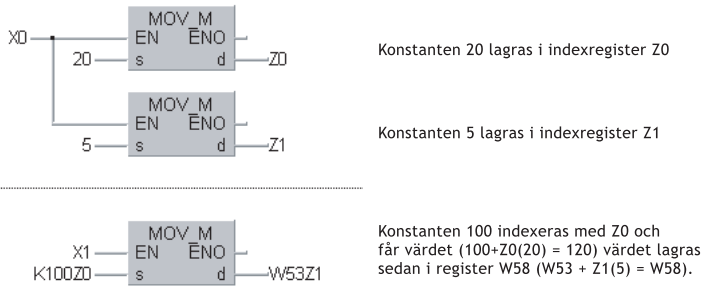
Programmet nedan ger ett exempel på hur indexering fungerar. På första programraden skrivs värdet 1 till indexregister Z0. Detta register fungerar som index till D10 i andra programraden. Alltså lagras värdet från D11 (D10Z0 = D(10+1) = D11) i register D0.




Konstanten 1 lagras i indexregister Z0

Data från det indexerade registret
D(10+Z0) = D(10+1) = D11 lagras i D0

Nedanstående bild visar ännu ett exempel på indexering med operanderna Z0 och Z1. I exemplet är Z0 = 20 och Z1 = 5.




Konstanten 20 lagras i indexregister Z0

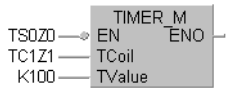
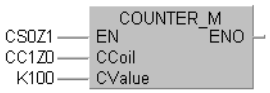
Konstanten 5 lagras i indexregister Z1

Konstanten 100 indexeras med Z0 och får värdet (100+Z0(20) = 120) värdet lagras sedan i register W58 (W53 + Z1(5) = W58).

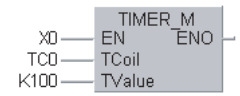
## Operander som inte kan indexeras

| Operand | Beskrivning |
|---|---|
| K, H | Dubbelordskonstanter |
| E | Flyttalsvärden |
| $ | Strängar |
| *.* | Bitadresserade ord |
| P | Pekare (använda som label) |
| I | Interruptpekare (använda som label) |
| Z | Indexregister |
| S | Sekvensflaggor |
| TV, STV | Förvalsvärden till tidskretsar |
| CV | Förvalsvärden till räknare |
| N | Markörer |

# Operander med begränsade indexerings-möjligheter

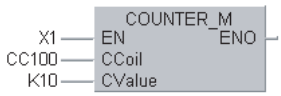
| Operand | Beskrivning | Exempel |
|---|---|---|
| T | Endast Z0 och Z1 kan vändas med tidskretsars ingångar och utgångar |  |
| C | Endast Z0 och Z1 kan vändas med räknares ingångar och utgångar |  |

**Observera!**
För aktuella värden på tidskretar och räknare finns inga begränsningar på vilka indexregister som kan användas.




Förvalsvärde till en tidskrets (TV)
Indexering **ej** tillåten

Aktuellt värde på en tidskrets (TN)
Indexering tillåten



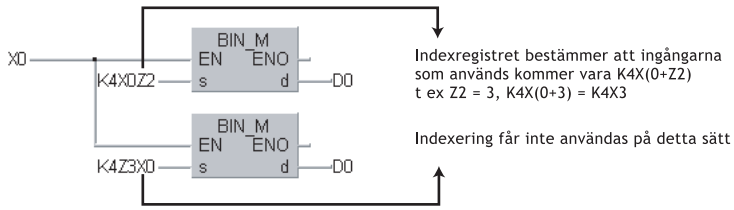

Förvalsvärde till en räknare (CV)
Indexering **ej** tillåten

Aktuellt värde på en tidskrets (TV)
Indexering tillåten

# Övrigt

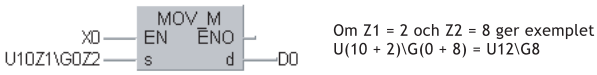
## Bitdata

Bitoperander kan indexeras även då de används som ordoperander. Däremot kan inte blockantalet indexeras.



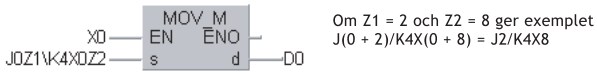

## Funktionsmoduler

Både adressplats och buffertminne kan indexeras med funktionsmoduloperander.




## Nätverksmoduler

Både nätverksnummer och operandnummer kan indexeras med direkta nätverksadresser.



# 7.6 Exekveringsfel

I följande fall kommer ett exekveringsfel att uppstå:

– Om ett buffertminne används på en plats där ingen funktionsmodul är ansluten på angiven plats.

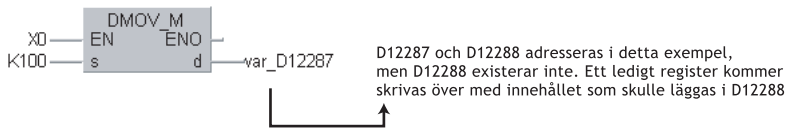
– Om en nätverksadress anropas utan att ett sådant nätverk existerar.

– Om en nätverksadress anropas utan att en nätverksmodul är ansluten på angiven plats.

**Observera!**
Om filregister är inställda i PLC-parametrarna men inget minneskort är installerat (endast QnA) kommer en felkod returneras (2401 = File Set Error).
Om ett filregister adresseras i PLC-programmet men inga filregister är inställda i PLC-parametrarna kommer en felkod returneras (FFFFH)

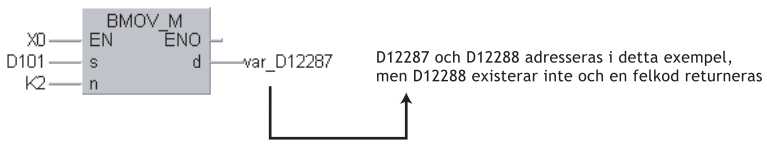
## 7.6.1 Verifiering av adressområde

– Om en instruktion använder operander med en förutbestämd längd (MOV, DMOV o s v), verifieras inte adressområdet. I de fall då det använda adresserna överskrider det tillgängliga området, kommer datan skrivas till en ledig operand. Ingen felkod returneras. Om t ex 12k adresser är parametrerade, kommer ingen felkod returneras förrän adresser över D12287 används.



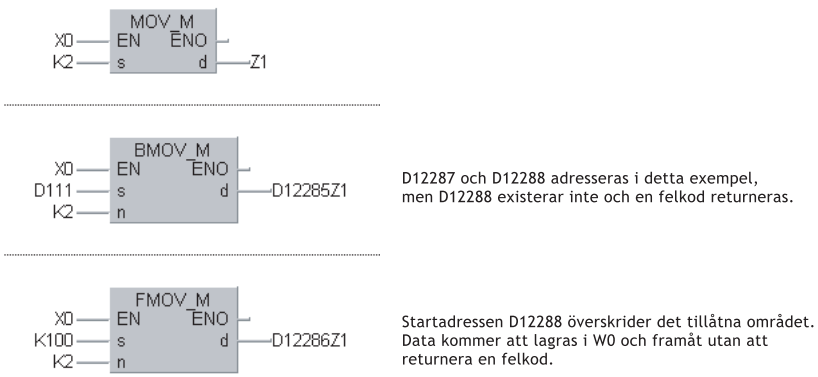

Vid indexering görs ingen verifiering av adressområdet heller.

– Om instruktioner använder operander med varierande längd kommer adressområdet verifieras, (BMOV, FMOV är t ex instruktioner som endast anger startadresser). I de fall då de använda adresserna överskrider det tillåtna området, returneras en felkod. Om t ex 12k adresser är parametrerade returneras en felkod om adress D12287 överskrids.

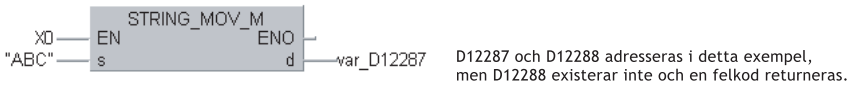



Vid indexering verifieras adressområdet också.

Ingen felkod returneras om startadressen överskrider tillåtet område.



– Eftersom strängar är av variabel längd verifieras adressområdet. Om motsvarande adress överskrider tillåtna gränser returneras en felkod.
Om t ex 12k adresser är parametrerade returneras en felkod om adress D12287 överskrids.




Observera att exekveringsfel inte inträffar även om startadressen överskrider tillåtet område genom indexering.

– Adressområdet verifieras då direktutgångar (DY) indexeras.

## 7.6.2 Verifiering av operanddata

### Verifiering av binär data

– Om resultatet överskrider tillåtet värde, returneras ingen felkod. Carry-flaggan sätts inte.

### Verifiering av BCD-data

– Varje tecken verifieras så att det är en BCD-siffra (0 - 9).
Om någon siffra överskrider 0 - 9 (A - F) kommer en felkod returneras.

– Om resultatet överskrider tillåtet värde, returneras ingen felkod. Carry-flaggan sätts inte.

### Verifiering av flyttalsvärden

Exekveringsfel inträffar i följande fall:

– Flyttalsvärdet är 0

– Det absoluta flyttalsvärdet är mindre än $1,0 \times 2^{-127}$

– Det absoluta flyttalsvärdet är större än $1,0 \times 2^{129}$

### Verifiering av strängar

Datan verifieras inte.

# 7.7 Instruktioners exekveringsvillkor

## 7.7.1 Exekveringsvillkor

Det finns fyra olika typer av exekveringsvillkor för instruktioner:

– Icke-villkorlig exekvering
Instruktionerna exekveras oavsett status på operanden.
Exempel: LD X0, OUT Y10.

– Exekvering då signalen är TILL
Instruktionen exekveras så länge ingångsvillkoret är TILL
Exempel: MOV, FROM.

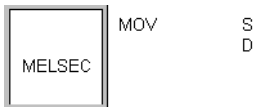
– Exekvering på positiv flank
Instruktionen exekveras på positiv flank (status ändras från 0 till 1) på ingångsvillkoret.
Exempel: PLS, MOVP.

– Exekvering på negativ flank
Instruktionen exekveras på negativ flank (status ändras från 1 till 0) på ingångsvillkoret.
Exempel: PLF.

De flesta instruktioner exekveras då ingångsvillkoret är TILL eller på positiv flank.

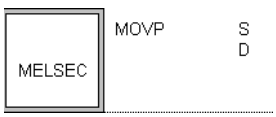
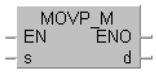
Exekvering då signalen är TILL. Dessa instruktioner är inte märkta på något särskilt sätt.

Exempel: MOV

| MELSEC Instruktionslista | Funktionsblocksdiagram |
|---|---|
|  | |

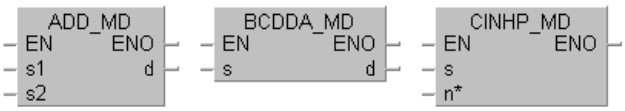
Exekvering på positiv flank utför instruktionen en gång då ingångsvillkoret ändrar status från 0 till 1.

Exempel: MOVP

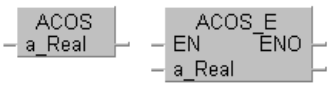
| MELSEC Instruktionslista | Funktionsblocksdiagram |
|---|---|
|  |  |

## 7.7.2 EN-ingången och ENO-utgången

Alla instruktioner som beskrivs i denna manualen finns i *Manufacturer Library* i GX IEC Developer. Förutom ingångar och utgångar finns en EN-ingång och en ENO-utgång på alla blocken.
Bilden nedan visar några MELSEC-instruktioner från *Manufacturer Library* i GX IEC Developer.



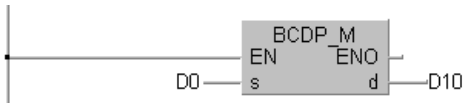

I IEC Standard Library finns nästan alla instruktioner med två gånger. I ena fallet har instruktionen en ändelse "_E". Dessa instruktioner har en EN-ingång och en ENO-utgång.
Bilden nedan visar två IEC-instruktioner från *Standard Library* i GX IEC Developer.




Följande exempel visar de olika exekveringssätten instruktionerna har med eller utan EN och ENO.

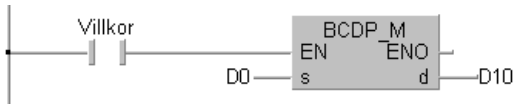
Exempel 1: Anslutning utan ingångskontakt

Utan ingångsvillkor utförs instruktionen varje programvarv.



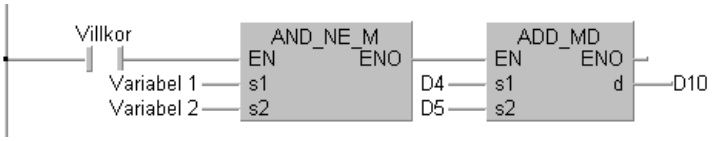

Exempel 2: Anslutning via ingångskontakt

Om EN ansluts till en ingångskontakt, exekveras instruktionen när ingångsvillkoret är uppfyllt.



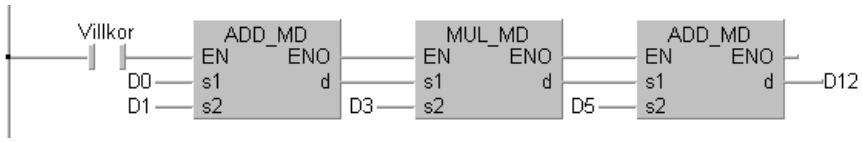

Exempel 3: Anslutning via exekveringsresultat

Om det booelska resultatet av en jämförelseinstruktionen ansluts till EN, utför instruktionen när jämförelsen är uppfylld.




Exempel 4: Anslutning via föregående instruktion

Om EN ansluts till ENO på föregående instruktion, exekveras instruktionerna om ingångsvillkoret är uppfyllt.




**Observera!**
ENO-utgången måste inte alltid anslutas. Signalen in på EN skickas rakt igenom och ut på ENO. Om EN är TILL är ENO det också.

# 7.8 Antal programsteg

Antalet steg i en PLC-instruktion, grundinstruktion eller applikationsinstruktion, skiljer sig beroende på om indirekt adressering är tillåten eller inte.
Antalet steg för grundinstruktioner och applikationsinstruktioner beräknas genom att räkna antalet använda operander plus 1.
Antalet steg i en PLC-instruktion, grundinstruktion eller applikationsinstruktion finns presenterat för respektive instruktion i kapitel 8.

# 8 Instruktioner

## 8.1 Allmänna anvisningar

Detta kapitel beskriver instruktionerna för MELSEC Q och QnA(S). Här beskrivs instruktionens funktion, vilka operander som kan användas antalet programrader som beläggs och vilket PLC-system som kan användas. Beskrivning av varje instruktion inleds med en översikt i tabellform.

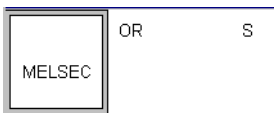
1 2 3

| Instruktionsnamn | | | | | | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | ● | ● | ● 1) | 1-3 2) |
| Tillåtna operander | | | | | | | | | |
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, E, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | DX, DY, BL | |
| S | • | • | • | • | • | • | - | - | • | - |

4

1) Instruktionen kan ej användas i Q00(J) och Q01
2) Antalet steg varierar:
Interna kanaler eller filregister R0-R32767 tar 1 steg
Direktingångar DX 2 steg
Övriga operander 3 steg

5

| MELSEC Instruktionslista | Ladder |
|---|---|
|   | |
| | Funktionsblock |

1. **Instruktionens namn.**
   Varje instruktion har ett kort prefix t ex MOV. Man skiljer mellan 16- och 32-bitarsinstruktioner. Skall dubbelord bearbetas föregås instruktionen av bokstaven D.
2. **Svart punkt i rutan innebär att PLC-systemet kan utföra denna instruktion.**
3. **Här anges vilket antal programsteg som erfordras.**

4. **Här presenteras vilka operander som instruktionen kan arbeta med**.
   **Interna kanaler, Bitar**: Detta är digitala ingångar och utgångar samt interna minnesceller (X, Y, M)
   **Interna kanaler, Ord**: Detta är interna dataregister (D)
   **R, ZR**: Filregister (R) adresserade i banker och filregister (ZR) direktadresserade.
   **J*/*, Bitar**: Bitoperander hämtade från ett annat PLC-system via MELSEC NET/10(H)
   **J*/*, Ord**: Ordoperander hämtade från ett annat PLC-system via MELSEC NET/10(H)
   **U*\G***: Operander hämtade från en funktionsmoduls buffertminne. U=Adressplats (Hex), G=Buffertminnesadress (Dec).
   **Zn**: Indexregister, Z0-Z15
   **K, E, H(16#)**: Decimala konstanter (K), flyttalskonstanter (E) och hexadecimala konstanter (H, 16#)
   **Övriga**: Direktadresserade ingångar (DX), direktadresserade utgångar (DY), sekvensblock (BL), adressplats för funktionsmodul (U) (ex. U2 om startadressen är H20)
   **Flaggor**: Här anges om vilka kontrollflaggor som kan aktiveras eller användas då instruktionen exekveras.

5. **Bilder på hur instruktionen ser ut i olika programeditorer**.
   Bilderna är tagna från programmeringsverktygen, GX Developer och GX IEC Developer. De visar hur instruktionen ser ut i MELSEC Instruktionslista, Ladder, Funktionsblocksdiagram och Strukturerad text. Om det inte finns en bild i någon av rutorna kan instruktionen inte programmeras i respektive editor.

# 8.2 Ingångsinstruktioner

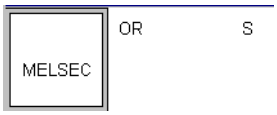
## 8.2.1 Ingångsvillkor, NO- och NC-kontakt, LD, AND, OR

| LD, LDI, AND, ANI, OR, ORI | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | • | • | • | 1-3 [1] |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, E, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | DX, BL | |
| S | • | • | • | • | • | • | - | - | • | - |

1) Antalet steg varierar:
Interna kanaler eller filregister R0-R32767 tar 1 steg
Direktingångar DX 2 steg
Övriga operander 3 steg

**MELSEC Instruktionslista**



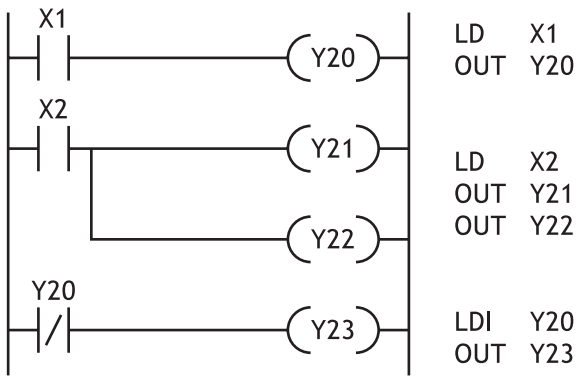

**Ladder**

**Funktionsblock**

## LD, LDI

LD: Början på ett villkor, slutande kontakt.
LDI: Början på ett villkor, brytande kontakt. Ett villkor börjar alltid med LD, LDI-instruktion. Instruktionerna används också för att markera början på ett block i samband med ANB- och ORB-instruktionerna. Alla instruktionerna bearbetas varje cykel oberoende av operandens status.

Programexempel

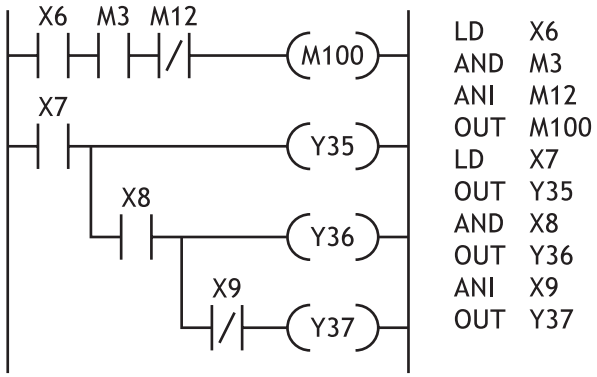



**Observera!**
I GX IEC Developer skall S definieras som en bitvariabel (t ex BOOL)

# AND, ANI

AND: Seriekoppling av slutande kontakt.
ANI : Seriekoppling av brytande kontakt. AND och ANI används för att seriekoppla kontakter och kan aldrig inleda en krets. Ett valfritt antal AND och ANI-instruktioner kan följa efter varandra.

## Programexempel

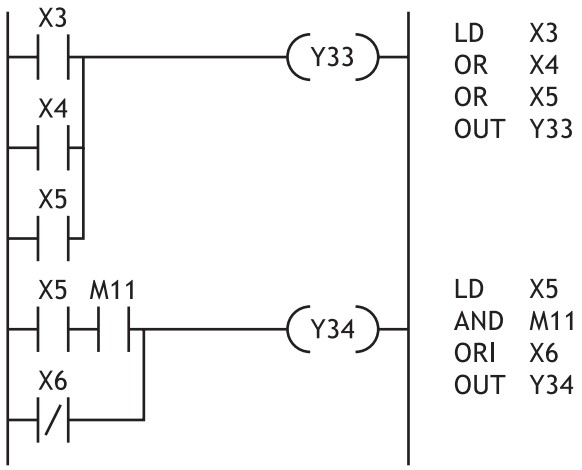



**Observera!**
I GX IEC Developer skall S definieras som en bitvariabel (t ex BOOL)

# OR, ORI

OR: Parallellkoppling av slutande kontakt.
ORI: Parallellkoppling av brytande kontakt. OR och ORI används för att parallellkoppla kontakter. OR och ORI-instruktionen kan inte aldrig inleda en krets. Antalet parallellkopplade kontakter är obegränsat.

## Programexempel




**Observera!**
I GX IEC Developer skall S definieras som en bitvariabel (t ex BOOL)

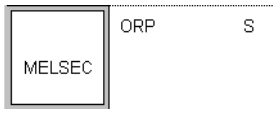
## 8.2.2 Ingångsvillkor, NO- och NC-kontakt, med pulsfunktion, LDP, ANDP, ORP

| LDP, LDF, ANDP, ANDF, ORP, ORF | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | • | • | • | 2-4 1) |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | DX | |
| S | • | • | • | • | • | • | - | - | • | - |

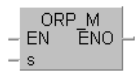
1) Antalet steg varierar:
Interna kanaler eller filregister R0-R32767 åtgår 2 steg
Direktingångar DX 3 steg
Övriga operander 4 steg

**MELSEC Instruktionslista**



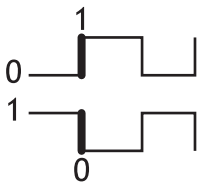

**Ladder**

**Funktionsblock**




### Pulsstart

**LDP** pulsstart på stigande (positiv) flank
**LDF** pulsstart på fallande (negativ) flank




Används på samma sätt som LD fast med flankavkänning.

### Pulsseriekoppling

**ANDP** pulsseriekoppling på stigande (positiv) flank
**ANDF** pulsseriekoppling på fallande (negativ) flank

Används på samma sätt som AND fast med flankavkänning.

### Pulsparallellkoppling

**ORP** pulsparallellkoppling på stigande (positiv) flank
**ORF** pulsparallellkoppling på fallande (negativ) flank

Används på samma sätt som OR fast med flankavkänning.

**Observera!**
I GX IEC Developer skall S definieras som en bitvariabel (t ex BOOL)

# 8.3 Anslutningsinstruktioner

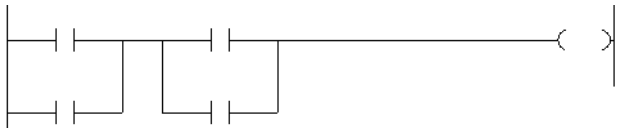
## 8.3.1 Sammankoppling av villkorsblock, ANB, ORB

| ANB, ORB | | | | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|---|---|---|
| | | | | ● | ● | ● | 1 |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| - | - | - | - | - | - | - | - | - | - | - |

**MELSEC Instruktionslista**

**Ladder**

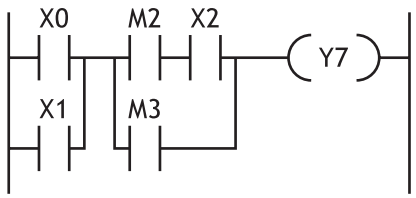


**Funktionsblock**

## ANB

Seriekoppling av villkorsblock. Instruktionen används för att seriekoppla två eller flera kontaktblock. Block A seriekopplas med block B. Varje block inleds med en LD eller LDI-instruktion. ANB är en instruktion utan operand.
Instruktionen kan användas obegränsat antal gånger dock max 7 gånger i följd.

### Programexempel



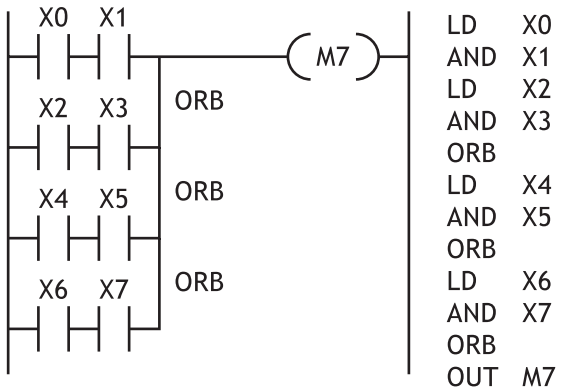

```
LD   X0
OR   X1
LD   M2
AND  X2
OR   M3
ANB
OUT  Y7
```

Instruktionen genereras automatiskt då man programmerar in ovanstående kod i en grafisk editor. Instruktionen syns bara i instruktionslistan.

# ORB

Parallellkoppling av villkorsblock. ORB instruktionen används för att parallellkoppla två eller flera kontaktblock. Block A parallellkopplas med block B. Varje block inleds med en LD eller LDI instruktion. ORB är en instruktion utan operand. Instruktionen kan användas obegränsat antal gånger dock max 7 gånger i följd.

## Programexempel




```
LD   X0
AND  X1
LD   X2
AND  X3
ORB
LD   X4
AND  X5
ORB
LD   X6
AND  X7
ORB
OUT  M7
```

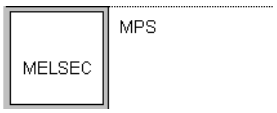
Instruktionen genereras automatiskt då man programmerar in ovanstående kod i en grafisk editor. Instruktionen syns bara i instruktionslistan

## 8.3.2 Förgreningsinstruktioner, MPS, MPP, MRD

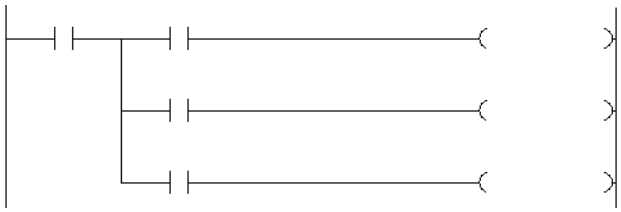
| MPS, MRD, MPP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 1 |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| - | - | - | - | - | - | - | - | - | - | - |

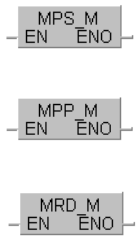
**MELSEC Instruktionslista**




**Ladder**



**Funktionsblock**




## MPS

Lagring av status. När MPS utförs lagras aktuell status i ett speciellt minne. Man får då möjlighet att bygga ut en villkorsgren från denna punkt. Maximalt 16 st MPS-instruktioner programmeras i en följd i en krets innan en MPP-instruktion utförs.

## MRD

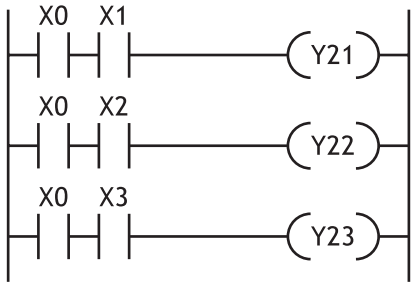
Läsning av lagrad status. MRD-instruktionen läser senaste status av lagrad MPS.

## MPP

MPP-instruktionen läser och nollställer senaste lagrad status. MPP inleder alltid den sista villkorsgrenen efter MPS-instruktionerna. MPS öppnar en gren och MPP avslutar den.

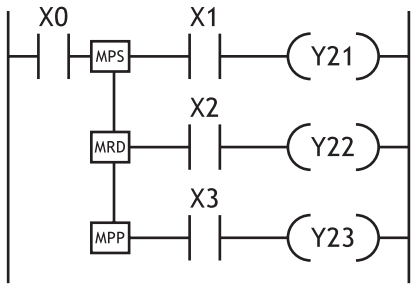
## Programexempel

För jämförelse visas skillnaden mellan att programmera med och utan förgrenings-instruktioner. X0 måste upprepas vid varje enskilt gren.



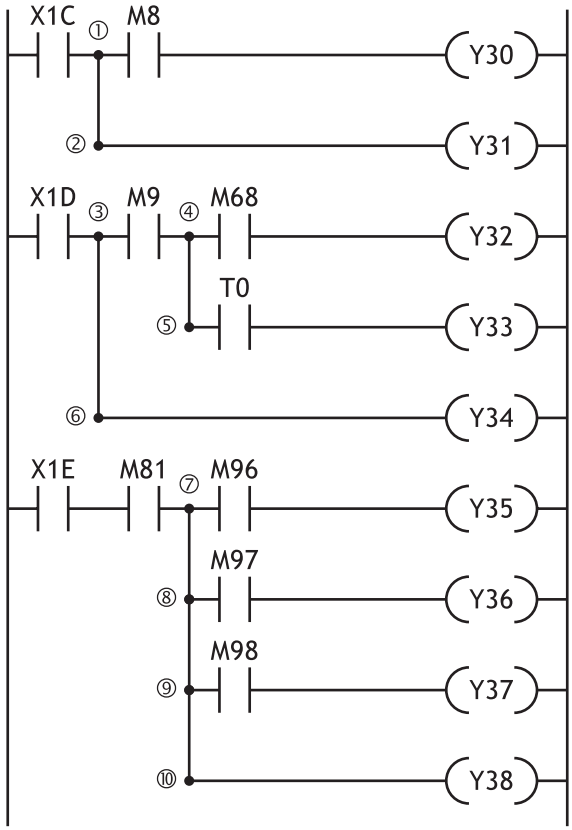

```
LD   X0
AND  X1
OUT  Y21
LD   X0
AND  X2
OUT  Y22
LD   X0
AND  X3
OUT  Y23
```

Instruktionen genereras automatiskt då man programmerar in nedanstående kod i en grafisk editor. Instruktionen syns bara i instruktionslistan.




```
LD   X0
MPS
AND  X1
OUT  Y21
MRD
AND  X2
OUT  Y22
MPP
AND  X3
OUT  Y23
```

## Programexempel




```
   LD   X1C
①MPS
   AND  M8
   OUT  Y30
②MPP
   OUT  Y31
   LD   X1D
③MPS
   AND  M9
④MPS
   AND  M68
   OUT  Y32
⑤MPP
   AND  T0
   OUT  Y33
⑥MPP
   OUT  Y34
   LD   X1E
   AND  M81
⑦MPS
   AND  M96
   OUT  Y35
⑧MRD
   AND  M97
   OUT  Y36
⑨MRD
   AND  M98
   OUT  Y37
⑩MPP
   OUT  Y38
```

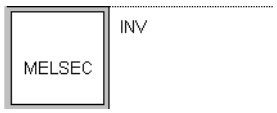
Antalet MPS- och MPP-instruktioner måste vara lika i en krets. Finns det fler MPP än MPS kommer inte programmeringsverktyget att presentera reläschemat korrekt (gäller GX Developer) och programmet fungerar ej som planerat.

## 8.3.3 Invers, INV

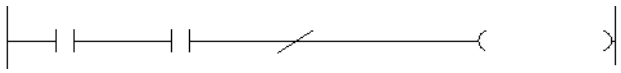
| INV | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 1 |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| - | - | - | - | - | - | - | - | - | - | - |

**MELSEC Instruktionslista**




**Ladder**

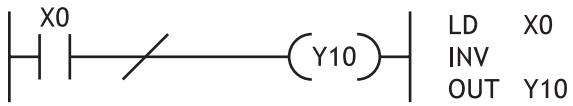


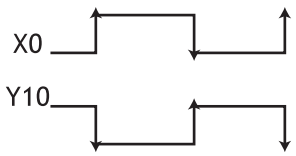
**Funktionsblock**

# INV

Inversinstruktionen inverterar resultatet. Om resultatet är TILL före funktionen kommer det att bli FRÅN efter funktionen. Om resultatet är FRÅN före funktionen kommer det att bli TILL efter funktionen.

### Programexempel



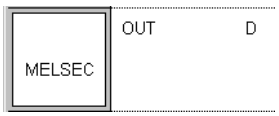
# 8.4 Utgångsinstruktioner

## 8.4.1 Aktivera utgångskontakt, OUT

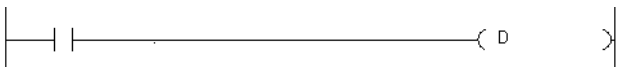
| OUT | | | | | | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | ● | ● | ● | 1-3[2] |
| Tillåtna operander | | | | | | | | | |
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | DY | |
| D | •[1] | • | • | • | • | • | - | - | • | - |

1) Utom T, C, F
2) Interna kanaler 1 steg. Direktutgångar DY 2 steg. Övriga operander 3 steg

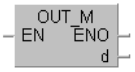
**MELSEC Instruktionslista**




**Ladder**



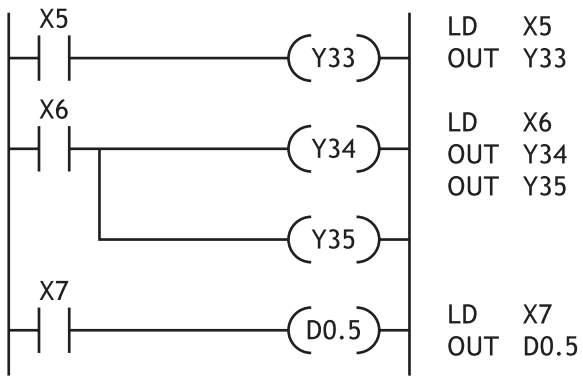

**Funktionsblock**




## OUT

OUT-instruktionen används för att förändra status hos operanderna Y, M, L, S, B, F samt för att aktivera T, C och bitar i ordoperander. Utgångskontakten aktiveras när ingångsvillkoret uppfylls. Flera OUT-instruktioner kan parallellkopplas och skrivas direkt efter varandra. En operand som påverkas av OUT kan avläsas på valfri plats i programmet, som slutande eller brytande kontakt. När OUT aktiverar en felminnescell (F-cell) åtgår tre programsteg.

### Programexempel




*D0.5 aktiverar bit 5 i dataregister D0.*

**Observera!**
I GX IEC Developer skall D definieras som en bitvariabel (t ex BOOL)

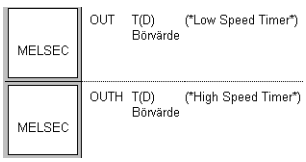
## 8.4.2 Aktivera tidskretskontakt, OUT T

| OUT T, OUTH T | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 4 |

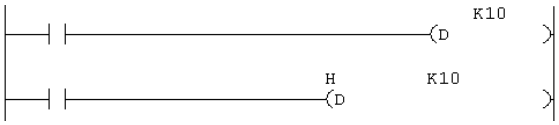
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| D | •1) | - | - | - | - | - | - | - | - | |
| 2) | - | •3) | • | - | • | • | - | •4) | - | |

1) Endast T
2) Förvalsvärde
3) Utom T och C
4) Tidinställning med decimalkonstant endast K. Hexadecimalkonstant kan inte läsas.

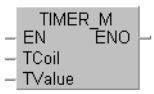
**MELSEC Instruktionslista**




**Ladder**




**Funktionsblock**



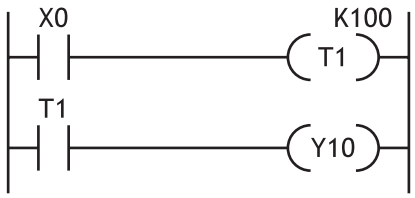

# OUT (T)

När ingångsvillkoret blir uppfyllt sluts kretsen till timern som börjar räkna upp till börvärdet som kan vara en konstant K eller ett värde lagrat i ett dataregister D. När inställd tid har uppnåtts aktiveras timerkontakten.

Förvalsvärdet måste vara inom intervallet 1-32 767.

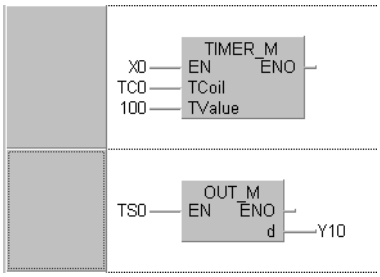
### Programexempel (GX Developer)

Program som aktiverar Y10 tio sekunder efter att X0 har ettställts.




```
LD   X0
OUT  T1
     K100 (=1 sek)
LD   T1
OUT  Y10
```

## Programexempel (GX IEC Developer)




När X0 går till aktiveras tidskretsen TC0 (T0) och fördröjer tillslaget på Y10 med 10 s.

| | | |
|---|---|---|
| TSx | BOOL | Statuskontakt |
| TCx | BOOL | Ingångskontakt |
| TNx | INT | Aktuellt värde |

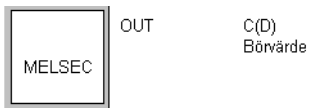
## 8.4.3 Aktivera räknarkontakt, OUT C

| OUT C | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 4 |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | U | |
| D | •1) | - | - | - | - | - | - | - | - | - |
| 2) | - | •3) | • | - | • | • | - | •4) | - | - |

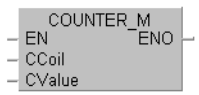
1) Endast C
2) Förvalsvärde
3) Utom T och C
4) Räknarinställning med decimalkonstant endast K. Hexadecimalkonstant kan inte läsas.

**MELSEC Instruktionslista**




**Ladder**

**Funktionsblock**

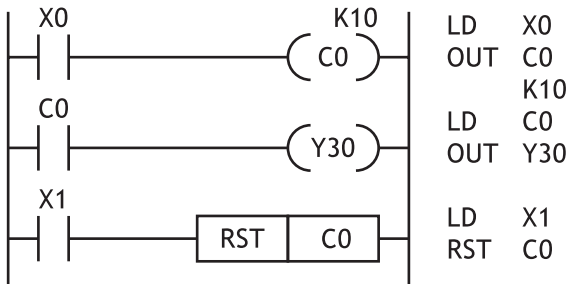



# OUT (C)

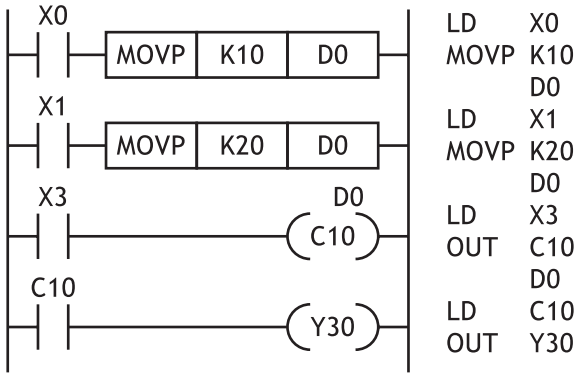
När ingångsvillkoret blir uppfyllt (positiv flank) sluts kretsen till räknaren som räknar till det förvalda värdet (börvärdet) som kan vara en konstant K eller ett värde lagrat i dataregister D. När inställt värde har uppnåtts aktiveras räknarkontakten. Förvalsvärdet måste vara inom intervallet 1-32 767.

### Programexempel (GX Developer)

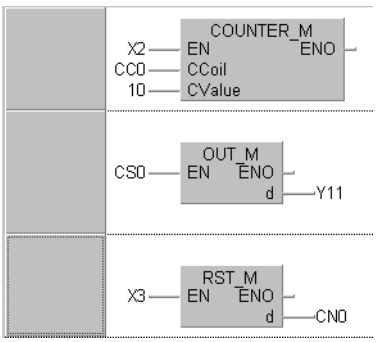
Program som ettställer Y30 när X0 aktiverats 10 gånger. När X1 aktiveras nollställs C0 och Y30.



Program som ändrar börvärdet i C10 till 10 när X0 ettställs och till 20 när X1 ettställs.




## Programexempel (GX IEC Developer)




Räknare C0 räknar tillslag på ingång X2.
Efter tillslag aktiveras utgång Y11. Återställning av räknaren sker genom att aktivera X3.

| | | |
|---|---|---|
| CCx *) | BOOL | Statuskontakt |
| CSx | BOOL | Ingångskontakt |
| CNx | INT | Aktuellt värde |

* ) Används även för återställning av räknaren.

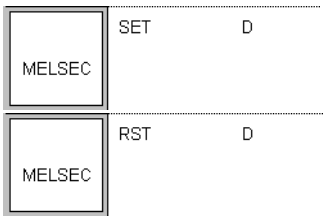
## 8.4.4 Ett- och nollställning av operand, SET, RST

| SET, RST | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 1-4 [1)] |

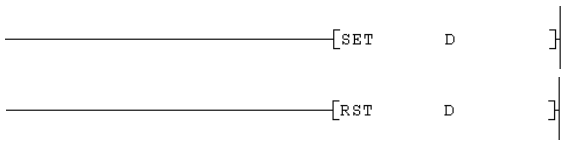
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | BL, DY | |
| D | • | • | • | • | • | • | - | - | • | - |

1) Interna kanaler 1 steg, direktutgångar DY och SFC-block BL 2 steg, övriga operander 3 steg, tidskretsar T eller räknare C 4 steg.

**MELSEC Instruktionslista**




**Ladder**




**Funktionsblock**

## SET

Ettställning av operand. När villkoret framför SET-instruktionen är uppfyllt, ettställs operanden. Den förblir ettställd även om ingångsvillkoret ändras. Återställning av operanden sker med en RST-instruktion. När SET-instruktionen används för ett felminne F ettställs operanden på positiv flank på ingångsvillkoret, dvs SET-instruktionen utförs endast under ett programvarv.

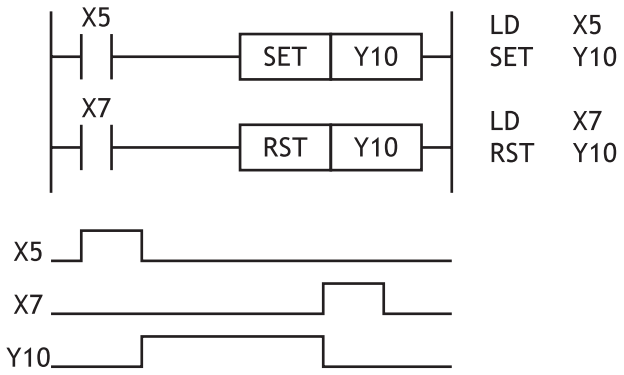
**Observera!**
I GX IEC Developer skall D definieras som en bitvariabel (t ex BOOL)

## RST

Reset (nollställning) av operand. När villkoret framför RST-instruktionen är uppfyllt, nollställs operanden. Den förblir nollställd även om ingångsvillkoret ändras. När RST-instruktionen används för ett felminne F nollställs operanden på positiv flank på ingångsvillkoret, dvs RST-instruktionen utförs endast under ett programvarv.

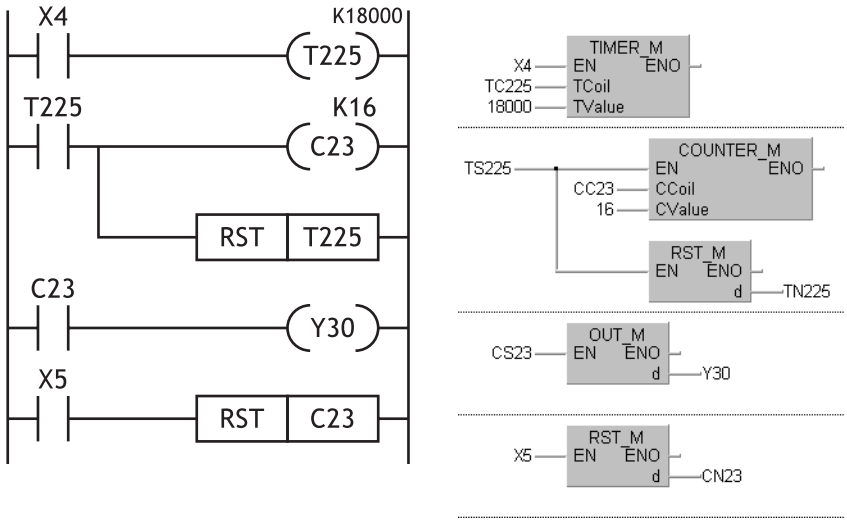
**Observera!**
I GX IEC Developer skall D definieras som en bitvariabel (t ex BOOL) eller enkelordsvariabel (t ext INT)

## Programexempel




## RST T, C

Tidsfunktionen T225 (ackumulerande tidskrets, anges i parametrarna) aktiverar sin utgång när X4 har varit påverkad under 30 min. Räknare C23 räknar upp ett steg då T225 aktiverar sin utgång. När T225 utgångskontakt sluts, återställs T225. Utgång Y30 aktiveras när C23 har räknat till 16. När X5 ettställs återställs C23 och Y30.

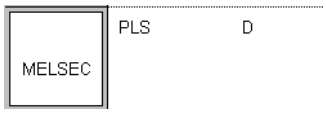


## 8.4.5 Puls på positiv eller negativ flank, PLS, PLF

| PLS, PLF | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 2 |

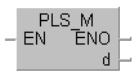
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | DY | |
| D | • | • | • | • | • | • | - | - | • | - |

**MELSEC Instruktionslista**




**Ladder**

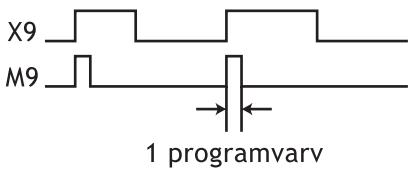
**Funktionsblock**




# PLS

Puls vid positiv flank. När villkoret framför PLS-instruktionen blir uppfyllt, ettställs operanden vid stigande signalflank ett programvarv, varefter återgång sker.

### Programexempel



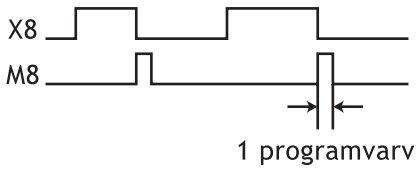

**Observera!**
I GX IEC Developer skall D definieras som en bitvariabel (t ex BOOL)

# PLF

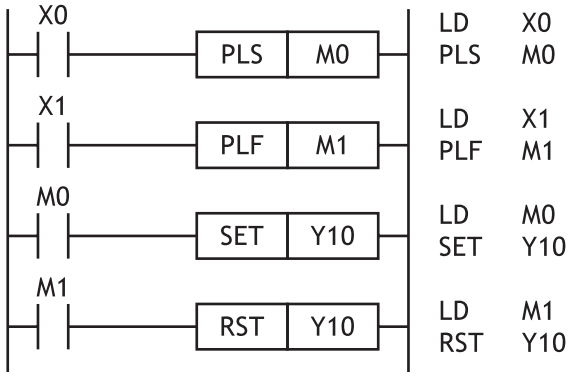
Puls vid negativ flank. När villkoret framför PLF-instruktionen blir uppfyllt, ettställs operanden på fallande signalflank ett programvarv, varefter återgång sker.

**Observera!**
I GX IEC Developer skall D definieras som en bitvariabel (t ex BOOL)

## Programexempel




Instruktionen används ofta ihop med SET och RST-instruktionerna

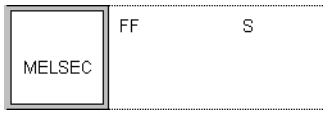


## 8.4.6 T-vippa (Flip-Flop), FF

| FF | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 2 |

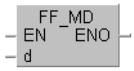
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | DY | |
| D | • | • | • | • | • | • | - | - | • | - |

**MELSEC Instruktionslista**




**Ladder**

**Funktionsblock**

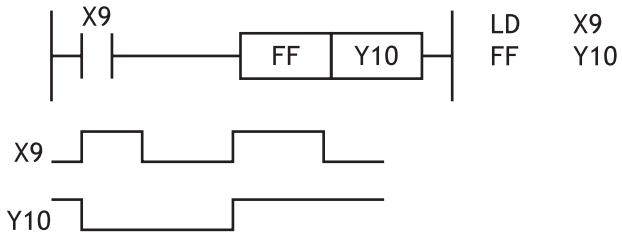



# FF

Med hjälp av FF-instruktionen kan en Flip/Flop-funktion erhållas, vilket innebär att utgångstillståndet hos D ändrar sig vid ettställning av ingångsvillkoret. När ingångsvillkoret ändras måste varaktigheten i ingångsvillkoret överstiga cykeltiden.

**Observera!**
I GX IEC Developer skall S definieras som en bitvariabel (t ex BOOL)

### Programexempel



# 8.5 Skiftinstruktioner

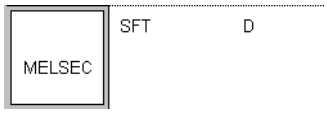
## 8.5.1 Skift av bitar, SFT

| SFT, SFTP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 2 |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | DY | |
| D | •[1)] | •[1)] | • | • | • | • | - | - | • | - |

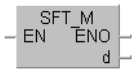
1) Utom T och C

**MELSEC Instruktionslista**




**Ladder**

**Funktionsblock**

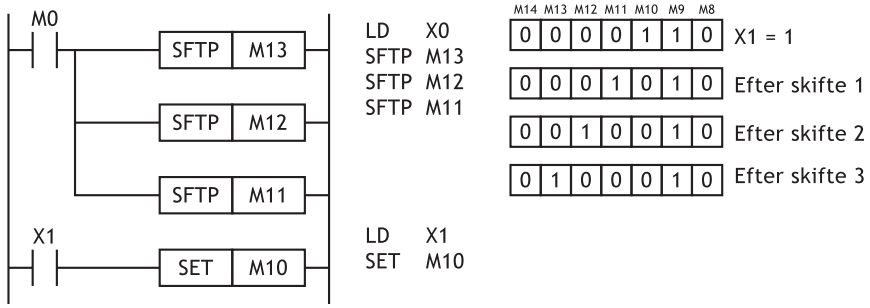



## SFT

SFT-instruktionen flyttar operanddata med hämtadressen (D)-1 till destinationsadressen D när ingångsvillkoret uppfylls. SFTP-instruktionen utförs på positiv flank. Efter skifte nollställs status i hämtadressen (D)-1. När man programmerar flera SFT, SFTP i följd skall den högre adressen skrivas först (jämför nedan).

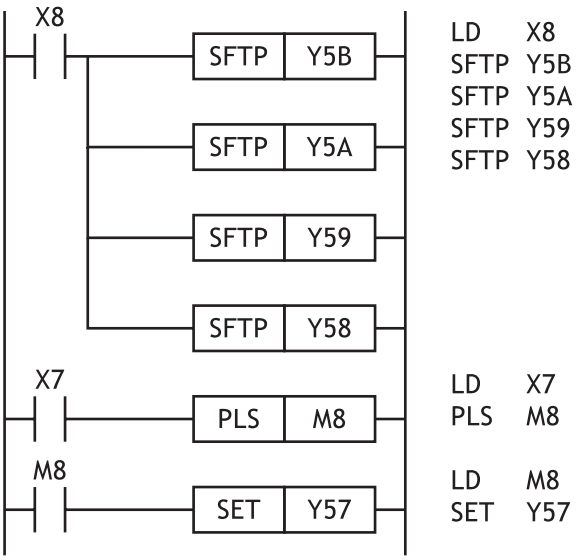
**Observera!**
I GX IEC Developer skall D definieras som en bitvariabel (BOOL).
Den första adressen i en kedja aktiveras med en SET-instruktion.

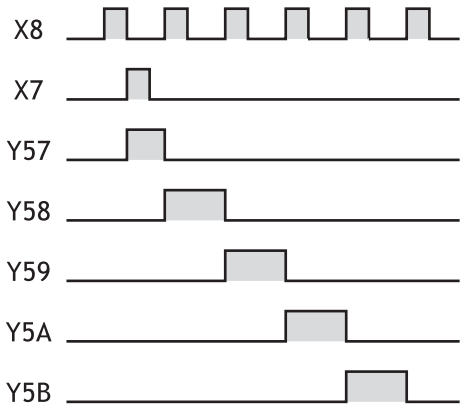
### Programexempel



## Programexempel

Med hjälp av SFTP-instruktionen kommer status Y57 till Y5B skiftas ett steg åt höger varje gång X8 aktiveras. Y57 aktiveras vid positiv flank på X7.



# 8.6 Master Control instruktioner

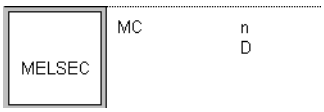
## 8.6.1 Aktivera och återställa överordnat villkor, MC, MCR

| MC, MCR | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 1/2 [1] |

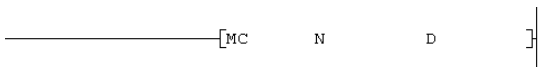
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | N/DY | |
| n | - | - | - | - | - | - | - | - | •/- | - |
| D | • | • | • | • | • | • | - | - | -/• | - |

1) MC använder 2 steg och MCR använder 1 steg.

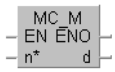
**MELSEC Instruktionslista**




**Ladder**




**Funktionsblock**




## MC

MC-instruktionen (Master Control) är en överordnad startinstruktion till en separat definierad programdel som köres under vissa förutsättningar. När ingångsvillkoren är uppfyllda bearbetas programmet mellan MC- och MCR-instruktionerna. När villkoret ej uppfylls exekveras inte programdelen utan att cykeltiden påverkas.

Status när MC inte är aktiverad

| Tidskretsar 100ms, 10ms | Tidskretsar 100ms (ackumulerande) Räknare | OUT | SET, RST, SFT |
|---|---|---|---|
| Nollställs | Ärvärdet fryses | Operanden nollställs | Operandens status bibehålls |

MC-operationen kan utföras i 8 nivåer. En undernivå måste ligga innanför nivån över. I första nivån skall MC-instruktionen ha markör n=0, nästa n=1 osv. MCR-instruktionerna börjar med den högsta markörnumret sedan näst högsta, osv. Se programexempel.

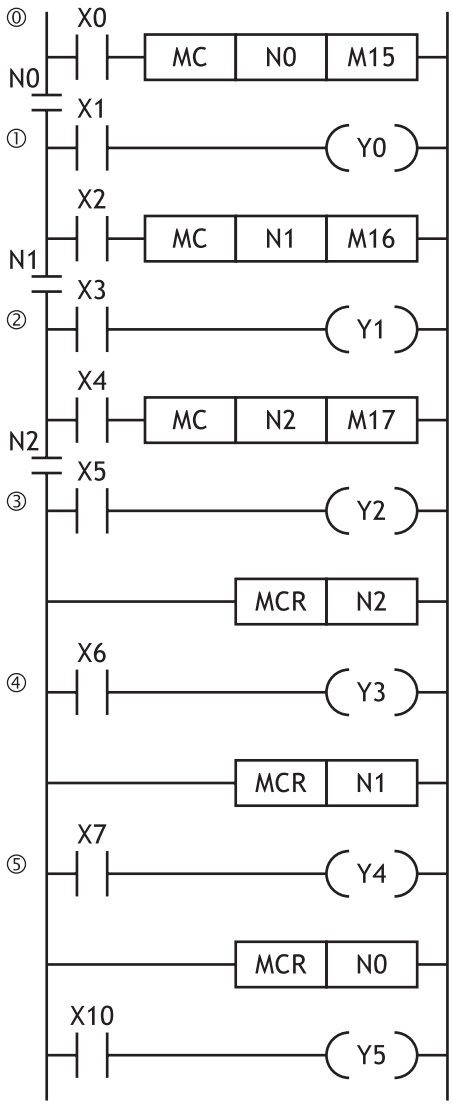
MC-instruktionen kan använda samma nivånummer valfritt antal gånger med olika operandadresser (D). En MC-instruktion måste åtföljas av en LD- eller LDI-instruktion.

**Observera!**
I GX IEC Developer skall D definieras som en bitvariabel (BOOL).

# MCR

MCR-instruktionen (Master Control Reset) återställer MC-instruktionen och markerar slutet på den programdel som har omfattats av det överordnade villkoret. MCR-instruktionen återställer de framförliggande MC-instruktionerna som har samma eller högre markörnummer.

## Programexempel




När X0 = 1 utförs programdel ①

När X0 och X2 = 1 utförs programdel ②

När X0, X2 och X4 = 1 utförs programdel ③

När delar ① och ② har blivit utförda (X1 och X2 = 1) utförs programdel ④

När del ① (X0 = 1) har blivit utförd utförs programdel ⑤

# 8.7 Programavslutande instruktioner

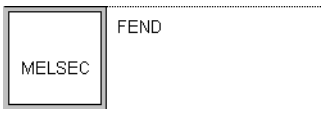
## 8.7.1 Programslut, FEND, END

| FEND, END | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 1 |

**Tillåtna operander**

| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | U | |
| - | - | - | - | - | - | - | - | - | - | SM0 |

**MELSEC Instruktionslista**




**Ladder**

**Funktionsblock**

## FEND, END

FEND-instruktionen markerar slutet på en programdel, t ex huvudprogrammet eller en subrutin. END-instruktionen är definitiv och kan inte användas i någon annan programdel. Efter END återgår exekveringen till steg 0.

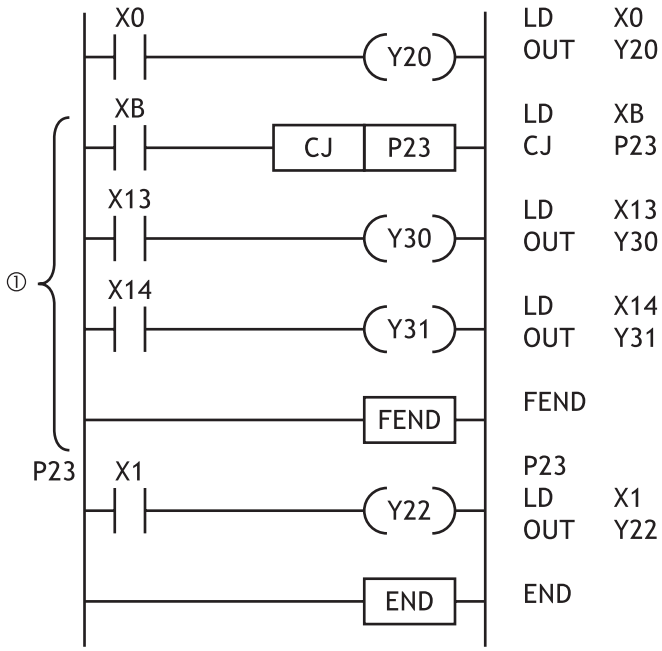
Användning av FEND- och END-instruktionerna.

| PLC-program |
|---|
| FEND |
| Subrutinprogram |
| Interruptprogram |
| END |

MAIN-Programarea

## Programexempel

### FEND, END




Programavsnitt 1 utförs när XB är FRÅN. När XB är TILL hoppar programmet avsnitt över avsnitt 1 ner till pekaren 23 och fortsätter bearbetningen där.

Se även beskrivning av CJ, *kapitel 8.13.1*.

# 8.8 Diverse instruktioner

## 8.8.1 Programavbrott, STOP

| STOP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 1 |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | U | |
| - | - | - | - | - | - | - | - | - | - | SM0 |

**MELSEC Instruktionslista**

**Ladder**

**Funktionsblock**

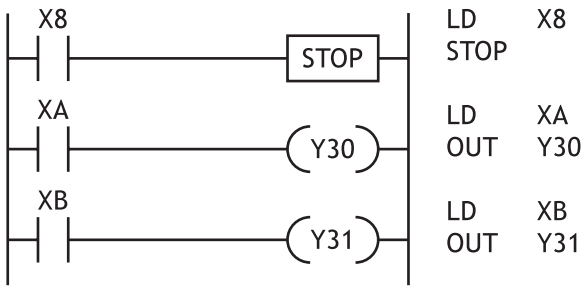
## STOP

När STOP aktiveras kommer programexekveringen att avbrytas och utgångarna nollställas. Samma funktion erhålls när RUN-omkopplaren sätts i STOP-läge. När STOP-instruktionen utförs kommer bit 4 - 7 i specialdataregister SD203 lagra binärvärdet 3.

| b15 | | | | | | | b8 | b7 | | | b4 | b3 | | | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 0 | 0 | 1 | 1 | | | | |

Återgång till programexekveringen sker genom att först sätta RUN-omkopplaren i STOP och sedan tillbaka till RUN. Om LATCH-CLEAR skall utföras sedan en STOP-instruktion har aktiverats måste RUN-omkopplaren först sättas i STOP-läge. STOP-instruktionen får inte användas i interrupt- eller subrutinprogram eller i FOR/NEXT loopar.

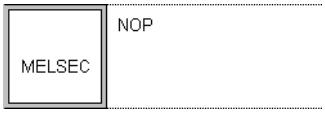
### Programexempel



## 8.8.2 Tomt programsteg, NOP

| NOP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 1 |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| | - | - | - | - | - | - | - | - | - | - |

| MELSEC Instruktionslista | Ladder |
|---|---|
|  | |
| | **Funktionsblock** |

# NOP

Denna instruktion påverkar inte föregående programdel. En NOP-instruktion (**No Operation**) används till att reservera plats i programmet för felsökning eller för att lägga till nya instruktioner. Med NOP-instruktionen kan man även radera (skriva över) befintliga instruktioner utan att stegnumren ändras i programmet, eller radera instruktion tillfälligt.

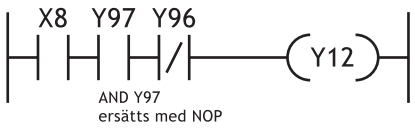
Utfyllnad i programmet för att förlänga cykeltiden.

## Programexempel

### NOP

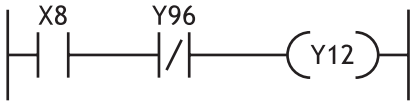
Ersätt en AND, ANI-instruktion med ett tomt steg NOP.

**Före ändring**




```
LD    X8
AND   Y97
ANI   Y96
OUT   Y12
```

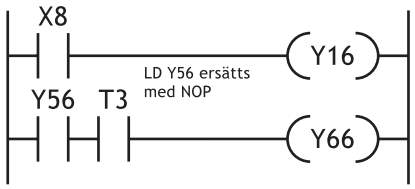
**Efter ändring**




```
LD    X8
NOP
ANI   Y96
OUT   Y12
```

Ersätt en LD, LDI-instruktion med ett tomt steg NOP. Försiktighet rekommenderas eftersom logiken förändras.

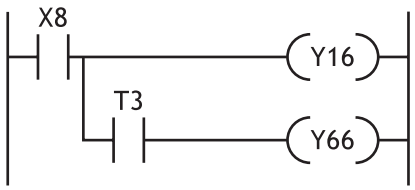
**Före ändring**




```
LD    X8
OUT   Y16

LD    Y56
AND   T3
OUT   Y66
```

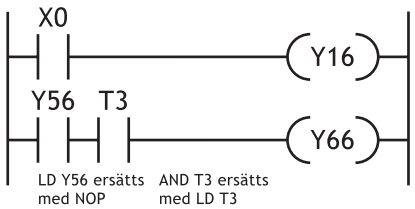
**Efter ändring**




```
LD    X8
OUT   Y16
NOP
AND   T3
OUT   Y66
```

Två ändringar måste göras för att få nedanstående funktion

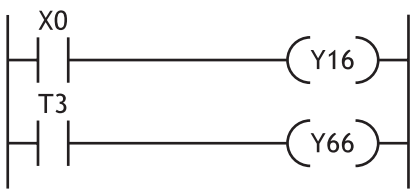
**Före ändring**




```
LD    X0
OUT   Y16

LD    Y56
AND   T3
OUT   Y66
```

**Efter ändring**




```
LD    X0
OUT   Y16
NOP
LD    T3
OUT   Y66
```

# 8.9 Jämförelseinstruktioner

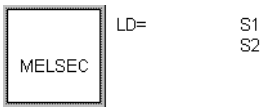
## 8.9.1 Jämförelse av villkor, 16 bitar, binärt

| =, <>, >, <, >=, <= | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 3 |

**Tillåtna operander**

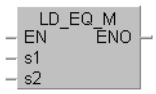
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S1 | • | • | • | • | • | • | • | • | - | - |
| S2 | • | • | • | • | • | • | • | • | - | - |

**MELSEC Instruktionslista**




**Ladder**

**Funktionsblock**




## Funktion

En jämförelseinstruktion används för att jämföra två tal S1 och S2. Jämförelseinstruktionen kan ses som ett villkor. När villkoret uppfylls enligt tabell aktiveras efterföljande funktion typ utgång, puls, aritmetisk operation osv.

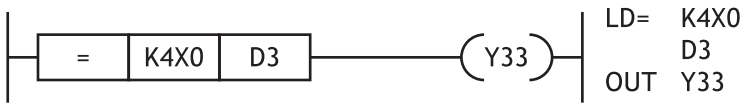
| Instruktion | Status | |
|---|---|---|
| | uppfyllt | ej uppfyllt |
| = | S1 = S2 | S1 ≠ S2 |
| <> | S1 ≠ S2 | S1 = S2 |
| > | S1 > S2 | S1 ≤ S2 |
| <= | S1 ≤ S2 | S1 > S2 |
| < | S1 < S2 | S1 ≥ S2 |
| >= | S1 ≥ S2 | S1 < S2 |

**Observera!**
I GX IEC Developer skall S1 och S2 definieras som enkelordsvariabler (t ex INT)

## Programexempel

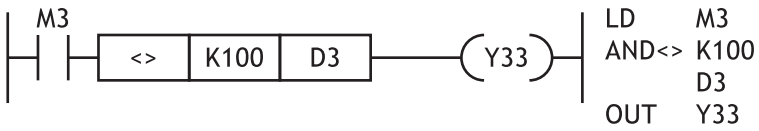
**=**

Program som jämför data X0 till XF med innehållet i dataregister D3. Är dessa lika kommer Y33 att aktiveras.



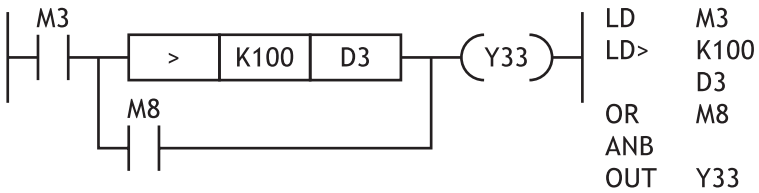

**<>**

Programexempel som visar jämförelse av decimaltalet 100 med datavärdet i D3. Är värdet i D3 skiljt från K100 kommer Y33 att aktiveras.




**>**

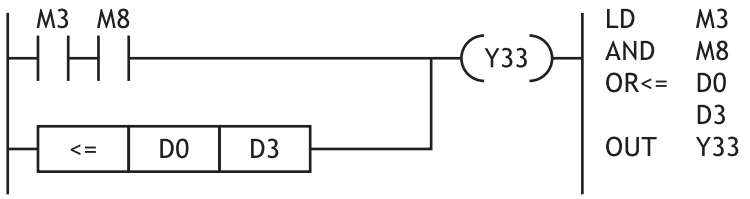
Exempel som jämför decimaltalet 100 med datavärdet i D3. Är värdet decimaltalet 100 större än värdet i D3, eller om M8 är TILL, kommer Y33 att aktiveras.




**<=**

Sista programexemplet jämför datavärdet i D0 med datavärdet i D3.

Är värdet i D0 mindre än eller lika med värdet D3, eller om M3 och M8 är TILL, kommer Y33 att aktiveras.



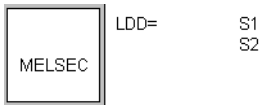
## 8.9.2 Jämförelse av villkor, 32 bitar, binärt

| D=, D<>, D>, D<, D>=, D<= | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | • | • | • | 3 |

**Tillåtna operander**

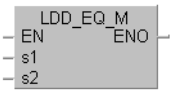
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S1 | • | • | • | • | • | • | • | • | - | - |
| S2 | • | • | • | • | • | • | • | • | - | |

**MELSEC Instruktionslista**




**Ladder**

**Funktionsblock**




# Funktion

En jämförelseinstruktion används för att jämföra två tal S1 och S2. Jämförelseinstruktionen kan ses som ett villkor. När villkoret uppfylls enligt tabell aktiveras efterföljande funktion typ utgång, puls, aritmetisk operation osv.

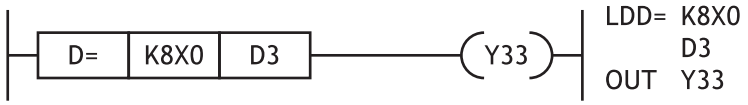
| Instruktion | Status | |
|---|---|---|
| | uppfyllt | ej uppfyllt |
| D= | (S1+1, S1) = (S2+1, S2) | (S1+1, S1) ≠ (S2+1, S2) |
| D<> | (S1+1, S1) ≠ (S2+1, S2) | (S1+1, S1) = (S2+1, S2) |
| D> | (S1+1, S1) > (S2+1, S2) | (S1+1, S1) ≤ (S2+1, S2) |
| D<= | (S1+1, S1) ≤ (S2+1, S2) | (S1+1, S1) > (S2+1, S2) |
| D< | (S1+1, S1) < (S2+1, S2) | (S1+1, S1) ≥ (S2+1, S2) |
| D>= | (S1+1, S1) ≥ (S2+1, S2) | (S1+1, S1) < (S2+1, S2) |

**Observera!**
GX IEC Developer skall S1 och S2 definieras som dubbelordsvariabler (t ex DINT)

## Programexempel

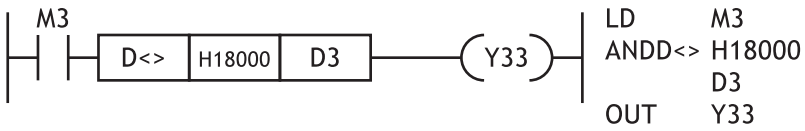
### D=

Program som jämför data X0 till X1F med innehållet i dataregister D3 och D4. Är dessa lika kommer Y33 att aktiveras.



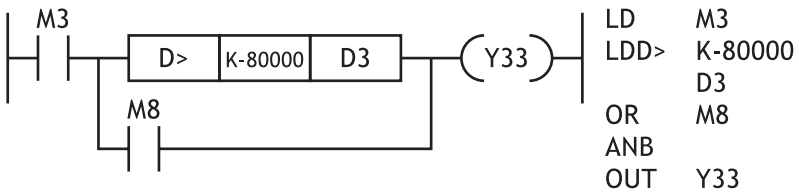

### D<>

Programexempel som visar jämförelse av HEXvärdet 18000 med datavärdet i D3 och D4. Är värdet i D3 och D4 skiljt från H18000 kommer Y33 att aktiveras.



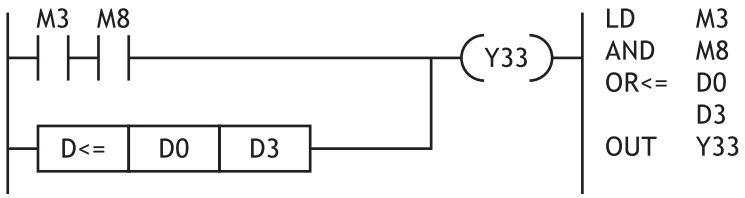

### D>

Exempel som jämför decimalvärdet -80000 med datavärdet i D3 och D4. Om M3 är TILL och om värdet -80000 är större än D3 och D4, eller om M3 och M8 är TILL, kommer Y33 att aktiveras.




### D<=

Sista programexemplet jämför datavärdet i D0 och D1 med datavärdet i D3 och D4. Är värdet i D0 och D1 mindre än eller lika med värdet D3 och D4, eller om M3 och M8 är TILL, kommer Y33 att aktiveras.



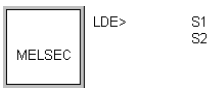
## 8.9.3 Jämförelse av villkor, flyttal

| E=, E<>, E>, E<, E>=, E<= | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ●1) | 3 |

| Tillåtna operander | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | E | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S1 | - | • | • | - | • | • | - | • | - | - |
| S2 | - | • | • | - | • | • | - | • | - | |

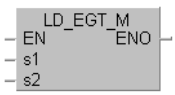
1) Kräver revision B av Q00(J) och Q01

**MELSEC Instruktionslista**




**Ladder**

**Funktionsblock**




# Funktion

Jämförelse av flyttal består av en instruktion och två operander S1 och S2 som skall jämföras. Resultatet av jämförelsen för instruktionerna som följer:

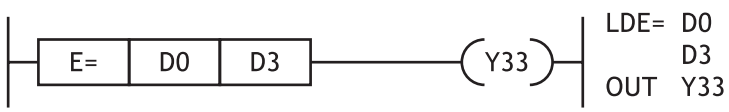
| Instruktion | Jämförelseresultat | |
|---|---|---|
| | uppfyllt | ej uppfyllt |
| E= | S1 = S2 | S1 ≠ S2 |
| E<> | S1 ≠ S2 | S1 = S2 |
| E> | S1 > S2 | S1 ≤ S2 |
| E<= | S1 ≤ S2 | S1 > S2 |
| E< | S1 < S2 | S1 ≥ S2 |
| E>= | S1 ≥ S2 | S1 < S2 |

**Observera!**
I GX IEC Developer skall S1 och S2 definieras som flyttalsvariabler (REAL)

### Programexempel

### E=

Följande exempel jämför flyttalen i D0 och D1 mot flyttalen i D3 och D4. Resultatet ettställer Y33 så länge talen är lika.



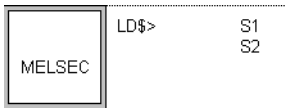
## 8.9.4 Jämförelse av villkor, teckensträngar

| $=,$<>, $>, $<, $>=, $<= | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 3 |

**Tillåtna operander**

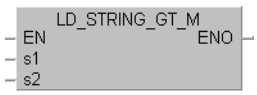
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | $ | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S1 | - | • | • | - | - | - | - | • | - | SM0 |
| S2 | - | • | • | - | - | - | - | • | - | |

**MELSEC Instruktionslista**




**Ladder**

**Funktionsblock**




# Funktion

Jämförelse av tecken består av en instruktion och två operander S1 och S2 som skall jämföras. Resultatet av jämförelsen för instruktionerna utörs med ASCII kod. Tecken för tecken jämföres de med början med första tecknet i strängen.

**Observera!**
I GX IEC Developer skall S1 och S2 definieras som strängvariabler (STRING)

## Programexempel

S1 och S2 innehåller alla tecknen som skall jämföras. Alla tecken fram till koden H00 jämförs.

Om alla tecken stämmer blir resultatet följande:

| | | |
|---|---|---|
| S1 | H42 (B) | H41 (A) |
| S1+1 | H44 (D) | H43 (C) |
| S1+2 | H00 | H45 (E) |

"ABCDE"

Symbol

| | | |
|---|---|---|
| S2 | H42 (B) | H41 (A) |
| S2+1 | H44 (D) | H43 (C) |
| S2+2 | H00 | H45 (E) |

"ABCDE"

| Symbol | Resultat |
|---|---|
| $= | Uppfyllt |
| $<> | Ej uppfyllt |
| $> | Ej uppfyllt |
| $<= | Uppfyllt |
| $< | Ej uppfyllt |
| $>= | Uppfyllt |

Om strängarna ej är lika bestämmer första tecknet som skiljer vilken sträng som är större eller mindre.

| | | |
|---|---|---|
| S1 | H32 (2) | H31 (1) |
| S1+1 | H33 (3) | H34 (4) |
| S1+2 | H00 | H35 (5) |

"12435"

Symbol

| | | |
|---|---|---|
| S2 | H32 (2) | H31 (1) |
| S2+1 | H34 (4) | H33 (3) |
| S2+2 | H00 | H35 (5) |

"123435"

| Symbol | Resultat |
|---|---|
| $= | Ej uppfyllt |
| $<> | Uppfyllt |
| $> | Uppfyllt |
| $<= | Ej uppfyllt |
| $< | Ej uppfyllt |
| $>= | Uppfyllt |

Om strängarna är olika långa kommer den längre strängen bedömas som större av jämförelsen.

# 8.10 Aritmetiska instruktioner

## 8.10.1 Binär addition och subtraktion, +, -

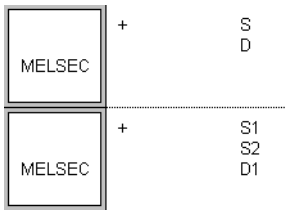
| +, +P,<br>-, -P | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 3/4 [1] |

**Tillåtna operander**

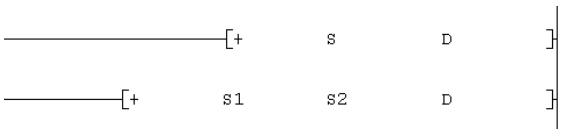
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | • | • | • | • | • | - | - |
| D | • | • | • | • | • | • | • | - | - | - |
| S1 | • | • | • | • | • | • | • | • | - | - |
| S2 | • | • | • | • | • | • | • | • | - | - |
| D1 | • | • | • | • | • | • | • | - | - | - |

1) 3 steg med två operander, 4 steg med tre operander

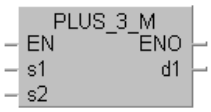
**MELSEC Instruktionslista**




**Ladder**

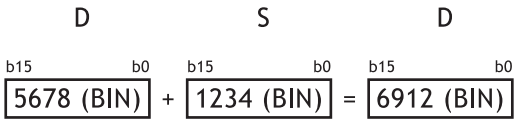



**Funktionsblock**

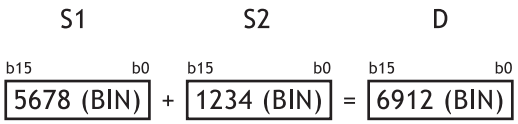



**+**

Till binärvärdet i (D) adderas binärvärdet i (S).




Binärvärdet i (S1) adderas med (S2) och resultatet lagras i (D1)

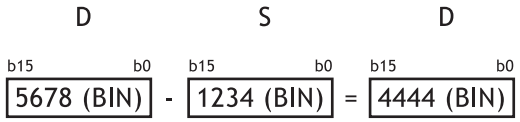


De värden som angivits i register (S1), (S2) och (D), måste vara binära enkelord mellan -32 768 och 32 767. Är den högsta biten (b15) är en nolla är talet positivt och negativt om den är en etta.Carryflaggan aktiveras inte om teckenbiten ettställs.

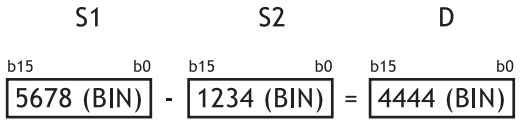
**Observera!**
I GX IEC Developer skall S, S1, S2, D och D1 definieras som enkelordsvariabler (t ex INT)

## -

Binärvärdet i (D) subtraheras med (S) och resultatet lagras i (D)




Binärvärdet i (S1) subtraheras med (S2) och resultatet lagras i (D1)



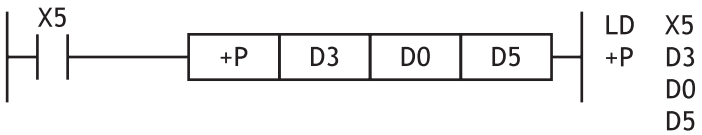

De värden som angivits i register (S), (S1), (S2), och (D), måste vara binära enkelord mellan -32 768 och 32 767. Om den högsta biten (b15) är en nolla är talet positivt och negativt om den är en etta. Carryflaggan aktiveras inte om teckenbiten ettställs.

**Observera!**
I GX IEC Developer skall S, S1, S2, D och D1 definieras som enkelordsvariabler (t ex INT)

## Programexempel

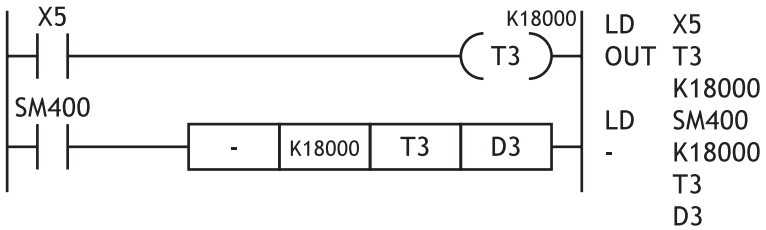
### Addition

Program som adderar datainnehållet i D0 till innehållet i D3 och resultatet överförs till D5 när X5 ettställs (positiv flank).




### Subtraktion

Programexemplet ger differensen mellan bör och ärvärde i timer T3 till D3.



## 8.10.2 Binär addition och subtraktion, D+, D-

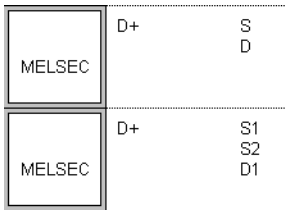
| D+, D+P, D-, D-P | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 3/4 1) |

**Tillåtna operander**

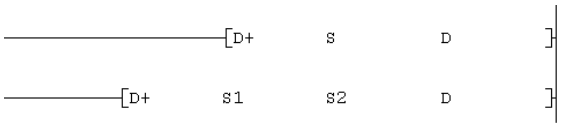
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | • | • | • | • | • | - | - |
| D | • | • | • | • | • | • | • | - | - | - |
| S1 | • | • | • | • | • | • | • | • | - | - |
| S2 | • | • | • | • | • | • | • | • | - | - |
| D1 | • | • | • | • | • | • | • | - | - | - |

1) 3 steg med två operander, 4 steg med tre operander

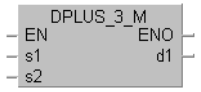
**MELSEC Instruktionslista**




**Ladder**

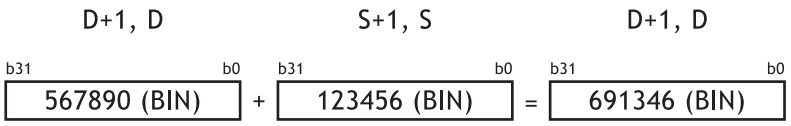



**Funktionsblock**

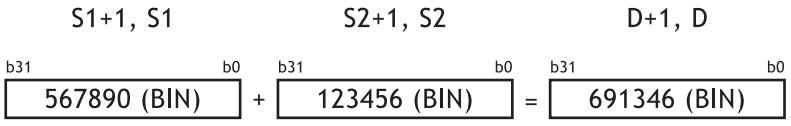



## D+

Till binärvärdet i (D) kommer binärvärdet i (S) att adderas




Binärvärdet i (S1) adderas med (S2) och resultatet lagras i (D1).

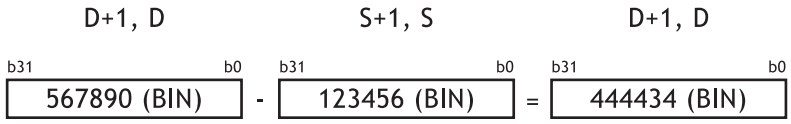



De värden som angivits i register (S), (S1), (S2) och (D), måste vara binära dubbelord mellan -2 147 483 648 och 2 147 483 647. Om den högsta biten (b31) är en nolla är talet positivt och negativt om den är en etta. Carryflaggan aktiveras inte om teckenbiten ettställs.

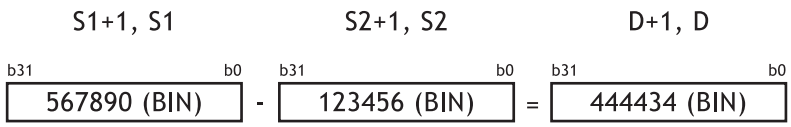
**Observera!**
I GX IEC Developer skall S, S1, S2, D och D1 definieras som dubbelordsvariabler (t ex DINT)

# D-

Binärvärdet i (D) subtraheras med binärvärdet i (S).




Binärvärdet i (S1) subtraheras med (S2) och resultatet lagras i (D1).




De värden som angivits i register (S), (S1), (S2), och (D), måste varabinära dubbelord mellan -2 147 483 648 och 2 147 483 647. Om den högsta biten (b31) är en nolla är talet positivt och negativt om den är en etta. Carryflaggan aktiveras inte om teckenbiten ettställs.

---

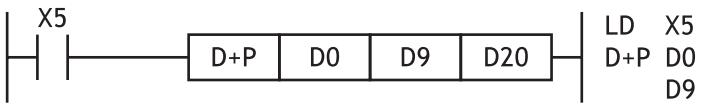
**Observera!**
I GX IEC Developer skall S, S1, S2, D och D1 definieras som dubbelordsvariabler (t ex DINT)

---

## Programexempel

### Addition

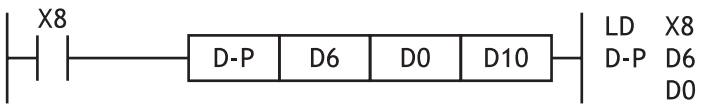
Program som adderar datainnehållet D0 och D1 till D9 och D10. Resultatet överförs till D20 och D21 när X0 ettställs (positiv flank).




```
LD   X5
D+P  D0
     D9
     D20
```

### Subtraktion

Program som subtraherar datainnehållet D0 och D1 från D6 och D7 och lagrar resultatet i D10 och D11 när X8 ettställs (positiv flank).




```
LD   X8
D-P  D6
     D0
     D10
```

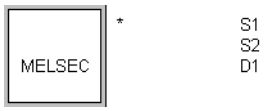
## 8.10.3 Binär multiplikation och division, *, /

| *, *P, /, /P | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 4 |

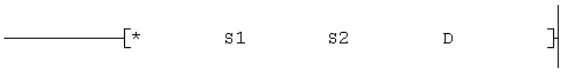
**Tillåtna operander**

| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S1 | • | • | • | • | • | • | • | • | - | SM0 |
| S2 | • | • | • | • | • | • | • | • | - | |
| D1 | • | • | • | • | • | • | • | - | - | |

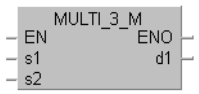
**MELSEC Instruktionslista**




**Ladder**

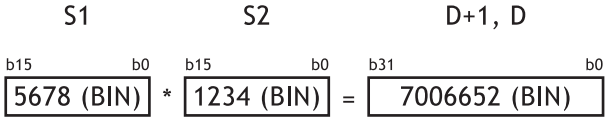



**Funktionsblock**




*****

Binärvärdet i (S1) multipliceras med binärvärdet i (S2) och resultatet lagras i (D).




De värden som angivits i register (S1) och (S2), måste vara binära enkelord mellan -32 768 och 32 767. Resultatet blir ett binärt dubbelord.
Om den högsta biten (b15, b31) är en nolla är talet positivt och negativt om den är en etta. Carryflaggan aktiveras inte om teckenbiten ettställs.

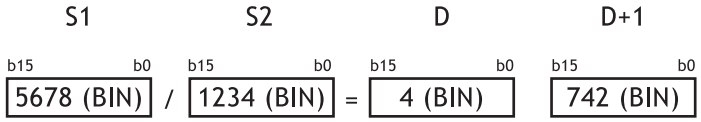
---

**Observera!**
I GX IEC Developer skall S1, S2 definieras som enkelordsvariabler (t ex INT) och D1 som en dubbelordsvariabel (t ex DINT)

---

## /

Det binära värdet i (S1) divideras med det binära värdet i (S2) och resultatet lagras i (D).




I division blir resultatet uppdelat i en kvot och en rest. Kvoten lagras i dataregistret med lägst adress och resten lagras i det efterföljande dataregistret med högst adress. Den högsta biten är teckenbit (b15). Värdet blir positivt när biten är lika med 0 och negativt när biten är lika med 1.

---

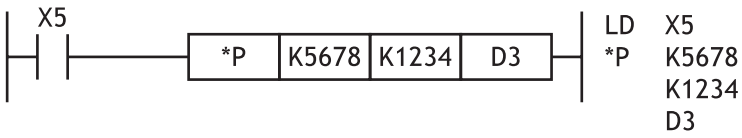
**Observera!**
I GX IEC Developer skall S1, S2 definieras som enkelordsvariabler (t ex INT) och D1 som en vektor med två enkelord (t ex INT)

---

## Programexempel

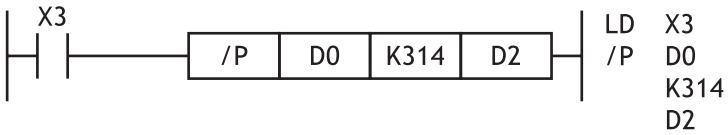
### Multiplikation

Program som multiplicerar 5678 med 1234 och lagrar resultatet i D3 och D4 när X5 ettställs (positiv flank).




### Division

Exemplet visar att när X3 ettställs (positiv flank) kommer innehållet i D0 att divideras med 314. Kvoten lagras i D2 och resten i D3.



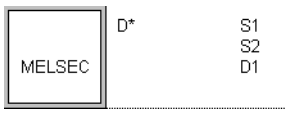
## 8.10.4 Binär multiplikation och division, D*, D/

| D*, D*P, D/, D/P | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | • | • | • | 4 |

**Tillåtna operander**

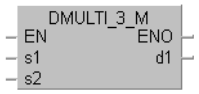
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S1 | • | • | • | • | • | • | • | • | - | SM0 |
| S2 | • | • | • | • | • | • | • | • | - | |
| D1 | • | • | • | - | - | - | - | - | - | |

**MELSEC Instruktionslista**



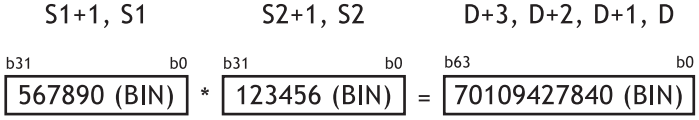

**Ladder**

**Funktionsblock**




## D*

Binärvärdet i (S1) multipliceras med binärvärdet i (S2) och resultatet lagras i (D).

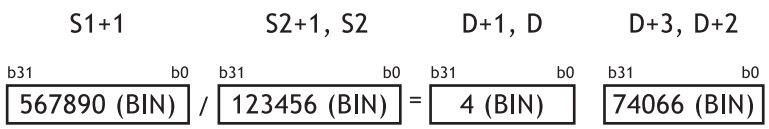



De värden som angivits i register (S1) och (S2) måste vara binära dubbelord mellan -2 147 483 648 och 2 147 483 647. Resultatet blir ett binärt tal i fyra dataregister. Är den högsta biten (D) (b31, b63) en nolla är talet positivt och negativt om den är en etta. Carryflaggan aktiveras inte om teckenbiten ettställs.

**Observera!**
I GX IEC Developer skall S1, S2 definieras som dubbelordsvariabler (t ex DINT) och D1 som en vektor med två dubbelord (t ex DINT)

# D/

Det binära värdet i (S1) divideras med det binära värdet i (S2) och resultatet lagras i (D)



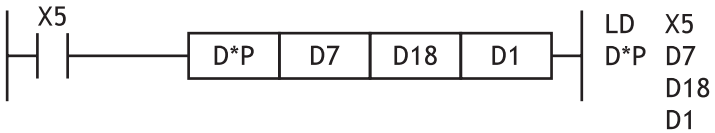

I division blir resultatet uppdelat i en kvot och en rest. Kvoten lagras i dataregistret med lägst adress och resten lagras i de efterföljande dataregister med högst adress. Den högsta biten är teckenbit (b31). Värdet blir positivt när biten är noll och negativt när biten är ett.

**Observera!**
I GX IEC Developer skall S1, S2 definieras som dubbelordsvariabler (t ex DINT) och D1 som en vektor med två dubbelord (t ex DINT)

## Programexempel

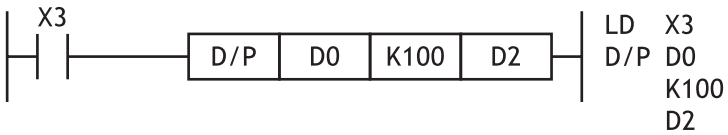
### Multiplikation

Program som multiplicerar binärdata i D7 och D8 med D18 och D19 och lagrar resultatet i D1 till D4 när X5 ettställs (på positiv flank).




### Division

Exemplet visar att när X3 ettställs (positiv flank), kommer innehållet i D0 och D1 att divideras med 100. Kvoten lagras i D2 och D3 och resten lagras i D4 och D5



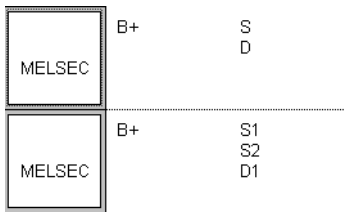
## 8.10.5 BCD-addition och BCD-subtraktion, 4 siffror, B+, B-

| B+, B+P, B-, B-P | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 3/4 1) |

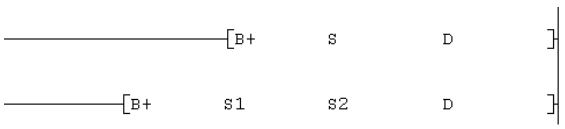
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J** | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | • | • | • | • | • | - | SM0 |
| D | • | • | • | • | • | • | • | - | - | |
| S1 | • | • | • | • | • | • | • | • | - | |
| S2 | • | • | • | • | • | • | • | • | - | |
| D1 | • | • | • | • | • | • | • | - | - | |

1)3 med två operander, 4 med tre operander

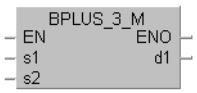
**MELSEC Instruktionslista**




**Ladder**




**Funktionsblock**




## B+

Till BCD-värdet i (D) adderas BCD-värdet i (S) och resultatet lagras i register (D).

D S D

| 5 | 6 | 7 | 8 | + | 1 | 2 | 3 | 4 | = | 6 | 9 | 1 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Till BCD-värdet i (S1) adderas BCD-värdet i (S2) och resultatet lagras i register (D1).

S1 S2 D1

| 5 | 6 | 7 | 8 | + | 1 | 2 | 3 | 4 | = | 6 | 9 | 1 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

De värden som angivits i register (S), (S1), (S2), och (D), måste vara 4-siffriga heltal i BCD-format mellan 0 och 9999. Om antalet siffror är mindre än fyra måste man göra utfyllnad med nollor (ex 12=0012). Om additionen skulle överstiga 9999 kommer carryflaggan inte att aktiveras.

**Observera!**
I GX IEC Developer skall S, S1, S2, D och D1 definieras som enkelordsvariabler (t ex INT)

# B-

Från BCD-värdet i (D) subtraheras BCD-värdet i (S) och resultatet lagras i (D1).

D S D

0 6 7 8 + 0 2 3 4 = 0 4 4 4

Från BCD-värdet i (S1) subtraheras BCD-värdet i (S2) och resultatet lagras i (D1).

S1 S2 D1

0 6 7 8 - 0 2 3 4 = 0 4 4 4

De värden som angivits i register (S), (S1), (S2), och (D), måste vara ett 4-siffrigt heltal i BCD-format (hex) mellan 0 och 9999. Det är nödvändigt att med hjälp av programmet bedöma när resultatet blir positivt eller negativt. Om ett negativt heltal används, kommer felflaggan att sättas.

---

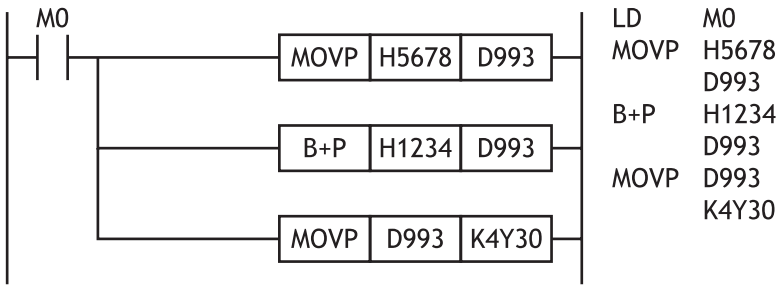
**Observera!**
I GX IEC Developer skall S, S1, S2, D och D1 definieras som enkelordsvariabler (t ex INT)

---

## Programexempel

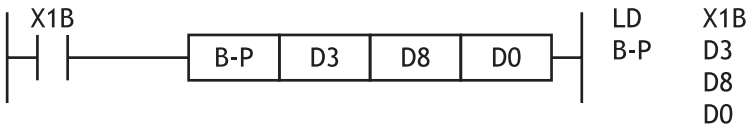
### BCD-addition

När M0 ettställes överförs i den första programdelen 5678 till D993. Sedan adderas BCD-värdet 1234 till värdet i D993 varefter resultatet i D993 överförs till utgångarna Y30 till Y3F (BCD-kodad display).




### BCD-subtraktion

Program som subtraherar innehållet i D8 från innehållet i D3 och placerar resultatet i D0 till M31 när X1B ettställs. Utförs endast en gång på positiv flank.



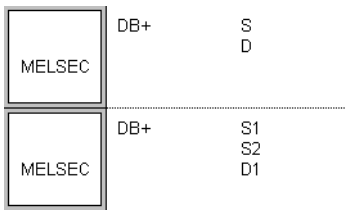
## 8.10.6 BCD-addition och BCD-subtraktion, 8 siffror, DB+, DB-

| DB+, DB+P, DB-, DB-P | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 3/4 [1] |

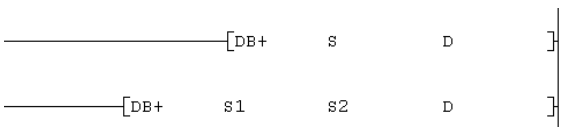
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | • | • | • | • | • | - | SM0 |
| D | • | • | • | • | • | • | • | - | - | |
| S1 | • | • | • | • | • | • | • | • | - | |
| S2 | • | • | • | • | • | • | • | • | - | |
| D1 | • | • | • | • | • | • | • | - | - | |

1) 3 med två operander, 4 med tre operander

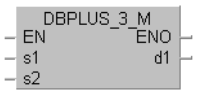
**MELSEC Instruktionslista**




**Ladder**




**Funktionsblock**




## DB+

Till BCD-värdet i (D) adderas BCD-värdet i (S) och resultatet lagras i register (D).

| D1+1, D1 | S+1, S | D1+1, D1 |
|---|---|---|
| 0 9 8 7 1 0 6 8 | + 0 0 3 2 3 4 5 6 | = 1 0 1 9 4 5 2 4 |

Till BCD-värdet i (S1) adderas BCD-värdet i (S2) och resultatet lagras i register (D1).

| S1+1, S1 | S2+1, S2 | D1+1, D1 |
|---|---|---|
| 5 6 7 8 9 1 2 3 | + 0 1 2 3 4 5 6 7 | = 5 8 0 2 3 6 9 0 |

De värden som angivits i register (S), (S1), (S2), och (D) måste vara ett heltal med åtta siffror i BCD-format mellan 0 och 99999999. När antalet siffror är mindre än åtta måste utfyllnad ske med nollor (ex 12345=00012345). Om additionen skulle överstiga 99999999 kommer carryflaggan inte aktiveras.

**Observera!**
I GX IEC Developer skall S, S1, S2, D och D1 definieras som dubbelordsvariabler (t ex DINT)

# DB-

Från BCD-värdet i (D) subtraheras BCD-värdet i (S) och resultatet lagras i (D).

| D1+1, D1 | | S+1, S | | D1+1, D1 |
|---|---|---|---|---|
| 0 9 8 7 1 0 6 8 | - | 0 0 3 2 3 4 5 6 | = | 0 9 7 4 7 6 1 2 |

Från BCD-värdet i (S1) subtraheras BCD-värdet i (S2) och resultatet lagras i (D1).

| S1+1, S1 | | S2+1, S2 | | D1+1, D1 |
|---|---|---|---|---|
| 5 6 7 8 9 1 2 3 | - | 0 1 2 3 4 5 6 7 | = | 5 5 5 5 4 5 5 6 |

De värden som angivits i register (S), (S1), (S2), och (D) måste vara ett heltal med åtta siffror i BCD format mellan 0 och 99999999. Det är nödvändigt att med hjälp av programmet bedöma om resultatet blir positivt eller negativt. Om ett negativt heltal används kommer felflaggan att sättas.

---

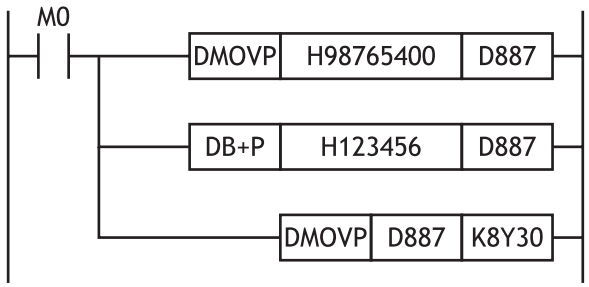
**Observera!**
I GX IEC Developer skall S, S1, S2, D och D1 definieras som dubbelordsvariabler (t ex DINT)

---

## Programexempel

### BCD-addition

När M0 ettställs kommer BCD-värdet 98765400 att lagras i D887 och D888. Därefter kommer BCD-värdet 123456 att adderas till D887 och D888 varefter resultatet överförs till utgångarna Y30 – Y4F (BCD- kodad sifferdisplay).




```
LD      M0
DMOVP   H98765400
        D887
DB+P    H123456
        D887
DMOVP   D887
        K8Y30
```

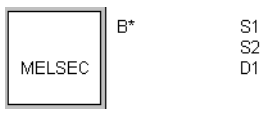
## 8.10.7 BCD-multiplikation och -division, 4 siffror, B*, B/

| **B*, B*P, B/, B/P** | **Q4AR** | **QnA(S)** | **Q** | **Programsteg** |
|---|---|---|---|---|
| | ● | ● | ● | 4 |

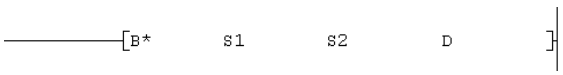
**Tillåtna operander**

| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| **S1** | • | • | • | • | • | • | • | • | - | **SM0** |
| **S2** | • | • | • | • | • | • | • | • | - | |
| **D1** | • | • | • | • | • | • | • | - | - | |

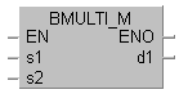
**MELSEC Instruktionslista**




**Ladder**

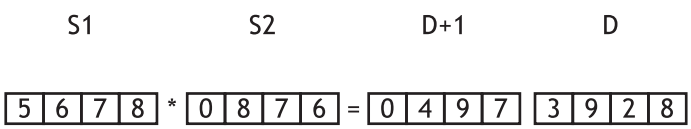



**Funktionsblock**




## B*

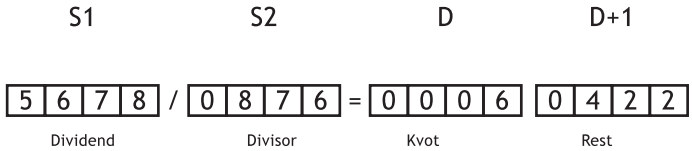
BCD-värdet i (S1) multipliceras med BCD-värdet i (S2) och resultatet påverkar två register (D). De angivna talen måste vara 4-siffriga heltal mellan 0 och 9999.




**Observera!**
I GX IEC Developer skall S1, S2, definieras som enkelordsvariabler (t ex INT) och D1 definieras som en dubbelordsvariabel (t ex DINT)

# B/

BCD-värdet i (S1) divideras med BCD-värdet i (S2) och resultatet lagras i (D). Resultatet kommer som ett 32-bitarsord att delas upp i en kvot och en rest. Resultatet dvs kvoten (4-siffrig BCD) lagras i de 16 lägsta bitarna (D) och restvärdet lagras i de högsta 16 bitarna. Då bitoperander används som resultatoperand lagras inte restvärdet.




De angivna talen i (S1), (S2) och (D) måste vara 4-siffriga heltal mellan 0 och 9999.

När antalet siffror är färre än fyra måste utfyllnad ske med nollor (ex 123=0123).

---

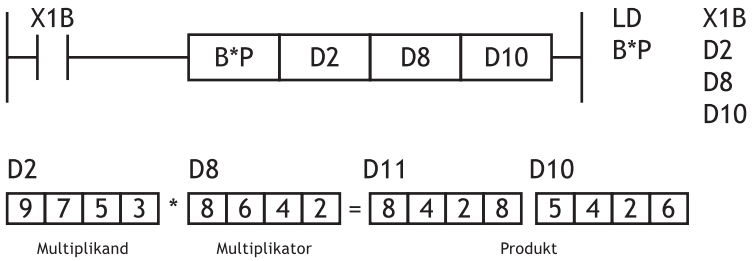
**Observera!**
I GX IEC Developer skall S1, S2, definieras som enkelordsvariabler (t ex INT) och D1 definieras som en vektor med två enkelord (t ex INT)

---

## Programexempel

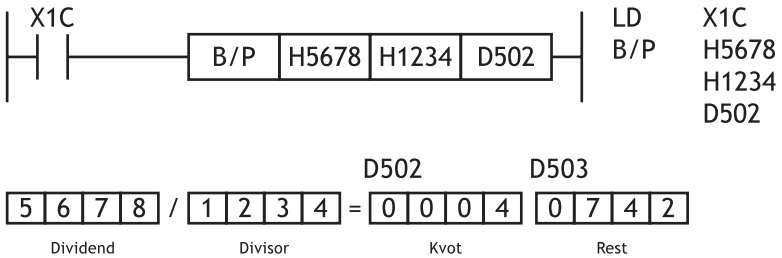
### BCD-multiplikation (4-siffrig)

När X1B ettställs (på positiv flank) multipliceras D2 med innehållet i D8 och lagrar resultatet i D10 och D11.




### BCD-division (4-siffrig)

När X1C ettställes divideras BCD-värdet 5678 med 1234. Resultatet lagras i D502 och restvärdet lagras i D503.

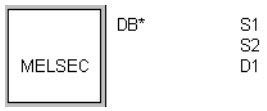


## 8.10.8 BCD-multiplikation och -division, 8 siffror, DB*, DB/

| DB*, DB*P, DB/, DB/P | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | • | • | • | 4 |

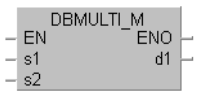
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S1 | • | • | • | • | • | • | • | • | - | SM0 |
| S2 | • | • | • | • | • | • | • | • | - | |
| D1 | • | • | • | • | • | • | • | - | - | |

**MELSEC Instruktionslista**



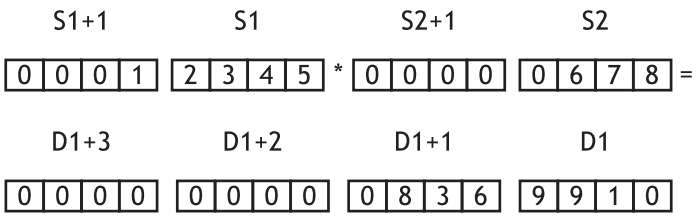

**Ladder**

**Funktionsblock**




## DB*

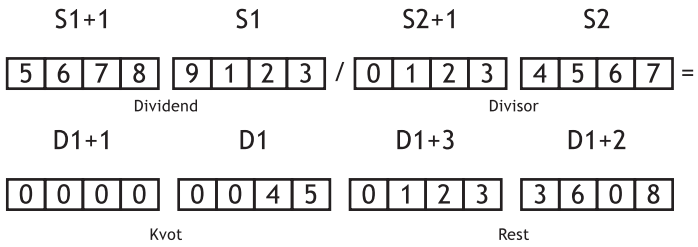
BCD-värdet i (S1) multipliceras med BCD-värdet i (S2) och resultatet lagras i (D). De angivna talen måste vara 8-siffriga heltal i BCD-format mellan 0 och 99999999. Resultatet påverkar 4 register. Om (D) är en grupp bitoperander kan endast de 8 sista siffrorna (lägsta 32 bitarna) presenteras. Mindre antal siffror än 8 måste fyllas ut med nollor (ex 12345=00012345).




**Observera!**
I GX IEC Developer skall S1, S2 definieras som dubbelordsvariabler (t ex DINT) och D1 som en vektor med fyra enkelord (t ex INT)

# DB/

BCD-värdet i (S1) divideras med BCD-värdet i (S2) och resultatet lagras i (D). Resultatet påverkar två dubbelord (64 bitar) som delas upp i en kvot och en rest. Resultatet dvs kvoten lagras i de 32 lägsta bitarna (D) och (D)+1 och restvärdet lagras i de högsta 32 bitarna (D)+2 och (D)+3.Vid bitoperander lagras inte restvärdet.

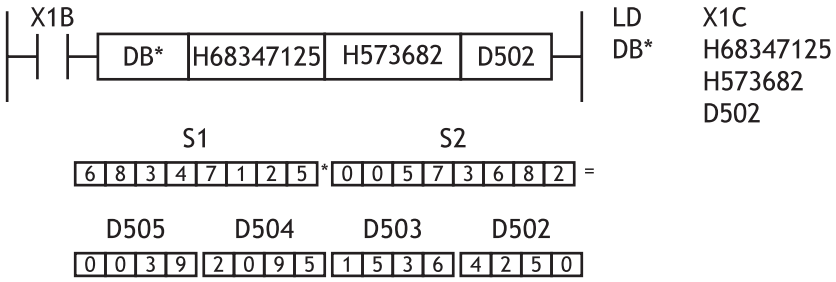



**Observera!**
I GX IEC Developer skall S1, S2 definieras som dubbelordsvariabler (t ex DINT) och D1 som en vektor med två dubbelord (t ex DINT)

## Programexempel

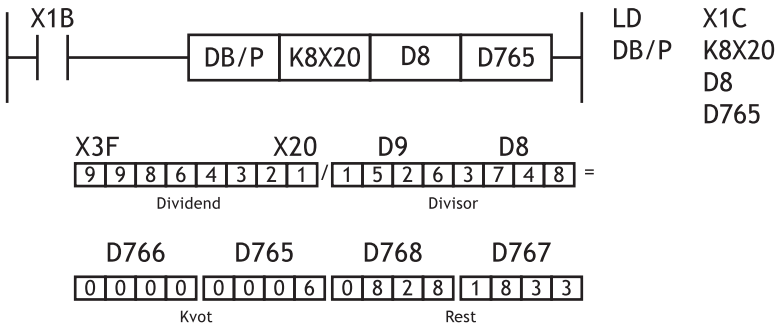
### BCD multiplikation (8-siffrig)

Programmet multiplicerar värdet 68347125 med 573682. Resultatet lagras i D502 till D505.




### BCD division (8-siffrig)

När X1B ettställes kommer BCD-innehållet X20 till X3F (tal från tumhjul) att divideras med BCD-innehållet i D8 och D9. Resultatet lagras i D765 till D768.



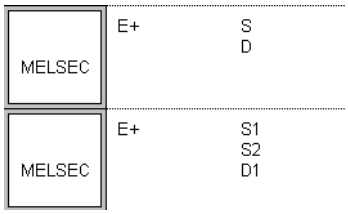
## 8.10.9 Addition och subtraktion av flyttal, E+, E-

| E+, E+P, E-, E-P | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● 1) | 3/4 2) |

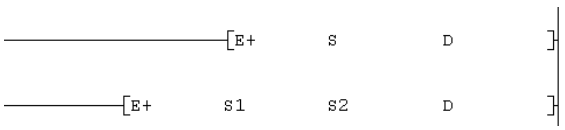
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | E | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | - | • | • | - | • | • | - | • | - | SM0 |
| D | - | • | • | - | • | • | - | - | - | |
| S1 | - | • | • | - | • | • | - | • | - | |
| S2 | - | • | • | - | • | • | - | • | - | |
| D1 | - | • | • | - | • | • | - | - | - | |

1) Kräver revision B av Q00(J) och Q01
2) 3 steg med två operander, 4 steg med tre operander

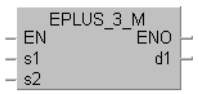
**MELSEC Instruktionslista**




**Ladder**

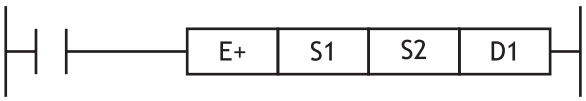



**Funktionsblock**




## E+

Flyttalsvärdet S1 adderas med flyttalsvärdet i S2 och resultatet lagras i D.




All käll- och destinationsdata måste vara i flyttalsformat. Det gäller även konstanter. Dessa skrivs på följande sätt:

– GX IEC Developer: 3.75E-3 (0,00375), 3.756E+2 (375,6)

– GX Developer: E3.75-3 (0,00375), E3.756+2 (375,6)

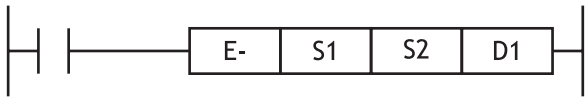
Additionen är matematiskt korrekt, t ex $2.3456 \times 10^2 + (-5.6 \times 10^{-1}) = 2.34 \times 10^2$

Samma register kan användas som käll- och destinationsregister. Då kommer E+ använda resultatet av föregående beräkning som källdata vid nästa beräkning osv. Detta kommer att utföras vid varje programvarv om man inte använder ett förreglingsprogram eller ett pulstillägg (P).

**Observera!**
I GX IEC Developer skall S1, S2 och D1 definieras som flyttalsvariabler (REAL)

# E-

Flyttalsvärdet i S2 subtraheras från flyttalsvärdet i S1 och resultatet lagras i D.




All käll- och destinationsdata måste vara i flyttalsformat. Det gäller även konstanter. Dessa skrivs på följande sätt:

– GX IEC Developer: 3.75E-3 (0,00375), 3.756E+2 (375,6)

– GX Developer: E3.75-3 (0,00375), E3.756+2 (375,6)

Subtraktionen är matematiskt korrekt tex $2.3456 \times 10^{2} - (-4.4 \times 10^{-1}) = 2.35 \times 10^{2}$

Samma register kan användas som käll- och destinationsregister. Då kommer E- använda resultatet av föregående beräkning som källdata vid nästa beräkning osv. Detta kommer att utföras vid varje programvarv om man inte använder ett förreglingsprogram eller ett pulstillägg (P).

**Observera!**
I GX IEC Developer skall S1, S2 och D1 som definieras som flyttalsvariabler (REAL)

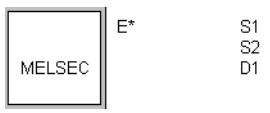
# 8.10.10 Multiplikation och division av flyttal, E*, E/

| E*, E*P, E/, E/P | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ●1) | 4 |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | E | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S1 | - | • | • | - | • | • | - | • | - | SM0 |
| S2 | - | • | • | - | • | • | - | • | - | |
| D1 | - | • | • | - | • | • | - | - | - | |

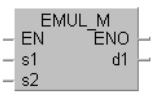
1) Kräver revision B av Q00(J) och Q01

**MELSEC Instruktionslista**



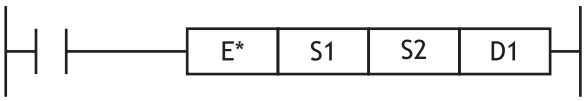

**Ladder**

**Funktionsblock**




## E*

Flyttalsvärdet i S1 multipliceras med flyttalsvärdet i S2 och resultatet lagras i D.




All käll- och destinationsdata måste vara i flyttalsformat. Det gäller även konstanter. Dessa skrivs på följande sätt:

– GX IEC Developer: 3.75E-3 (0,00375), 3.756E+2 (375,6)

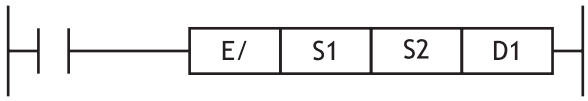
– GX Developer: E3.75-3 (0,00375), E3.756+2 (375,6)

Multiplikationen är matematiskt korrekt t ex $2.2 \times 10^{2} * (-1.2 \times 10^{-1}) = -0.234 \times 10^{2}$

**Observera!**
I GX IEC Developer skall S1, S2 och D1 som definieras som flyttalsvariabler (REAL)

# E/

Flyttalsvärdet i S1 divideras med flyttalsvärdet i S2 och resultatet lagras i D.




All käll- och destinationsdata måste vara i flyttalsformat. Det gäller även konstanter. Dessa skrivs på följande sätt:

– GX IEC Developer: 3.75E-3 (0,00375), 3.756E+2 (375,6)

– GX Developer: E3.75-3 (0,00375), E3.756+2 (375,6)

Divisionen är matematiskt korrekt t ex $2.3456 \times 10^{2} / (-5.6 \times 10^{-1}) = -4.188 \times 10^{2}$

**Observera!**
I GX IEC Developer skall S1, S2 och D1 som definieras som flyttalsvariabler (REAL)

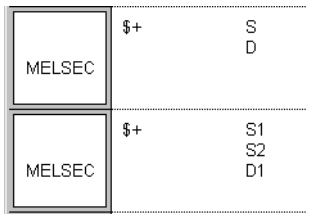
# 8.10.11 Addition av teckensträngar, $+

| $+, $+P | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● 1) | 3/4 2) |

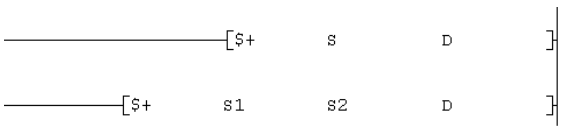
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | $ | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | - | • | • | - | - | - | - | • | - | SM0 |
| D | - | • | • | - | - | - | - | - | - | |
| S1 | - | • | • | - | - | - | - | • | - | |
| S2 | - | • | • | - | - | - | - | • | - | |
| D1 | - | • | • | - | - | - | - | - | - | |

1) Instruktionen kan ej användas i Q00(J) eller Q01
2) 3 steg med två operander, 4 steg med tre operander

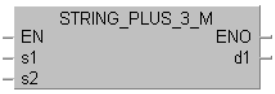
**MELSEC Instruktionslista**




**Ladder**




**Funktionsblock**




## $+

Alt. 1

Teckensträngen i S utökas med tecknen i D. Den sammansatta strängen lagras i D. Den sammansatta strängen börjar med tecknet i den minst signifikanta byten i D och slutar med koden “00H” i S. När två strängar sätts samman ignoreras den den första koden “00H”. Däremot markerar den andra koden “00H” slutet på den sammansatta strängen.

Alt. 2

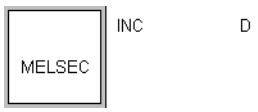
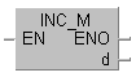
Teckensträngen i S2 läggs till teckensträngen i S1 och resultatet lagras i D. Den sammansatta strängen börjar med tecknet i den minst signifikanta byten i S1 och slutar med koden “00H” i S2. När två strängar sätts samman ignoreras den den första koden “00H”. Däremot markerar den andra koden “00H” slutet på den sammansatta strängen.

**Observera!**
I GX IEC Developer skall S1, S2 och D1 som definieras som strängvariabler (STRING)

# 8.10.12 Öka och minska operand med 1, INC, DEC

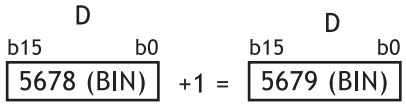
| INC, INCP, DEC, DECP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | • | • | • | 2 |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| D | • | • | • | • | • | • | • | - | - | - |

| MELSEC Instruktionslista | Ladder |
|---|---|
|   | |
| **Strukturerad text** | **Funktionsblock** |
| |   |

## INC

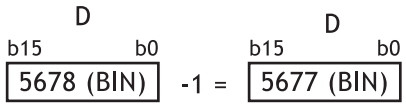
INC-instruktionen ökar angivet enkelord med 1. Om instruktionen endast skall utföras en gång när ingångsvillkoret blir uppfyllt skall P läggas till (positiv flank). Är innehållet i (D) lika med 32 767 blir resultatet -32 768.




**Observera!**
I GX IEC Developer skall D definieras som en enkelordsvariabel (t ex INT)

## DEC

DEC-instruktionen minskar angivet enkelord med 1. Om instruktionen endast skall utföras en gång när ingångsvillkoret blir uppfyllt skall P läggas till (positiv flank). Är innehållet i (D) lika med 0 blir resultatet -1.

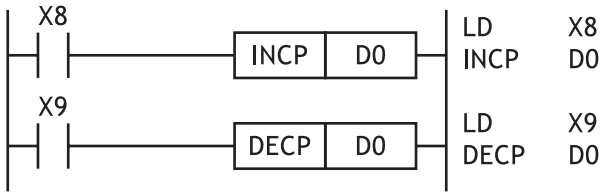



**Observera!**
I GX IEC Developer skall D definieras som en enkelordsvariabel (t ex INT)

## Programexempel

### INC, DEC

Program som ökar värdet i D0 med ett varje gång X8 ettställs och minskar värdet med ett varje gång X9 ettställs (positiv flank).



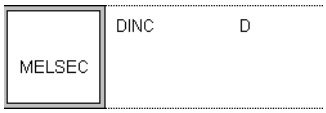
# 8.10.13 Öka och minska operand med 1, DINC, DDEC

| DINC, DINCP, DDEC, DDECP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 2 |

**Tillåtna operander**

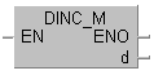
| | Interna kanaler | | R, ZR | J*/ * | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| D | • | • | • | • | • | • | • | - | - | - |

**MELSEC Instruktionslista**



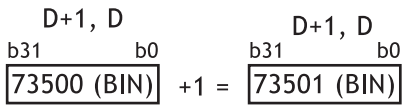

**Ladder**

**Funktionsblock**




## DINC

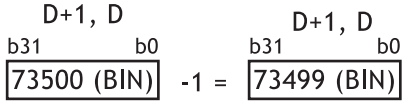
DINC-instruktionen ökar angivet dubbelord med 1. Om instruktionen endast skall utföras en gång när ingångsvillkoret blir uppfyllt skall P läggas till (positiv flank). Är innehållet i (D) lika med 2 147 483 647 blir resultatet -2 147 483 648.




**Observera!**
I GX IEC Developer skall D definieras som en dubbelordsvariabel (t ex DINT)

## DDEC

DDEC-instruktionen minskar angivet dubbelord med 1. Om instruktionen endast skall utföras en gång när ingångsvillkoret blir uppfyllt skall P läggas till (positiv flank). Om innehållet i (D) är 0, blir resultatet -1.

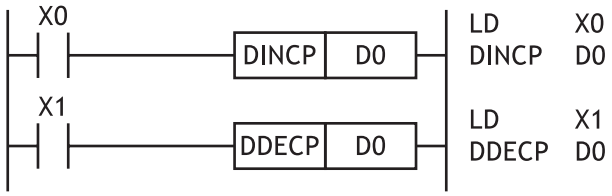



**Observera!**
I GX IEC Developer skall D definieras som en dubbelordsvariabel (t ex DINT)

## Programexempel

### DINC, DDEC öka/minska med ett

Program som ökar värdet i D0, D1 med ett när X0 ettställs och minskar värdet i D0, D1 när X1 ettställs.



# 8.11 Omvandlingsinstruktioner

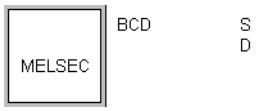
## 8.11.1 Omvandling till BCD-format, BCD

| **BCD, BCDP, DBCD, DBCDP** | **Q4AR** | **QnA(S)** | **Q** | **Programsteg** |
|---|---|---|---|---|
| | ● | ● | ● | 3 |

**Tillåtna operander**

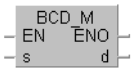
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | • | • | • | • | • | - | SM0 |
| D | • | • | • | • | • | • | • | - | - | - |

**MELSEC Instruktionslista**



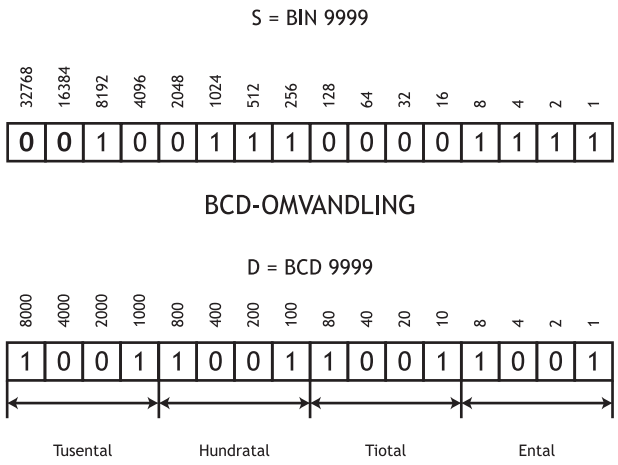

**Ladder**

**Funktionsblock**




## BCD

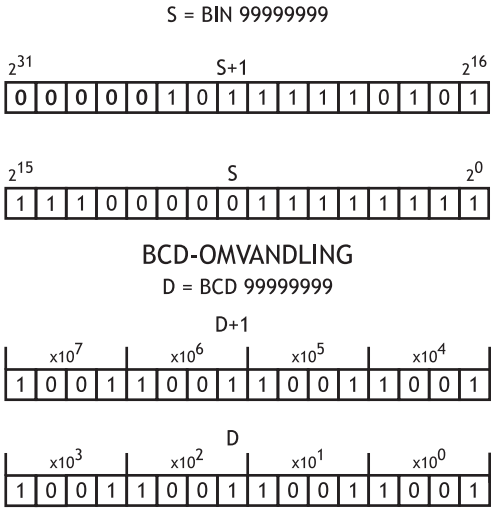
BCD-instruktionen omvandlar det i (S) angivna binärvärdet till ett BCD-värde (0 till 9999) och lagrar resultatet i (D). Om instruktionen endast skall utföras en gång när ingångsvillkoret blir uppfyllt skall P läggas till (positiv flank).




**Observera!**
De två högsta bitarna i binärvärdet i (S) måste vara 0, annars är värdet garanterat större än 9999. I GX IEC Developer skall D definieras som en dubbelordsvariabel (t ex DINT)

# DBCD

DBCD-instruktionen omvandlar det i (S) 8-siffriga binärvärdet (0 till 99999999) till ett BCD-värde och lagrar resultatet i (D).

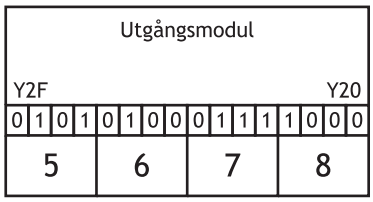


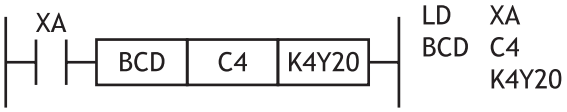

**Observera!**
De fem högsta bitarna i binärvärdet i (S) måste vara 0, annars är värdet garanterat större än 9999999. I GX IEC Developer skall D definieras som en dubbelordsvariabel (t ex DINT)

## Programexempel

### BCD

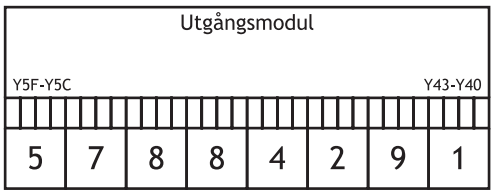
Räknarens C4 aktuella värde visas på displayen när XA är aktiverad (ettställd).

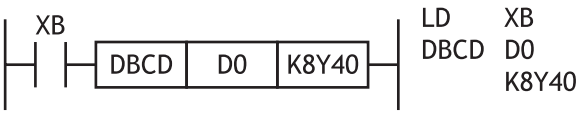


## DBCD

Innehållet i D0 och D1 presenteras på utgångarna Y40 till Y5F när XB är aktiverad.



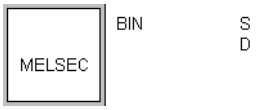
## 8.11.2 Omvandla till binärformat, BIN

| BIN, BINP, DBIN, DBINP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 3 |

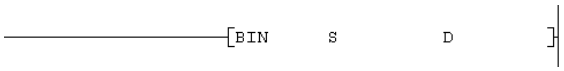
**Tillåtna operander**

| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | • | • | • | • | • | - | SM0 |
| D | • | • | • | • | • | • | • | - | - | |

**MELSEC Instruktionslista**



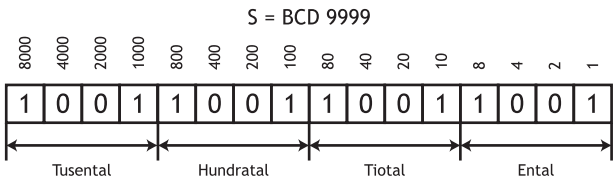

**Ladder**



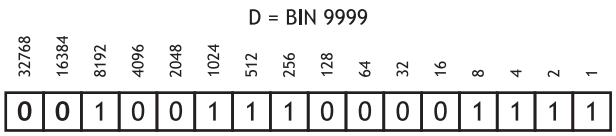

**Funktionsblock**

## BIN

BIN-instruktionen omvandlar det (S) angivna BCD-värdet (0 till 9999) till ett BIN-värde och lagrar resultatet i (D).



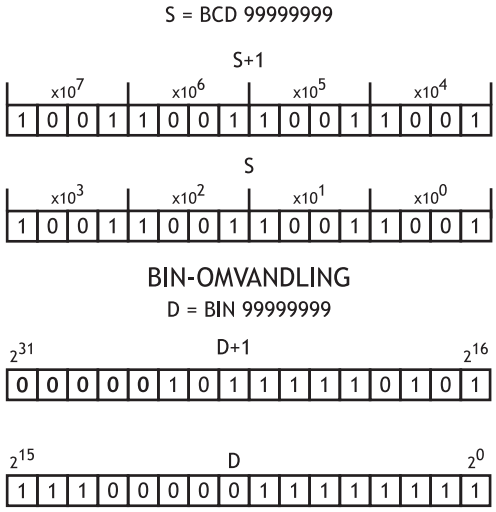

BIN-OMVANDLING




**Observera!**
De första två bitarna i binärvärdet i (S) sättes till 0.
Om en siffra i hämtadressen (S) inte ligger mellan 0 och 9 sätts felflaggan.
I GX IEC Developer skall S och D definieras som en enkelordsvariabel (t ex INT)

# DBIN

DBIN-instruktionen omvandlar det i (S) 8-siffriga BCD-värdet (0 till 99999999) till ett BIN-värde och lagrar resultatet i (D).

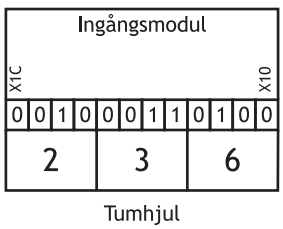


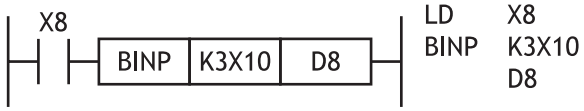

**Observera!**
De högsta fem bitarna i binärvärdet i (S) sätts till 0.
Om en siffra i hämtadressen (S) inte ligger mellan 0 och 9 sätts felflaggan.
I GX IEC Developer skall S och D definieras som en dubbelordsvariabel (t ex DINT)

## Programexempel

### BIN

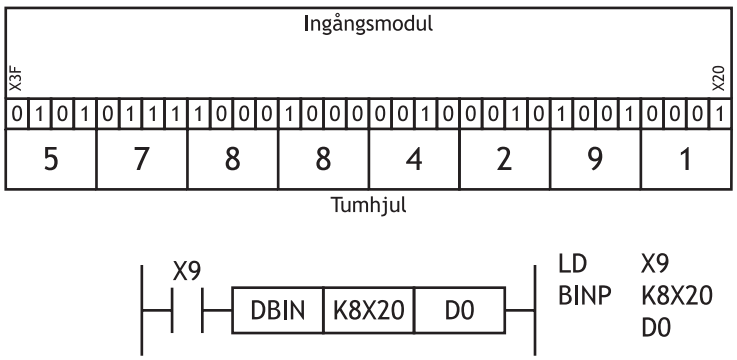
Programmet omvandlar BCD-värdet från ingångarna X10 till X1B till binärvärde och lagrar resultatet i D8 efter ettställning av X8.



## DBIN

Ingångsvärdena i X20 till X3F omvandlas till binärvärde och lagras i D0 och D1 när X9 aktiveras.



## 8.11.3 Omvandling till flyttalsformat, FLT

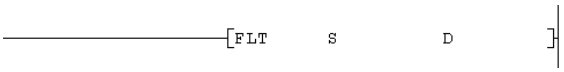
| FLT, FLTP, DFLT, DFLTP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ●1) | 3 |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | • | • | • | • | • | - | - |
| D | - | • | • | - | • | • | - | - | - | |

1) Kräver revision B av Q00(J) och Q01

**MELSEC Instruktionslista**

**Ladder**




**Funktionsblock**

# FLT

Med hjälp av FLT-instruktionen kan man omvandla ett heltal till ett flyttal.

Binärvärdet i (S), enkelord, omvandlas till ett flyttal som lagras i (D). Flyttal lagras alltid i två dataregister (32-bitar).

**Observera!**
I GX IEC Developer skall S definieras som en enkelordsvariabel (t ex INT) och D som flyttalsvariabel (REAL).

# DFLT

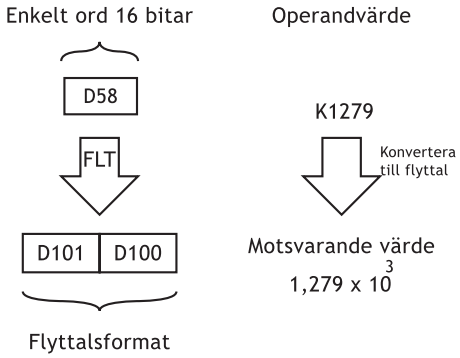
Med hjälp av DFLT-instruktionen kan man omvandla ett heltal till ett flyttal.

Binärvärdet i (S), dubbelord, omvandlas till ett flyttal som lagras i (D). Flyttal lagras alltid i två dataregister (32-bitar).

**Observera!**
I GX IEC Developer skall S definieras som en dubbelordsvariabel (t ex DINT) och D som flyttalsvariabel (REAL).

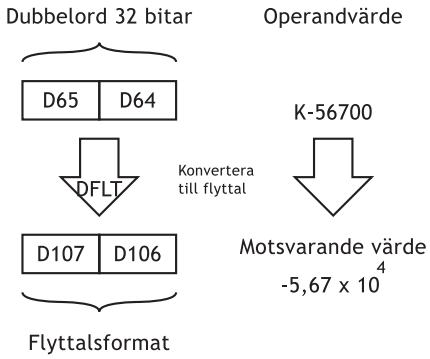
## Programexempel enkelord

När SM400 ettställs kommer instruktionen att omvandla innehållet i register D58 och lagra resultatet i D101 och D100 som ett flyttal.




## Programexempel dubbelord

När SM400 ettställs kommer instruktionen att omvandla innehållet i register D64, D65 och lagra resultatet i D107 och D106 som ett flyttal.



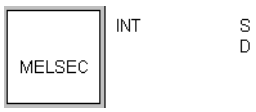
## 8.11.4 Omvandling till heltalsformat, INT

| INT, INTP, DINT, DINTP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● 1) | 3 |

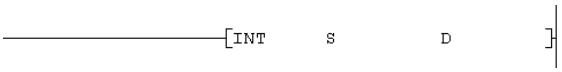
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | E | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | - | • | • | - | • | • | - | • | - | SM0 |
| D | • | • | • | • | • | • | • | - | - | |

1) Kräver revision B av Q00(J) och Q01

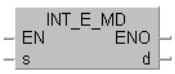
**MELSEC Instruktionslista**




**Ladder**




**Funktionsblock**




## INT

Flyttalsvärdet i S avrundas neråt till närmaste heltalsvärde och lagras i normal binär form i D.

Källdatan S är alltid ett dubbelord (32 bitar) i flyttalsformat. Efter omvandling lagras resultatet i D som ett enkelord. Om källdatan S inte är ett helt tal måste detta avrundas neråt. Om resultatet i D ligger utanför det tilllåtna området kommer SM0 felflagga att ettställas och markera overflow vilket innebär ogiltigt värde.

**Observera!**
I GX IEC Developer skall S definieras som en flyttalsvariabel (REAL) och D som en enkelordsvariabel (t ex INT).

# DINT

Flyttalsvärdet i S avrundas neråt till närmaste heltalsvärde och lagras i normal binär form i D.

Källdatan S är alltid ett dubbelord (32 bitar) i flyttalsformat. Efter omvandling lagras resultatet i D som ett dubbelord. Om källdatan S inte är ett helt tal måste detta avrundas neråt. Om resultatet i D ligger utanför det tilllåtna området kommer SM0 felflagga att ettställas och markera overflow vilket innebär ogiltigt värde.

---

**Observera!**
I GX IEC Developer skall S definieras som en flyttalsvariabel (REAL) och D som en dubbelordsvariabel (t ex DINT).

---

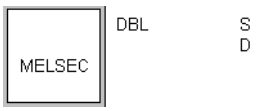
## 8.11.5 Omvandling till dubbelord, DBL

| DBL, DBLP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 3 |

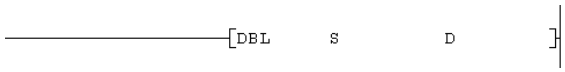
**Tillåtna operander**

| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | - | • | • | • | • | - | - |
| D | • | • | • | • | • | • | • | - | - | |

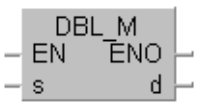
**MELSEC Instruktionslista**




**Ladder**

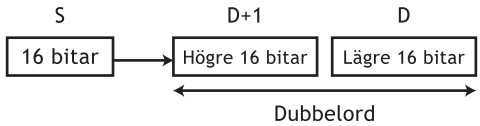



**Funktionsblock**




## DBL

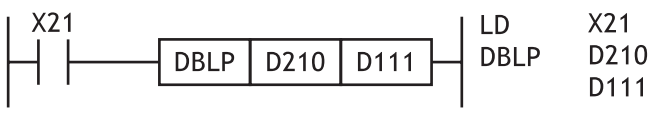
Omvandling av enkelord till dubbelord med teckenbit. S innehåller ett binärt enkelord och resultatet lagras som ett dubbelord i D+1, D.

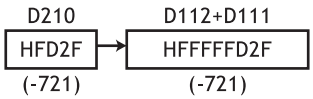



**Observera!**
I GX IEC Developer skall S definieras som en enkelordsvariabel (t ex INT) och D som en dubbelordsvariabel. (t ex DINT)

### Programexempel

När X21 ettställs omvandlas enkelordet D210 till ett dubbelord som lagras i D112 och D111.



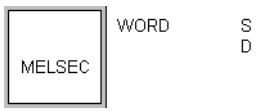



## 8.11.6 Omvandling till enkelord, WORD

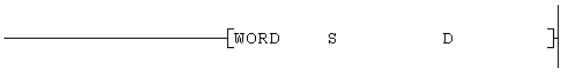
| WORD, WORDP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 3 |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | - | • | • | • | • | - | SM0 |
| D | • | • | • | • | • | • | • | - | - | |

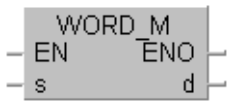
**MELSEC Instruktionslista**




**Ladder**

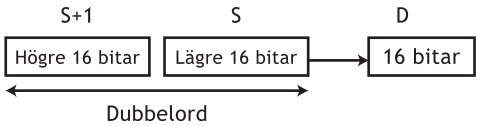



**Funktionsblock**




# WORD

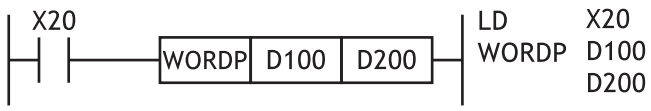
Omvandling av dubbelord till enkelord. S innehåller ett binärt dubbelord, (S+1, S) och resultatet lagras som ett enkelord i D.

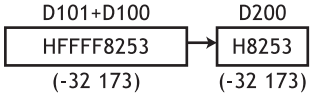



**Observera!**
I GX IEC Developer skall S definieras som en dubbelordsvariabel (t ex DINT) och D som en enkelordsvariabel (t ex INT).

### Programexempel

När X20 ettställs omvandlas dubbelordet i D101 och D100 till ett enkelord som lagras i D200.




Felflaggan ettställs om värdet i källregistret är utanför området -32 768 - 32 767.

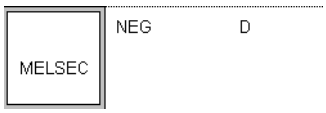
## 8.11.7 Ändra tecken på heltal, NEG

| NEG, NEGP, DNEG, DNEGP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | • | • | • | 2 |

**Tillåtna operander**

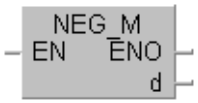
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| D | • | • | • | • | • | • | • | - | - | - |

**MELSEC Instruktionslista**



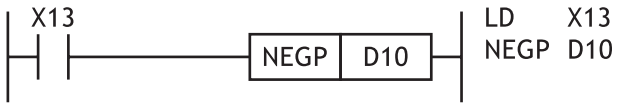

**Ladder**

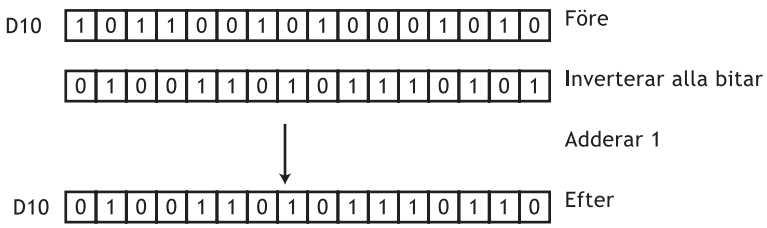
**Funktionsblock**




## NEG

Instruktionen används för att byta tecken (till negativt eller positivt värde), på ett enkelord.



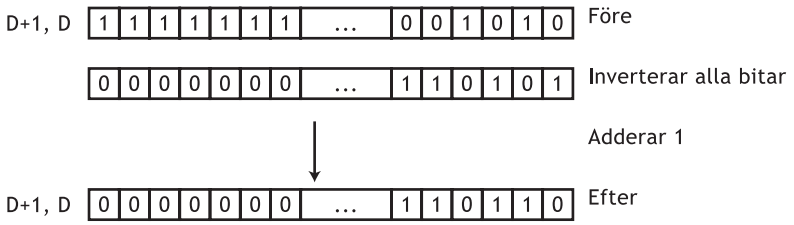




Först inverteras innehållet i D, dvs 1 blir 0 och 0 blir 1. Därefter adderas värdet ett och resultatet lagras i D.
Samma register används som käll-och destinationsregister (D). Därför kommer NEG-instruktionen utföra växlingen varje programvarv om man inte använder ett förreglingsprogram eller pulstillägg (P).

**Observera!**
I GX IEC Developer skall D definieras som en enkelordsvariabel (t ex INT).

# DNEG

Först inverteras de 32 bitarna varefter en etta adderas och resultatet lagras i D.




DNEG-instruktionen (2-komplement/NOTfunktion) ändrar tecknet för dubbelordet. Binärvärdet i D inverteras först, sedan adderas värdet ett. Resultatet lagras i D.

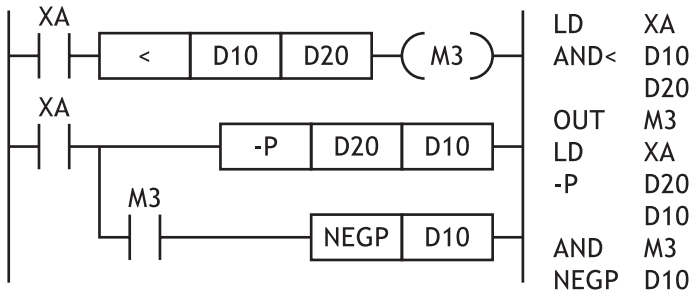
---

**Observera!**
I GX IEC Developer skall D definieras som en dubbelordsvariabel (t ex DINT).

---

## Programexempel

När XA ettställs, beräknas D20 minus D10 och absolutvärdet lagras i D10. M3 aktiveras när innehållet i D10 är mindre än innehållet i D20 när XA ettställs och NEGP-instruktionen gör värdet positivt.



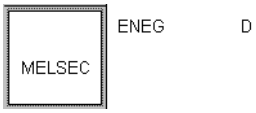
## 8.11.8 Ändra tecken på flyttal, ENEG

| ENEG, ENEGP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ●1) | 2 |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| D | - | • | • | - | • | • | • | - | - | - |

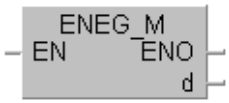
1) Kräver revision B av Q00(J) och Q01

**MELSEC Instruktionslista**




**Ladder**

**Funktionsblock**



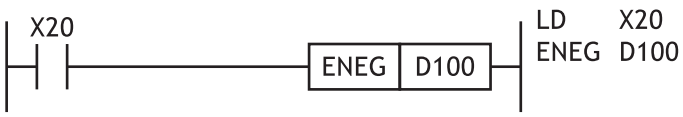

## ENEG

Teckenbyte för flyttalsdata som ändrar sig från positivt/negativt värde. Den logiska instruktionen för 2-komplement (teckenbyte) används för flyttalsdata.
Samma register används som käll-och destinationsregister (D). Därför kommer ENEG-instruktionen utföra växlingen varje programvarv om man inte använder ett förreglingsprogram eller pulsfunktionen (ENEGP).

**Observera!**
I GX IEC Developer skall D definieras som en flyttalsvariabel (REAL).

### Programexempel

När X20 blir ettställd växlar flyttalet i D101 och D100 tecken.



# 8.12 Dataöverföringsinstruktioner

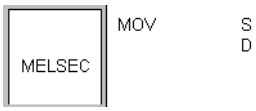
## 8.12.1 Överföring av binär data, MOV

| MOV, MOVP, DMOV, DMOVP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 3 |

**Tillåtna operander**

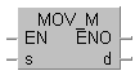
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | - | • | • | • | • | - | - |
| D | • | • | • | • | • | • | • | - | - | - |

**MELSEC Instruktionslista**



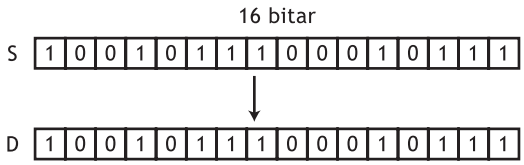

**Ladder**

**Funktionsblock**




### MOV

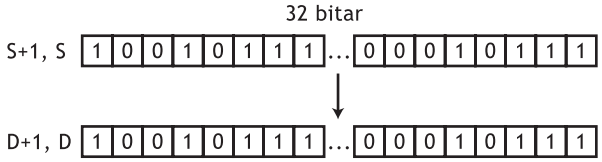
MOV-instruktionen kopierar enkelord från (S) till (D).




**Observera!**
I GX IEC Developer skall S och D definieras som en enkelordsvariabel (t ex INT)

### DMOV

DMOV-instruktionen kopierar dubbelord från (S) till (D).

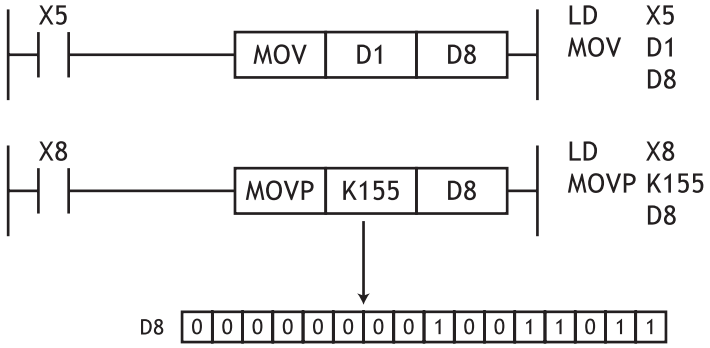



**Observera!**
I GX IEC Developer skall S och D definieras som en dubbelordsvariabel (t ex DINT)

## Programexempel

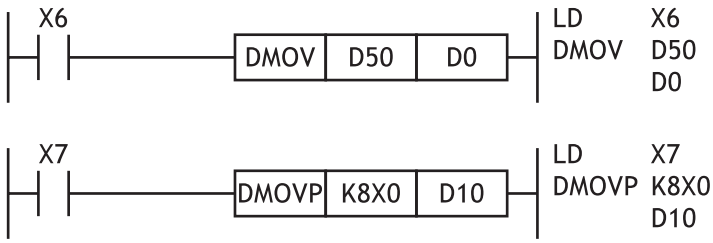
### MOV

Programmet överför innehållet i D1 till D8 så länge X5 är aktiverad. Det andra programmet överför konstanten 155 till D8 i binär form när X8 aktiveras (positiv flank).




### DMOV

Programmet överför data från D50 och D51 till D0 och D1 så länge X6 är aktiverad. Det andra programmet överför ingångarna X0 - X1F till D10 och D11 när X7 aktiveras (positiv flank).




Innehållet i hämtregistren påverkas ej av MOV-/DMOV-instruktionerna.

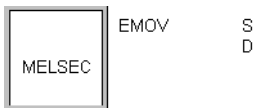
## 8.12.2 Överföring av flyttalsdata, EMOV

| EMOV, EMOVP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ●1) | 3 |

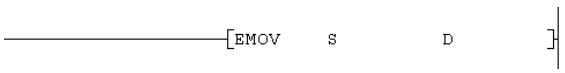
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | E | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | - | • | • | - | • | • | - | • | - | - |
| D | - | • | • | - | • | • | - | - | - | - |

1) Kräver revision B av Q00(J) och Q01

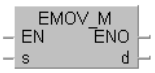
**MELSEC Instruktionslista**




**Ladder**

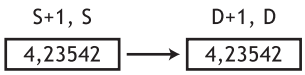



**Funktionsblock**




# EMOV(P)

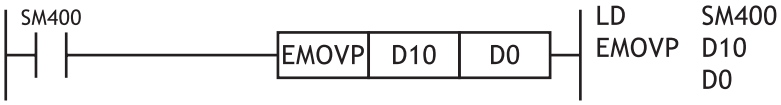
EMOV-instruktionen överför flyttalsdata i S till destinationen D.

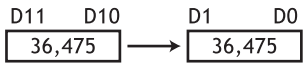



**Observera!**
I GX IEC Developer skall S och D definieras som en flyttalsvariabel (REAL)

### Programexempel

På positiv flank från SM400 överför flyttalsdata från två register D10 och D11 till D0 och D1.



## 8.12.3 Överföring av teckensträngar, $MOV

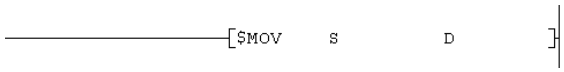
| $MOV, $MOVP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 3 |

**Tillåtna operander**

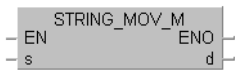
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | $ | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | - | • | • | - | - | - | - | • | - | SM0 |
| D | - | • | • | - | - | - | - | - | - | |

**MELSEC Instruktionslista**

**Ladder**

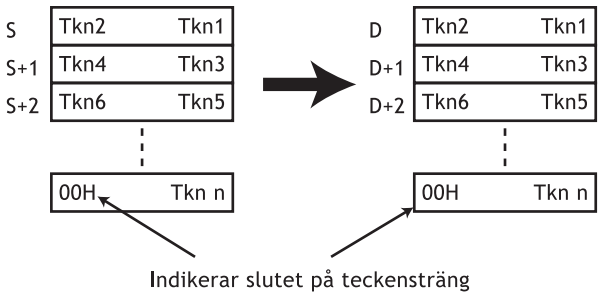



**Funktionsblock**




## $MOV

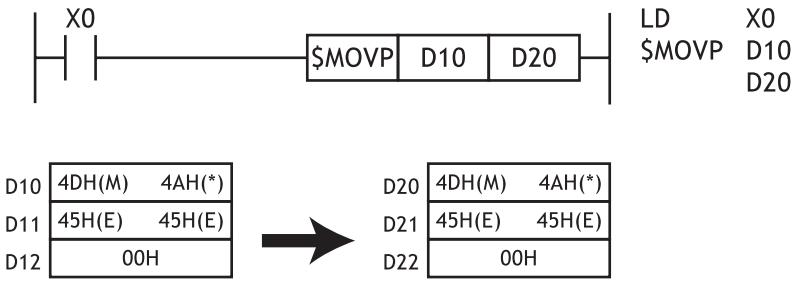
$MOV-instruktionen överför teckensträng i S till D tills koden “00H” träffas på, som betyder slut på strängen.




**Observera!**
I GX IEC Developer skall S och D definieras som strängvariabler (STRING)

## Programexempel

När X0 är ettställd (positiv flank) överförs strängen i D10-D12 till register D20-D22.



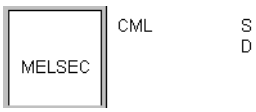
## 8.12.4 Invertering och överföring, CML

| CML, CMLP, DCML, DCMLP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | • | • | • | 3 |

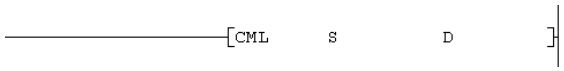
**Tillåtna operander**

| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | • | • | • | • | • | - | - |
| D | • | • | • | • | • | • | • | - | - | |

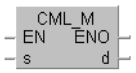
**MELSEC Instruktionslista**




**Ladder**

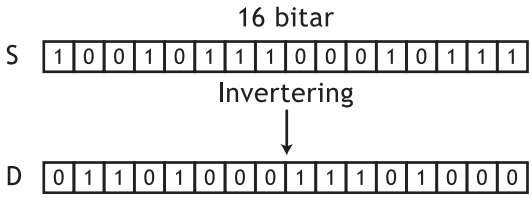



**Funktionsblock**




## CML

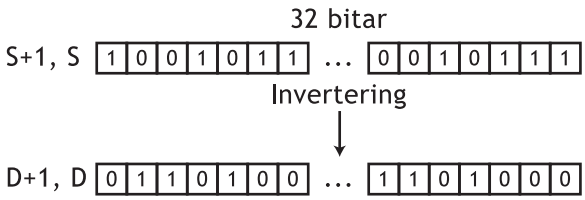
CML-instruktionen inverterar enkelord i (S) och överför dessa till (D).




**Observera!**
I GX IEC Developer skall S och D definieras som en enkelordsvariabel (t ex INT)

## DCML

DCML-instruktionen inverterar dubbelord i (S) och överför dessa till (D).

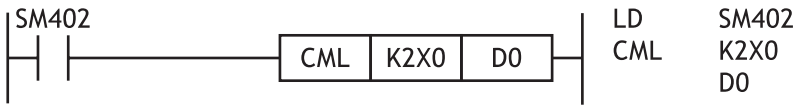


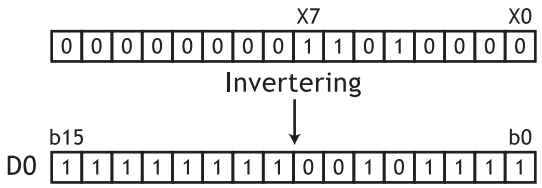

**Observera!**
I GX IEC Developer skall S och D definieras som en dubbelordsvariabel (t ex DINT)

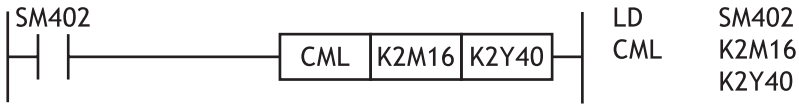
## Programexempel

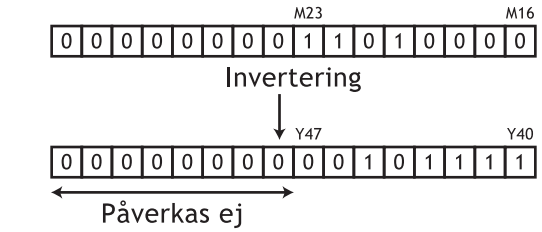
### CML

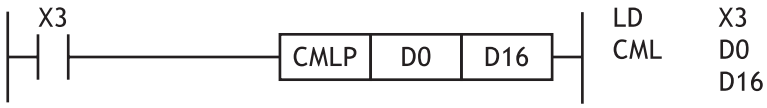
Programmet inverterar X0 till X7 och överför resultatet till D0. Resterande 8-bitar betraktas som 0. Nästa exempel inverterar M16 till M23 och överför resultatet till utgångarna Y40 till Y47. Sista exemplet utför vid ettställning av X3, invertering av innehållet i register D0 och överför resultatet till register D16.

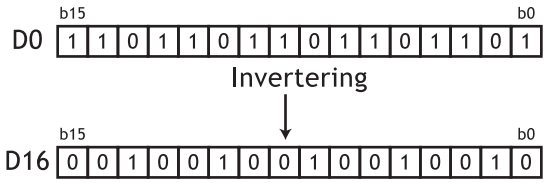


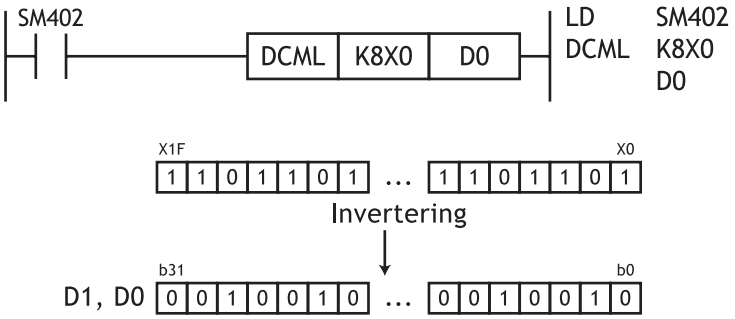



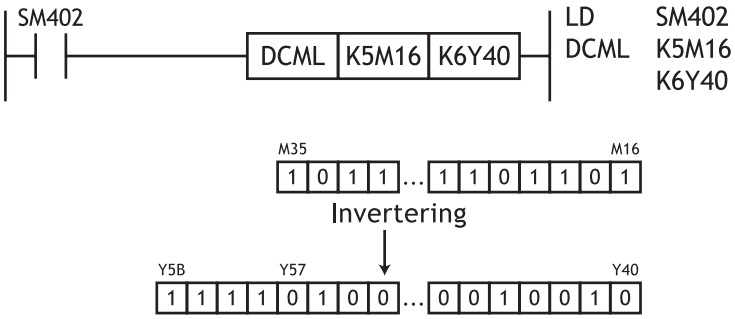



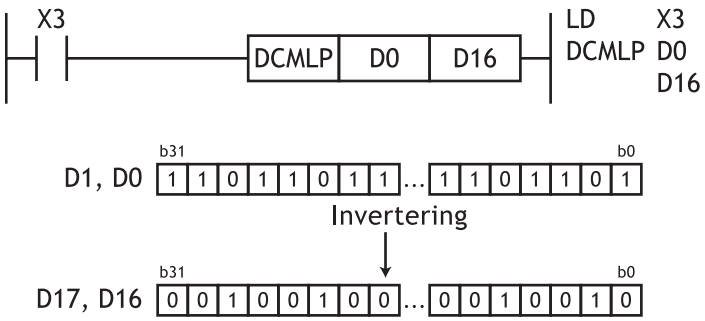



## DCML

I första exemplet inverteras X0 till X1F och överför resultatet till D0 och D1. Nästa exempel inverterar M16 till M35 och överför resultatet till utgångarna Y40 till Y53. Allmänt måste man beakta att antalet bitar i källan (S) måste vara mindre än eller lika med antalet bitar i destinationen (D).
Sista exemplet utför vid aktivering av X3 invertering av innehållet i registerna D0 och D1 resultatet överförs till registerna D16 och D17.



## 8.12.5 Blocköverföring, BMOV

| BMOV, BMOVP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 4 |

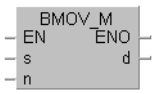
**Tillåtna operander**

| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | • | • | • | - | - | - | SM0 |
| D | • | • | • | • | • | • | - | - | - | |
| n | • | • | • | • | • | • | • | • | - | |

**MELSEC Instruktionslista**

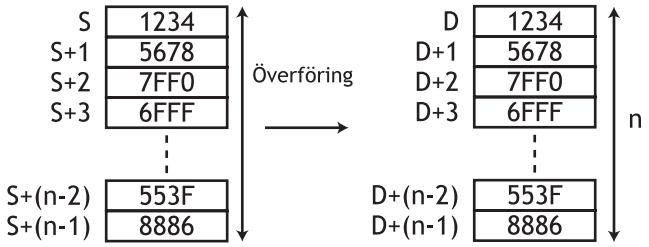
**Ladder**

**Funktionsblock**




## BMOV

Med hjälp av BMOV-instruktionen kan ett block med på varandra följande adresser överföras samtidigt. I S är den första adressen definierad och n anger antalet enkelord som skall överföras. Destinationsadressen börjar i D. Adressområdena kan överlappa varandra. En överföring till operander med lägre adresser börjar med S och överföring till operander med högre adresser börjar med S + n-1.

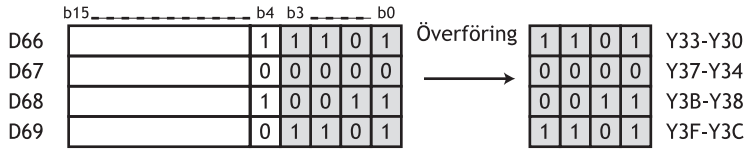



**Observera!**
I GX IEC Developer skall S, D och n definieras som en enkelordsvariabel (t ex INT)

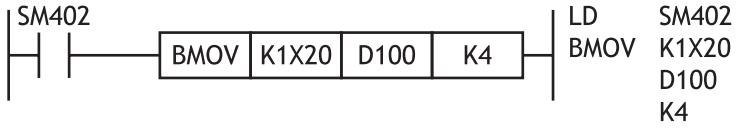
## Programexempel

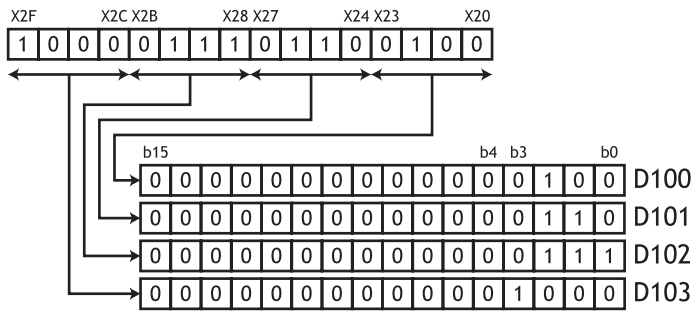
### BMOV

I det första exemplet kommer de lägsta 4 bitarna (b0-b3) i dataregister D66 till D69 att överföras till utgångarna Y30 till Y3F i block om 4 adresser.

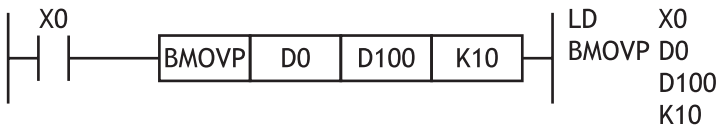



Det andra exemplet överför X20 till X2F på register D100 till D103.




Då X0 ettställs överförs innehållet i D0 - D9 till D100 - D109.



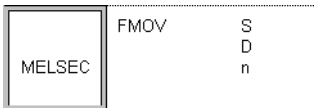
## 8.12.6 Blocköverföring, FMOV

| FMOV, FMOVP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 4 |

**Tillåtna operander**

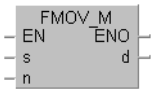
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | • | • | • | • | • | - | SM0 |
| D | • | • | • | • | • | • | - | - | - | |
| n | • | • | • | • | • | • | • | • | - | |

**MELSEC Instruktionslista**



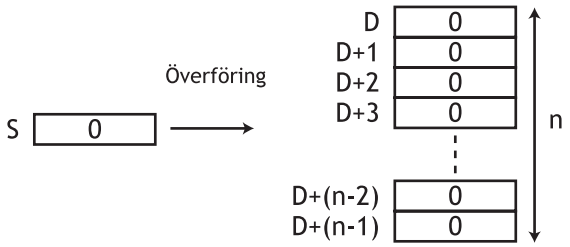

**Ladder**

**Funktionsblock**




## FMOV

FMOV-instruktionen överför operandinnehållet i (S) till n stycken enkelord i ett block med startadressen (D).

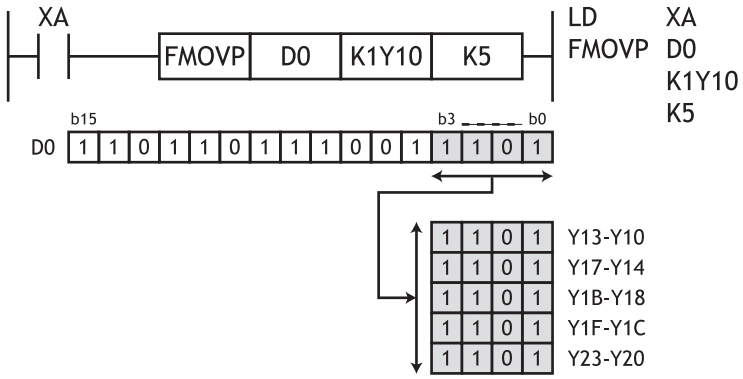



**Observera!**
I GX IEC Developer skall S, D och n definieras som en enkelordsvariabel (t ex INT)

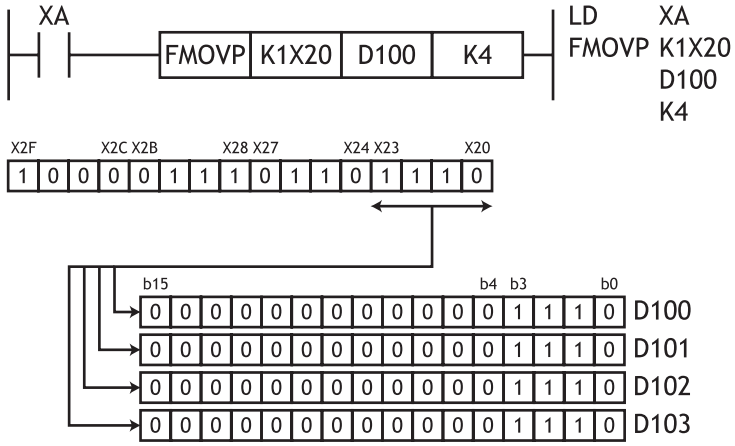
## Programexempel

### FMOV

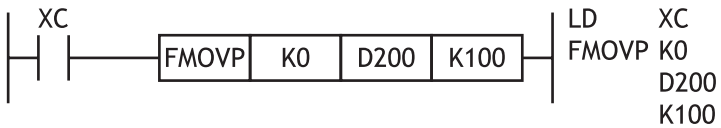
När XA ettställs överförs de 4 lägsta bitarna från D0 till utgångarna Y10 - Y23 till 5 adresser.




Intervallet X20 till X23 överförs till de 4 lägsta bitarna i D100 - D103 när XA ettställs.




När XC ettställs överförs värdet 0 till D200 - D299.



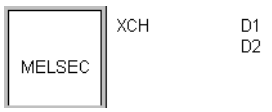
## 8.12.7 Växling av ord, XCH

| XCH, XCHP, DXCH, DXCHP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | • | • | • | 3 |

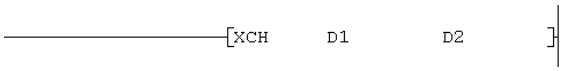
**Tillåtna operander**

| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| D1 | • | • | • | • | • | • | • | - | - | - |
| D2 | • | • | • | • | • | • | • | - | - | - |

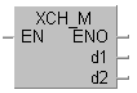
**MELSEC Instruktionslista**




**Ladder**

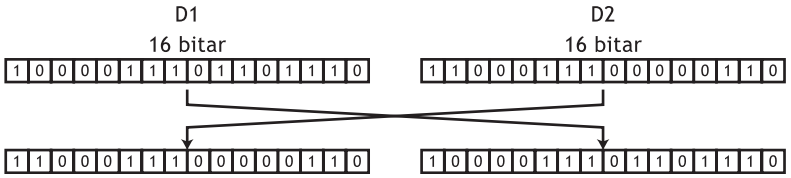



**Funktionsblock**




# XCH

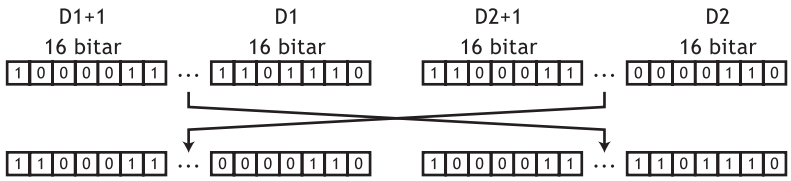
Med växlingsinstruktionen XCH byter innehållen i (D1) och (D2) plats med varandra.




**Observera!**
I GX IEC Developer skall D1och D2 definieras som en enkelordsvariabel (t ex INT)

# DXCH

Med växlingsinstruktionen DXCH byter innehållen i (D1+1, D1) och (D2+1, D2) plats med varandra.



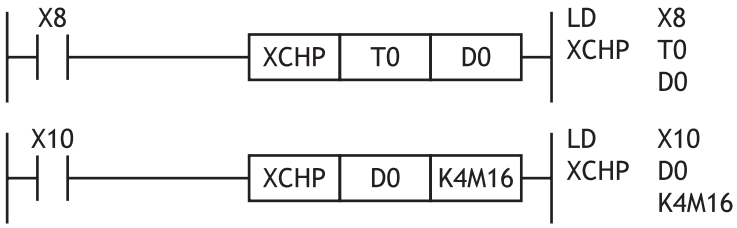

**Observera!**
I GX IEC Developer skall D1 och D2 definieras som en dubbelordsvariabel (t ex DINT)

Växling sker varje programvarv om man ej använder en programförregling eller instruktion med pulsfunktion.

## Programexempel

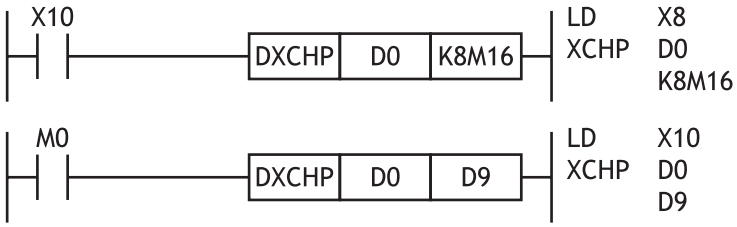
### XCH

Programmet byter efter ettställning av X8 innehållet i D0 mot ärvärdet i T0.
I andra exemplet kommer innehållet i D0 att bytas mot innehållet i M16 till M31 då X10 ettställs.




### DXCH

Programmet byter efter ettställning av X10 innehållet i D0 och D1 mot innehållet i M16 till M47. I andra exemplet byts innehållet i D0 och D1 med innehållet i D9 och D10 då M0 ettställs.

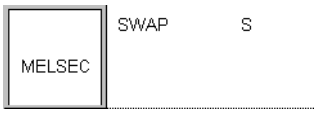


## 8.12.8 Växlar hög och låg byte i enkelord, SWAP

| SWAP, SWAPP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 3 |

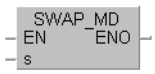
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | • | • | • | • | - | - | - |

**MELSEC Instruktionslista**



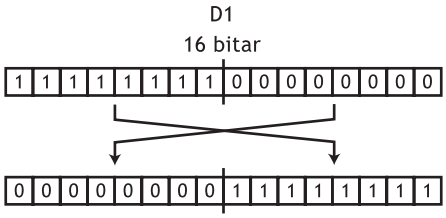

**Ladder**

**Funktionsblock**




## SWAP

Växlar höga och låga byten i enkelordet S.




**Observera!**
I GX IEC Developer skall S definieras som en enkelordsvariabel (t ex INT)

Växling sker varje programvarv om man ej använder en programförregling eller instruktion med pulsfunktion.

# 8.13 Programgrensinstruktioner

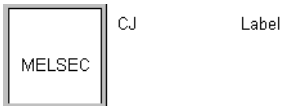
## 8.13.1 Programhopp, CJ, SCJ, JMP, GOEND

| CJ, SCJ, JMP, GOEND | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 2 |

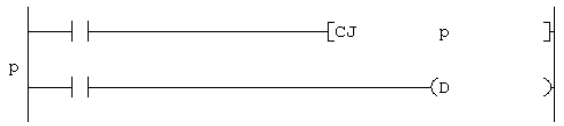
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | P | |
| P | - | - | - | - | - | - | - | - | ●[1)] | SM0 |

1) Ej GOEND

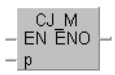
**MELSEC Instruktionslista**




**Ladder**




**Funktionsblock**




## CJ

Hoppinstruktionerna används tillsammans med pekaren P. Denna anger hoppdestinationen och har ett värde mellan P0 och P4095. Hopp utförs när ingångsvillkoret är uppfyllt.

## SCJ

Fördröjd hoppinstruktion.
Hopp utförs till pekaren P men fördröjs ett programvarv efter att ingångsvillkoret har aktiverats. Ingångsvillkoret kan i detta fall användas för att bryta upp OUT-instruktioner i det programavsnitt som kommer att hoppas över i efterföljande programvarv.

## JMP

Ovillkorligt hopp till pekaren P.

## GOEND

Hopp till slutet av programdelen

## Anmärkningar

Om en tidsfunktion är aktiverad fortsätter denna att räkna även om den omfattas av en hoppinstruktion (CJ, SCJ, JMP). Med en CJ-, SCJ- eller JMP-instruktion är det möjligt att hoppa bakåt i programmet. Sker detta kontinuerligt, förlängs cykeltiden och risk finns att Watch Dog Timern (cykeltidsövervakningen) löser ut.
Hopp framåt i programmet används för att förkorta cykeltiden.
I GX IEC Developer får en pekaradress inte som definieras som en global variabel.
Hopp inom en och samma POU löser man med “Network Labels”.

## Exempel på felkällor

PLC-systemet stoppar och felindikerar i följande fall.

– Pekare med samma nummer är inprogrammerade.

– Pekaradress saknas.

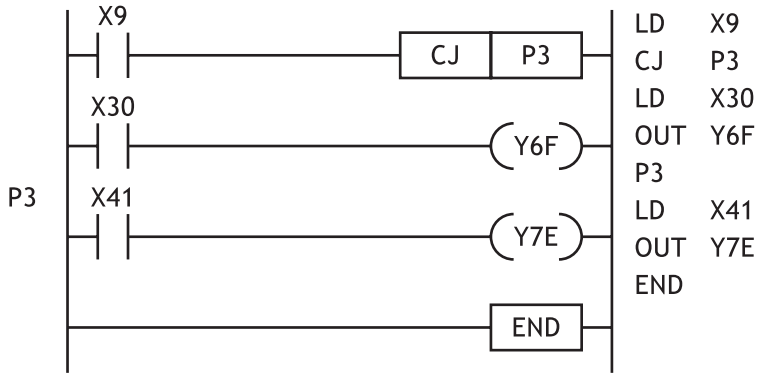
– Hopp har utförts till en adress inne i en FOR/NEXT loop.

– Hopp har skett till en pekaradress placerad efter END instruktion.

## Programexempel

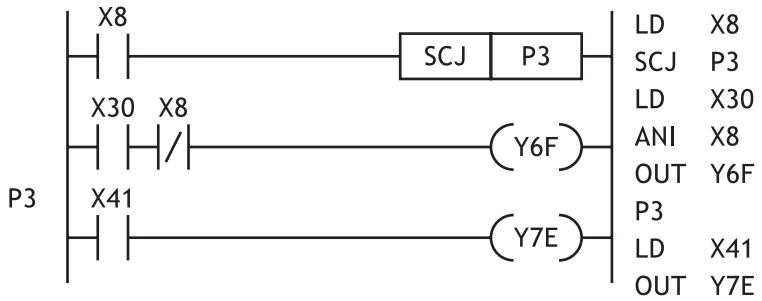
### CJ

Program som orsakar ett hopp till END så länge X9 aktiveras.



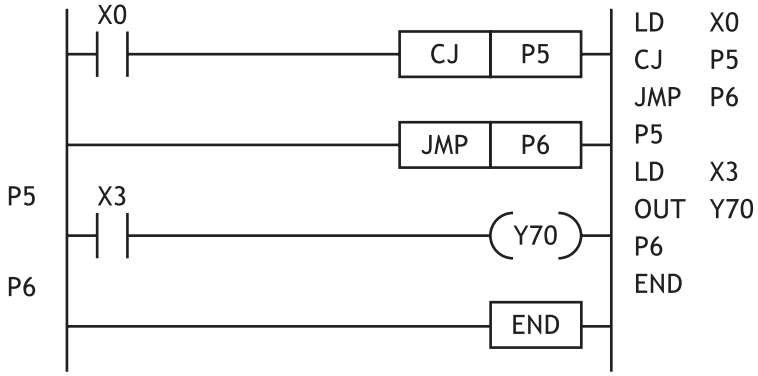

### SCJ

Program som orsakar ett hopp till P3 nästa programvarv när X8 aktiveras.



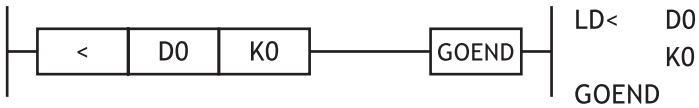
### JMP

Program som orsakar ett ovillkorligt hopp till END när X0 är nollställd.




### GOEND

Program som orskar ett hopp till programdelens END eller FEND när värdet i D0 är mindre än noll.



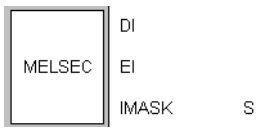
# 8.14 Programkontrollinstruktioner

## 8.14.1 Styrning av interruptprogram, EI, DI, IMASK

| EI, DI, IMASK | | | | | | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | ● | ● | ● | 2 1)/1 2) |
| Tillåtna operander | | | | | | | | | |
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | - | •1) | •1) | - | • | • | - | - | - | - |
| - | - | - | - | - | - | - | - | - | - | - |

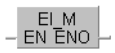
1) Endast IMASK
2) ENdast EI och DI

**MELSEC Instruktionslista**




**Ladder**

**Funktionsblock**




## DI

Instruktionen (disable interrupt) förreglar och förhindrar hopp till interruptprogram.

## EI

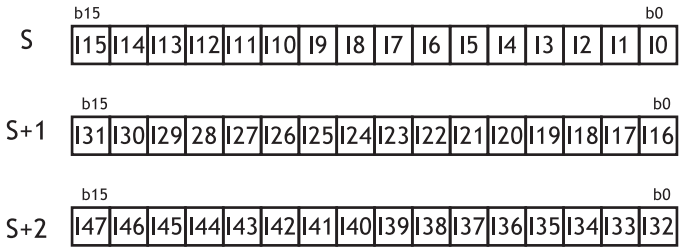
I0-I255 i Q, I0-I47 i QnA(S).
Ett interruptprogram är en infogad programdel med en definierad interruptadress I** som anropas genom en extern interruptsignal, en interruptsignal från en funktionsmodul eller ett tidsintervall. Instruktionen (enable interrupt) ger möjlighet till hopp till interruptprogram som aktiverats med IMASK-instruktionen. Mellan DI och EI kan inget interrupt utföras, interruptsignalen sparas och utörs efter EI.

## IMASK

Med hjälp av bitmönstret i S (16 dataregister i MELSEC Q, 3 dataregister i QnA) bestäms vilka interrupt som får utföras då EI-instruktionen exekveras. Om interruptpekarens bit är TILL kan interruptprogrammet utföras, om biten är FRÅN blockeras interruptet.

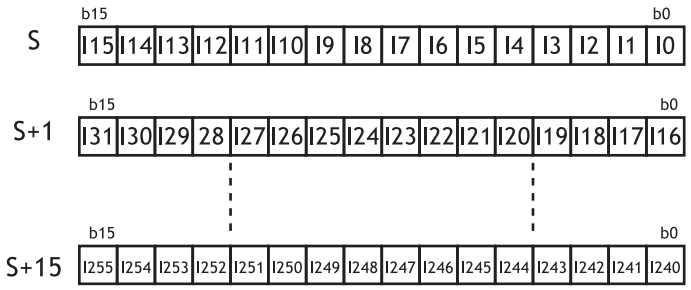
## Programexempel

### QnA




Om t ex bit 6 i S ettställs kan interrupt I6 exekveras. Nollställs biten kan I6 inte exekveras.

### Q




Om t ex bit 6 i S+1 ettställs kan interrupt I22 exekveras. Nollställs biten kan I22 inte exekveras.

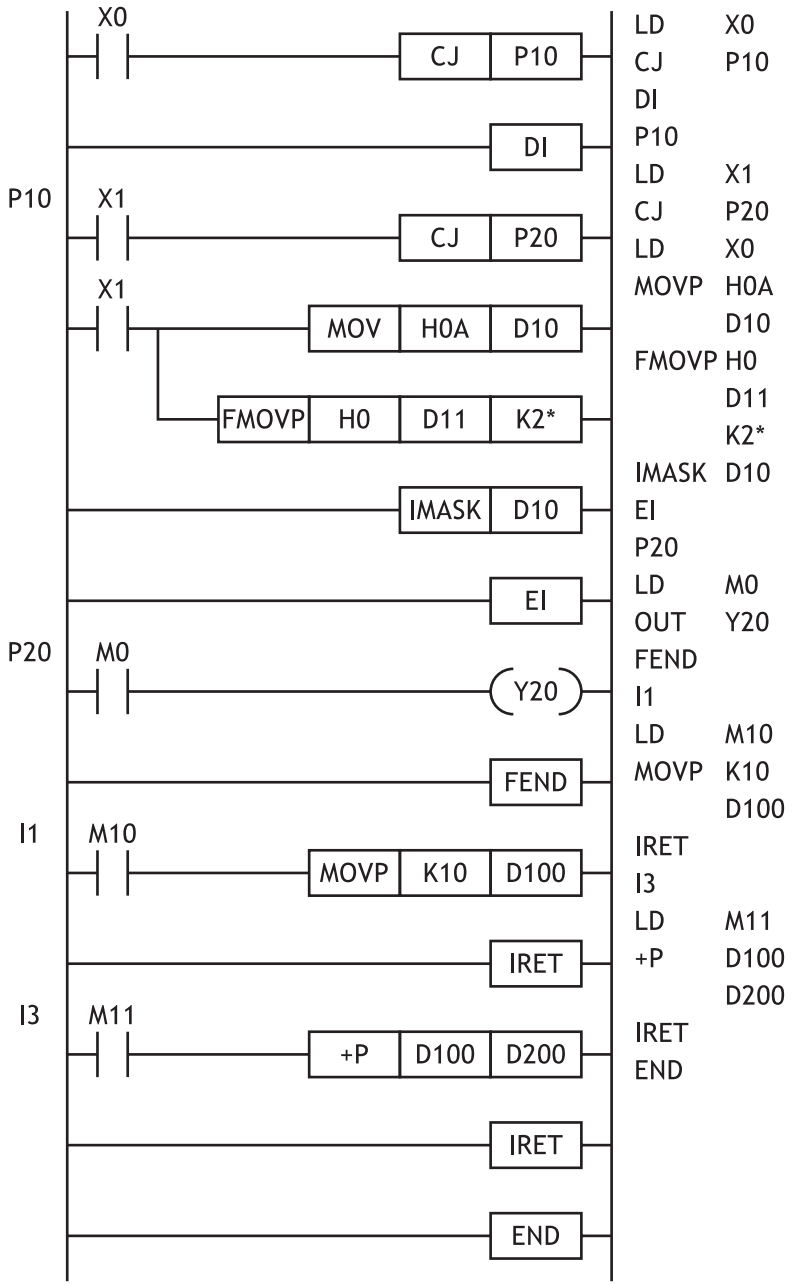
---

**Observera!**
När PLC-systemet går till RUN-läge är I0-I31 aktiverade och I32-I255 blir avaktiverade.

---

## EI, DI, IMASK

Förhindra respektive tillåta anrop av interruptprogram. När X0 ettställs tillåts interruptprogrammet utföras. När X0 nollställs blockeras interruptprogrammet.




**K15 med MELSEC Q*

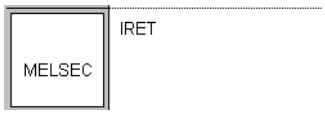
Utförs en kommunikationsuppdatering när ett interrupt erhålls, avbryts uppdateringen och interruptprogrammet bearbetas.

## 8.14.2 Slut på interruptprogram, IRET

| IRET | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 1 |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| - | - | - | - | - | - | - | - | - | - | SM0 |

**MELSEC Instruktionslista**




**Ladder**

**Funktionsblock**

# IRET

Betecknar slutet på ett interruptprogram och återgång till PLC programmet sker. Alla interruptrutiner måste avslutas med en IRET-instruktion. Om räknare behövs i programmet använd interruppträknaren (anges i parametrar). Vid bearbetning av ett interruptprogram aktiveras DI och innebär att flera interruptprogram inte kan bearbetas samtidigt.

Kommer IRET-instruktionen före interruptadressen I** stoppar programbearbetningen och felflaggan SM0 aktiveras.

# 8.15 Uppdateringsinstruktioner

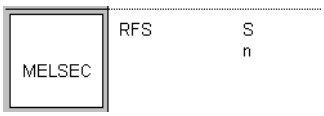
## 8.15.1 Uppdatering av in-/utgångar, RFS

| RFS, RFSP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 3 |

| Tillåtna operander | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • 1) | - | - | - | - | - | - | - | - | SM0 |
| n | • | • | • | • | • | • | • | • | - | |

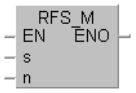
(1) Endast X och Y

**MELSEC Instruktionslista**




**Ladder**

**Funktionsblock**




## RFS

Uppdatererar in-och utgångarna i ett I/O område under ett scanvarv.

Vid normal I/O-uppdatering, utan RFS-instruktionen uppdateras in- och utgångarna efter END-instruktionen. Genom att använda sig av RFS-instruktionen kan status på in- och utgångar uppdateras under pågående programvarv. I S anges vilken in- eller utgång som skall uppdateras, i n anges hur många signaler som skall uppdateras.

Om adresstyperna DX/DY används uppdateras in-/utgångarna bit för bit.

# 8.16 Logiska instruktioner

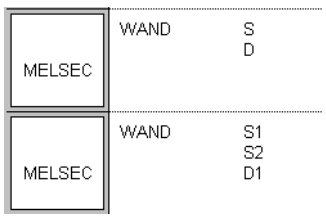
## 8.16.1 Logisk AND, WAND

| WAND, WANDP, DAND, DANDP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 3/4 1) |

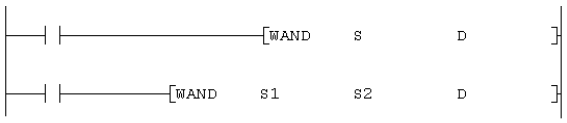
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | • | • | • | • | • | - | - |
| D | • | • | • | • | • | • | • | - | - | - |
| **WAND** | | | | | | | | | | |
| S1 | • | • | • | • | • | • | • | • | - | - |
| S2 | • | • | • | • | • | • | • | • | - | - |
| D1 | • | • | • | • | • | • | • | - | - | - |
| **DAND** | | | | | | | | | | |
| S1 | • | • | • | • | • | • | • | • | - | - |
| S2 | • | • | • | • | • | • | • | • | - | - |
| D | • | • | • | • | • | • | • | - | - | - |

1) 3 steg med två operander, 4 steg med tre operander

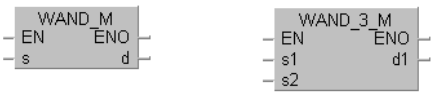
**MELSEC Instruktionslista**




**Ladder**

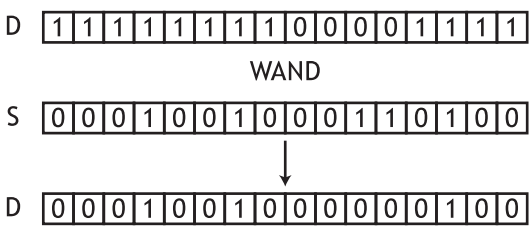



**Funktionsblock**



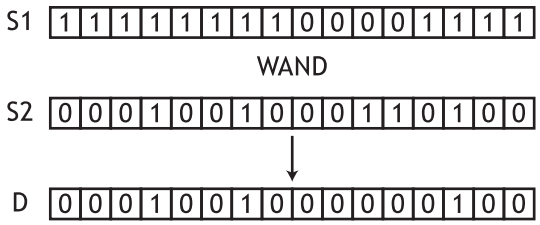

## WAND (2 operander)

WAND bildar en logisk produkt (AND) av två enkelord (D) och (S) och lagrar resultatet i (D). (S) förblir opåverkad. DAND utförs på samma sätt men med dubbelord.



# WAND (3 operander)

WAND bildar en logisk produkt (AND) av två enkelord (S1) och (S2) och lagrar resultatet i (D1). (S1) och (S2) förblir opåverkade. DAND utförs på samma sätt men med dubbelord.




**Observera!**
**WAND**
I GX IEC Developer skall S, S1, S2, D och D1 definieras som en enkelordsvariabel (t ex INT)
**DAND**
I GX IEC Developer skall S, S1, S2, D och D1 definieras som en dubbelordsvariabel (t ex DINT)

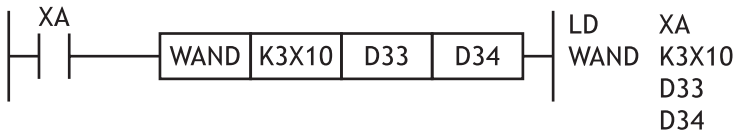
## Sanningstabell

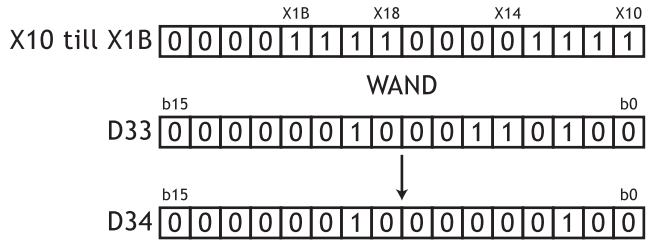
| D (S1) | S (S2) | D (D1) |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

## Programexempel

### WAND

En logisk produkt skall bildas mellan X10 till X1B och data i D33. Resultatet lagras i D34.



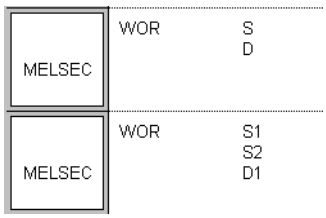
## 8.16.2 Logisk OR, WOR

| WOR, WORP, DOR, DORP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | • | • | • | 3/4 1) |

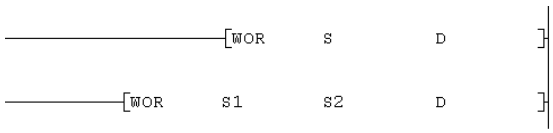
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | • | • | • | • | • | - | - |
| D | • | • | • | • | • | • | • | - | - | - |
| **WOR** | | | | | | | | | | |
| S1 | • | • | • | • | • | • | • | • | - | - |
| S2 | • | • | • | • | • | • | • | • | - | - |
| D1 | • | • | • | • | • | • | • | - | - | - |
| **DOR** | | | | | | | | | | |
| S1 | • | • | • | • | • | • | • | • | - | - |
| S2 | • | • | • | • | • | • | • | • | - | - |
| D | • | • | • | • | • | • | • | - | - | - |

1) 3 steg med två operander, 4 steg med tre operander

**MELSEC Instruktionslista**



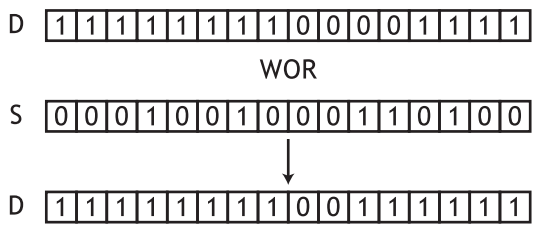

**Ladder**




**Funktionsblock**

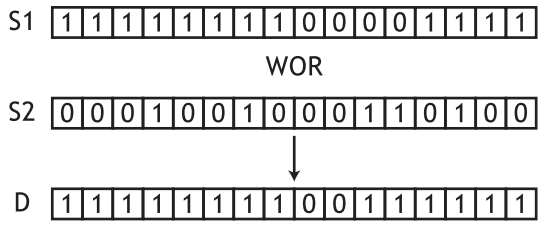
## WOR (2 operander)

WOR bildar en logisk summa (adderas bitvis) av två enkelord (D) och (S) och lagrar resultatet i (D). (S) förblir opåverkad. DOR utförs på samma sätt men med dubbelord.



# WOR (3 operander)

WOR bildar en logisk summa av två enkelord (S1) och (S2) och lagrar resultatet i (D1). (S1) och (S2) förblir opåverkade. DOR utförs på samma sätt men med dubbelord.




**Observera!**
**WOR**
I GX IEC Developer skall S, S1, S2, D och D1 definieras som en enkelordsvariabel (t ex INT)
**DOR**
I GX IEC Developer skall S, S1, S2, D och D1 definieras som en dubbelordsvariabel (t ex DINT)

## Sanningstabell

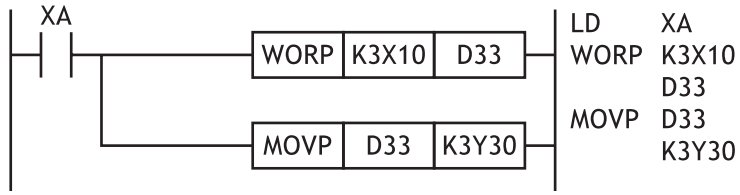
| D (S1) | S (S2) | D (D1) |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

## Programexempel

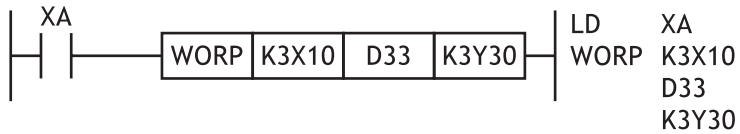
### WOR

En logisk summa skall bildas mellan två enkelord. I programmet adderas data i D10 till D20 och lagrar resultatet i D10.

Datan i X10 till X1B (12 bitar) adderas till data i D33. Resultatet överförs till utgångarna Y30 - Y3B (12 bitar).




När XA ettställs adderas ingångarna X10 - X1B (12 bitar) till datan i D33. Resultatet överförs till utgångarna Y30 - Y3B (12 bitar).

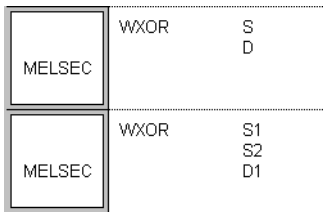


## 8.16.3 Logisk exklusiv OR, WXOR

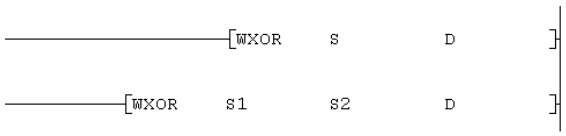
| WXOR, WXORP, DXOR, DXORP | | | | | | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | ● | ● | ● | 3/4 [1] |
| **Tillåtna operander** | | | | | | | | | |
| | **Interna kanaler** | | **R, ZR** | **J*/*** | | **U*\G*** | **Zn** | **K, H(16#)** | **Övriga** | **Flaggor** |
| | **Bitar** | **Ord** | | **Bitar** | **Ord** | | | | | |
| **S** | • | • | • | • | • | • | • | • | - | - |
| **D** | • | • | • | • | • | • | • | - | - | - |
| **WOR** | | | | | | | | | | |
| **S1** | • | • | • | • | • | • | • | • | - | - |
| **S2** | • | • | • | • | • | • | • | • | - | - |
| **D1** | • | • | • | • | • | • | • | - | - | - |
| **DOR** | | | | | | | | | | |
| **S1** | • | • | • | • | • | • | • | • | - | - |
| **S2** | • | • | • | • | • | • | • | • | - | - |
| **D** | • | • | • | • | • | • | • | - | - | - |

1) 3 steg med två operander, 4 steg med tre operander

**MELSEC Instruktionslista**



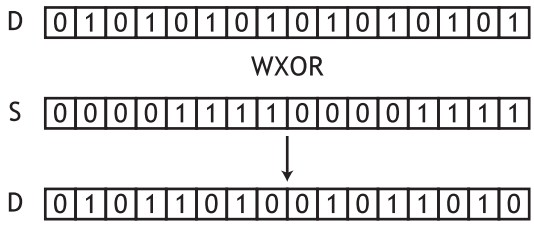

**Ladder**




**Funktionsblock**

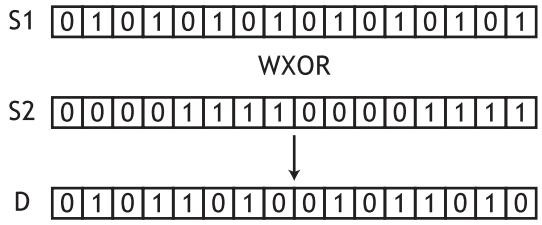
# WXOR (2 operander)

En logisk exklusiv ELLER bildas av två enkelord (D) och (S) och lagrar resultatet i (D). DXOR utförs på samma sätt men med dubbelord.



# WXOR (3 operander)

WXOR bildar en logisk exklusiv ELLER av två enkelord (S1) och (S2) och lagrar resultatet i (D1). DXOR utförs på samma sätt men med 32 bitar.




**Observera!**
**WXOR**
I GX IEC Developer skall S, S1, S2, D och D1 definieras som en enkelordsvariabel (t ex INT)
**DXOR**
I GX IEC Developer skall S, S1, S2, D och D1 definieras som en dubbelordsvariabel (t ex DINT)

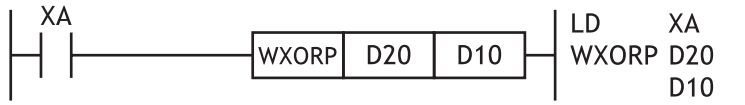
## Sanningstabell

| D (S1) | S (S2) | D (D1) |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

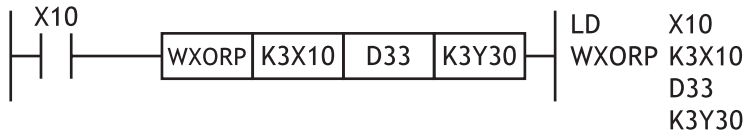
## Programexempel

### WXOR

I programmet utförs exklusiv ELLER mellan data i D10 och D20 och lagrar resultatet i D10.




När X10 ettställs utförs en exklusiv ELLER mellan ingångarna X10 - X1B och datan i D33. Resultatet överförs till utgångarna Y30 -Y3B.

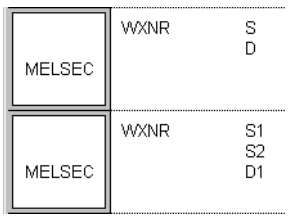


## 8.16.4 Logisk exklusiv NOR, WXNR

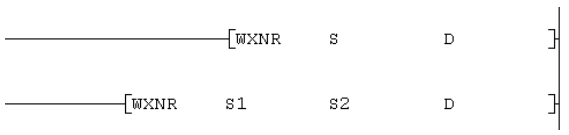
| WXNR, WXNRP, DXNR, DXNRP | | | | | | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | ● | ● | ● | 3/4 [1] |
| **Tillåtna operander** | | | | | | | | | | | |
| | Interna kanaler | | R, ZR | J** | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | • | • | • | • | • | - | - |
| D | • | • | • | • | • | • | • | - | - | - |
| **WXNR** | | | | | | | | | | |
| S1 | • | • | • | • | • | • | • | • | - | - |
| S2 | • | • | • | • | • | • | • | • | - | - |
| D | • | • | • | • | • | • | • | - | - | - |
| **DXNRP** | | | | | | | | | | |
| S1 | • | • | • | • | • | • | • | • | - | - |
| S2 | • | • | • | • | • | • | • | • | - | - |
| D1 | • | • | • | • | • | • | • | - | - | - |

1) 3 steg med två operander 4 steg med tre operander

**MELSEC Instruktionslista**



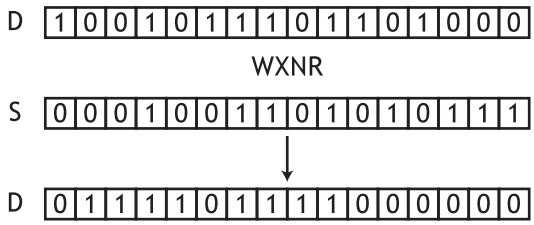

**Ladder**




**Funktionsblock**

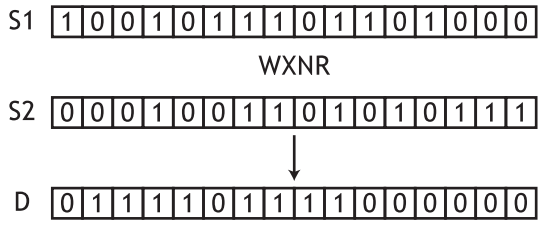
## WXNR (2 operander)

WXNR bildas av två enkelord (D) och (S) och lagrar resultatet i (D). DXNR utförs på samma sätt men med dubbelord.



# WXNR (2 operander)

WXNR bildar en logisk exklusiv NOR med två enkelord (S1) och (S2) och lagrar resultatet i (D1). DXNR utförs på samma sätt men med dubbelord.




**Observera!**
**WXNR**
I GX IEC Developer skall S, S1, S2, D och D1 definieras som en enkelordsvariabel (t ex INT)
**DXNR**
I GX IEC Developer skall S, S1, S2, D och D1 definieras som en dubbelordsvariabel (t ex DINT)

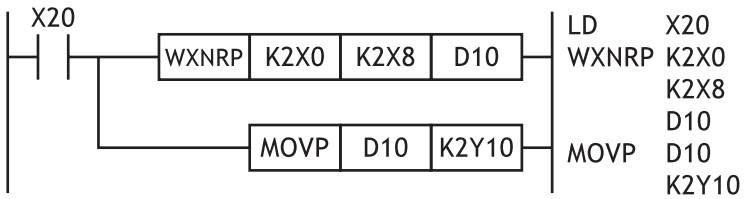
## Sanningstabell

| D (S1) | S (S2) | D (D1) |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

## Programexempel

### WXNR

Med hjälp av instruktionen exklusiv NOR ska data på ingångarna X0 till X7 knytas till data på ingångarna X8 till XF och resultatet läggas i D10 när X20 ettställs. Data i D10 överförs till utgångarna Y10 till Y17.



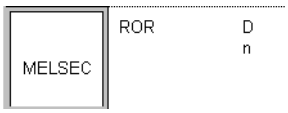
# 8.17 Rotationsinstruktioner

## 8.17.1 Rotation åt höger, ROR, RCR

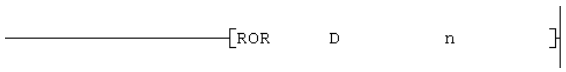
| ROR, RORP, RCR, RCRP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | • | • | • | 3 |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| D | • | • | • | • | • | • | • | - | - | - |
| n | • | • | • | • | • | • | • | • | - | |

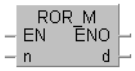
**MELSEC Instruktionslista**




**Ladder**

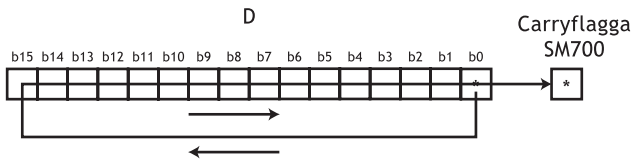



**Funktionsblock**



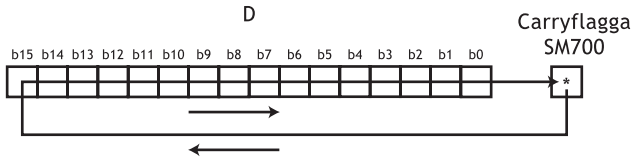

## ROR

Roterar databitarna i D n bitar till höger utan carryflagga. Carryflaggan har samma status som biten som sist flyttas från b0 till b15.




## RCR

Roterar databitarna i D n bitar till höger med carryflagga. Carryflaggan behåller status på biten som roteras n bitar.



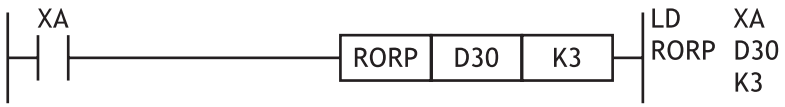

Rotation utförs så länge ingångsvillkoret är TILL om inget förreglingsprogram eller pulstillägg (P) används.

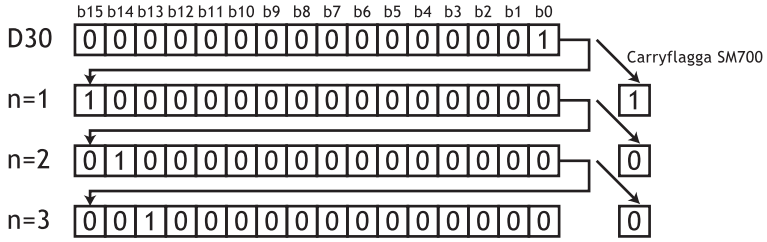
**Observera!**
I GX IEC Developer skall D och n definieras som en enkelordsvariabel (t ex INT)

## Programexempel

### RORP

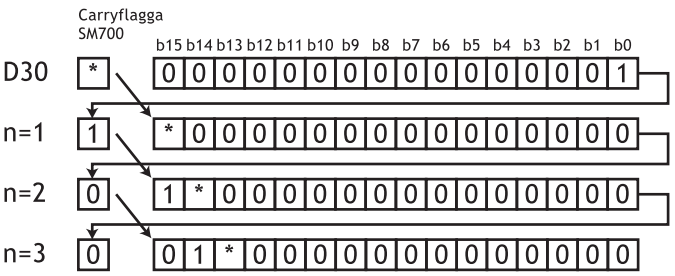
Program som efter aktivering av XA roterar innehållet i D30 tre bitar till höger.




### RCRP

Program som efter aktivering av XC roterar innehållet i D30 med carryflagga tre bitar åt höger.



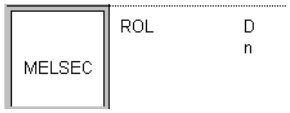
## 8.17.2 Rotation åt vänster, ROL, RCL

| ROL, ROLP, RCL, RCLP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | • | • | • | 3 |

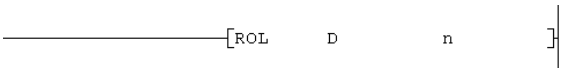
**Tillåtna operander**

| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| D | • | • | • | • | • | • | • | - | - | - |
| n | • | • | • | • | • | • | • | • | - | |

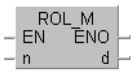
**MELSEC Instruktionslista**




**Ladder**

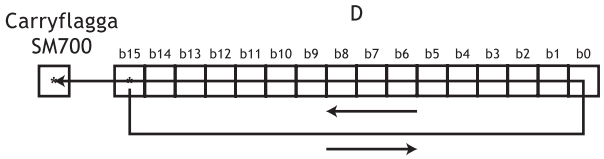



**Funktionsblock**



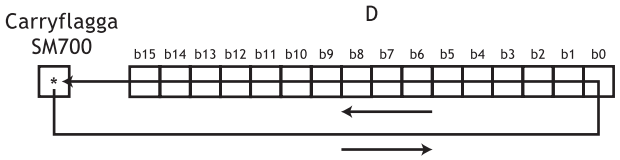

## ROL

Roterar databitarna i D, n bitar till vänster utan carryflagga. Carryflaggan har samma status som biten som sist flyttas från b15 till b0.




## RCL

Roterar databitarna i D, n bitar till vänster med carryflagga. Carryflaggann behåller sin status på biten som roterats n bitar. Status på carryflaggan före rotationen flyttas till vänster n bitar med början från b0.



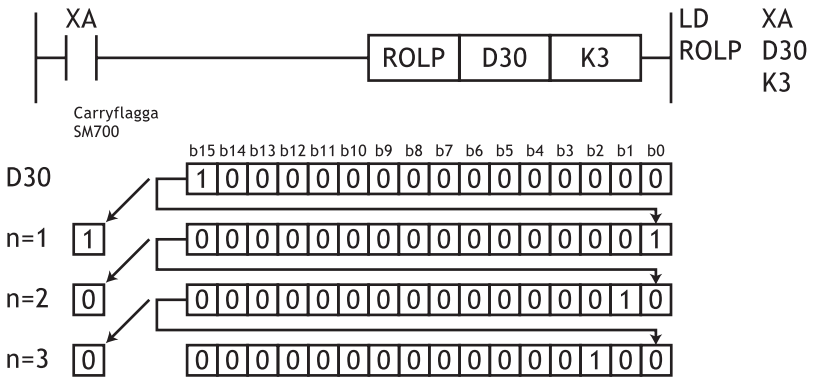

Rotation utförs så länge ingångsvillkoret är TILL om inget förreglingsprogram eller pulstillägg (P) används.

**Observera!**
I GX IEC Developer skall D och n definieras som en enkelordsvariabel (t ex INT)

## Programexempel

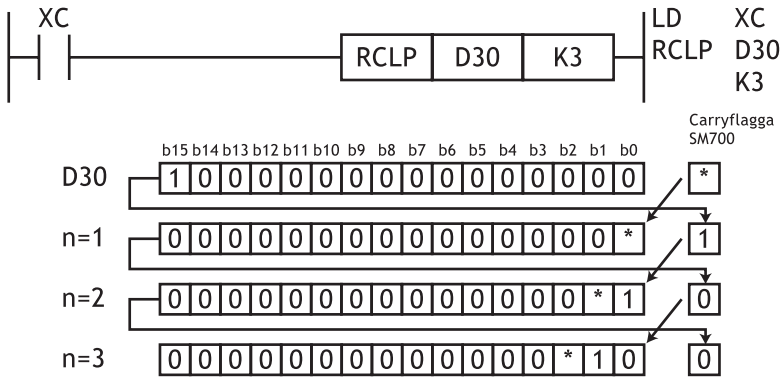
### ROL

Program som efter aktivering av XA roterar innehållet i D30 tre bitar åt vänster.




### RCL

Program som efter aktivering av XC roterar innehållet i D30 med carryflagga tre bitar åt vänster.



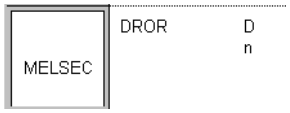
## 8.17.3 Rotation till höger, DROR, DRCR

| DROR, DRORP, DRCR, DRCRP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 3 |

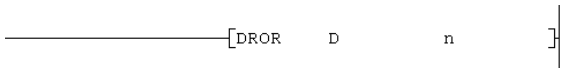
**Tillåtna operander**

| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| D | • | • | • | • | • | • | • | - | - | - |
| n | • | • | • | • | • | • | • | • | - | - |

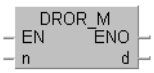
**MELSEC Instruktionslista**




**Ladder**

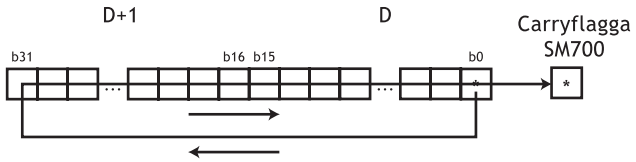



**Funktionsblock**



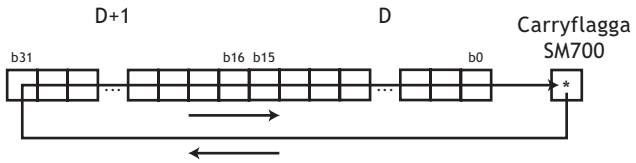

## DROR

Roterar databitarna i D och D+1, n bitar till höger utan carryflagga. Carryflaggan har samma status som biten som sist flyttas från b0 till b31.




## DRCR

Roterar databitarna i D och D+1, n bitar till höger med carryflagga. Carryflaggan behåller status på biten som roteras n bitar.



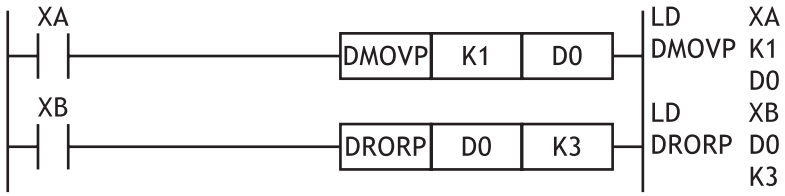

Rotation utförs så länge ingångsvillkoret är TILL om inget förreglingsprogram eller pulstillägg (P) används.

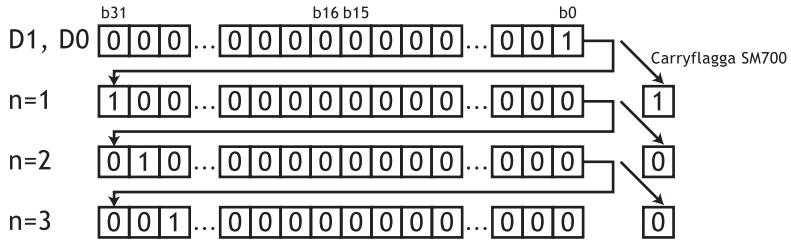
**Observera!**
I GX IEC Developer skall D definieras som en dubbelordsvariabel (t ex DINT) och n som en enkelordsvariabel (t ex INT)

## Programexempel

### DROR

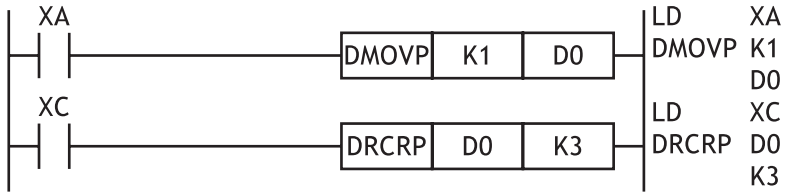
Program som efter aktivering av XB roterar innehållet i D0 och D1 tre bitar till höger.

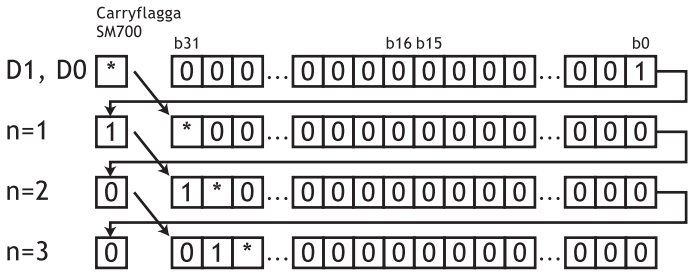



### DRCR

Program som efter aktivering av XC roterar innehållet i D0 och D1 med carryflagga tre bitar till höger.



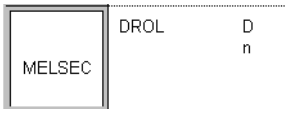
## 8.17.4 Rotation till vänster, DROL, DRCL

| DROL, DROLP, DRCL, DRCLP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | • | • | • | 3 |

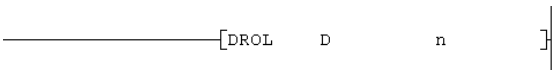
**Tillåtna operander**

| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| D | • | • | • | • | • | • | • | - | - | - |
| n | • | • | • | • | • | • | • | • | - | |

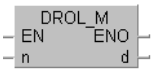
**MELSEC Instruktionslista**




**Ladder**

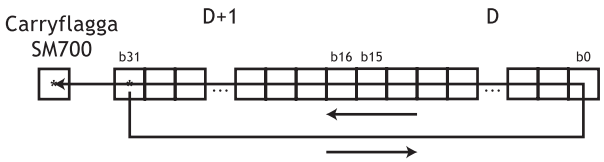



**Funktionsblock**



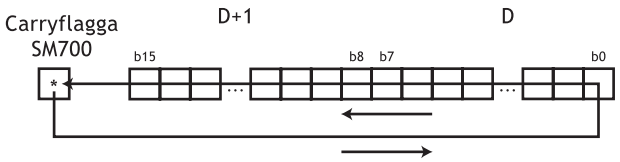

## DROL

Roterar databitarna i D och D+1, n bitar till vänster utan carryflagga. Carryflaggan har samma status som biten som sist flyttas från b31 till b0.




## DRCL

Roterar databitarna i D och D+1, n bitar till vänster med carryflagga. Status på carryflaggan före rotationen flyttas till vänster n bitar med början från b0.



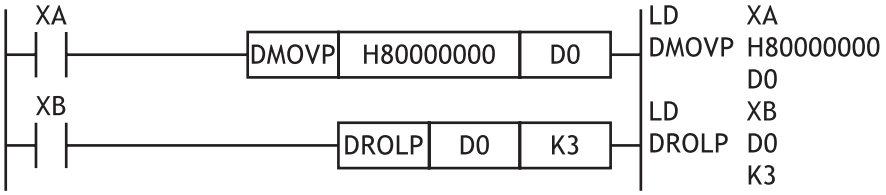

Rotation utförs så länge ingångsvillkoret är TILL om inget förreglingsprogram eller pulstillägg (P) används.

**Observera!**
I GX IEC Developer skall D definieras som en dubbelordsvariabel (t ex DINT) och n som en enkelordsvariabel (t ex INT)

## Programexempel

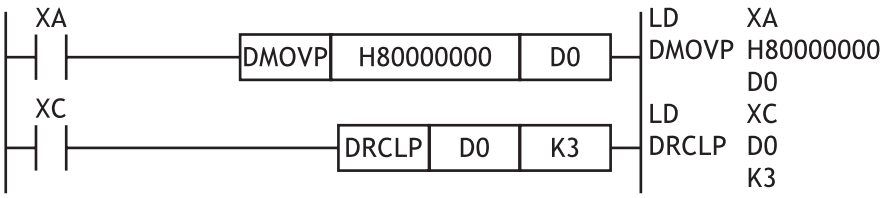
### DROL

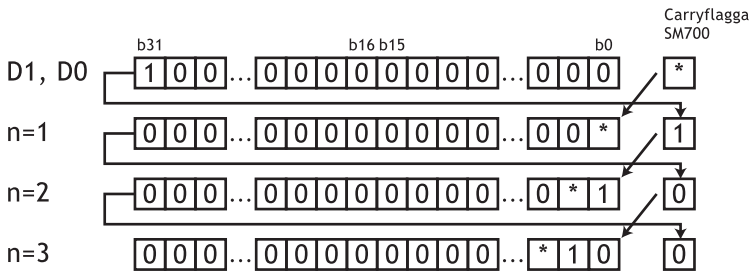
Program som efter aktivering av XB roterar innehållet i D0 och D1 tre bitar åt vänster.



### DRCL

Program som efter aktivering av XC roterar innehållet i D0 med carryflagga tre bitar åt vänster.



# 8.18 Dataskiftinstruktioner

## 8.18.1 Skift av enkelord, SFR, SFL

| SFR, SFRP, SFL, SFLP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 3 |

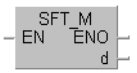
**Tillåtna operander**

| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| D | • | • | • | • | • | • | • | - | - | - |
| n | • | • | • | • | • | • | • | • | - | |

**MELSEC Instruktionslista**

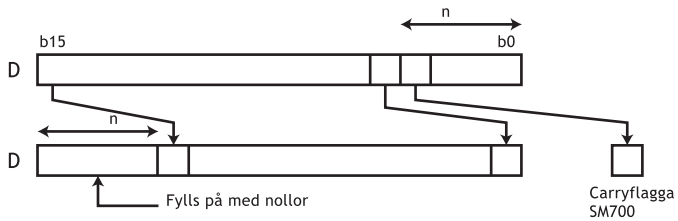
**Ladder**

**Funktionsblock**




## SFR

SFR-instruktionen skiftar enkelordet i (D) till höger med n bitar. Den sist utskiftade biten vid höger skift b(n-1) överförs till carryflaggan.

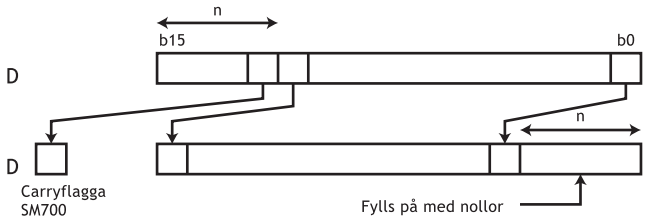



Vid skiftning åt höger fylls nollor på från vänster.

**Observera!**
I GX IEC Developer skall D definieras och n som en enkelordsvariabel (t ex INT)

# SFL

SFL instruktionen skiftar enkelordet i (D) till vänster med n bitar.




Vid skiftning åt vänster, fylls nollor på från höger.

Skifte utförs så länge ingångsvillkoret är TILL om inget förreglingsprogram eller pulstillägg (P) används.

---

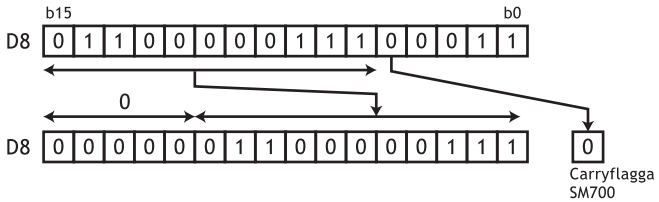
**Observera!**
I GX IEC Developer skall D definieras och n som en enkelordsvariabel (t ex INT)

---

## Programexempel

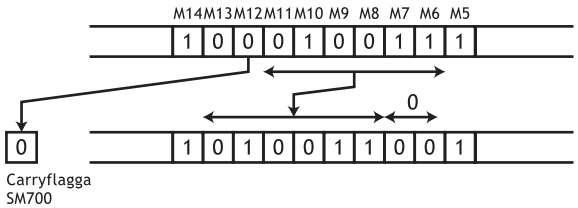
### SFR

Program som efter aktivering av XC flyttar innehållet i D8, 5 bitar åt höger. Status i bit b4 kommer att lagras i SM700.




### SFL

Program som efter aktivering av X8 skiftar innehållet M6 till M13, 2 bitar åt vänster. Status i bit M12 kommer att lagras i SM700.



## 8.18.2 Skift av n bitar, ett steg, BSFR, BSFL

| **BSFR, BSFRP, BSFL, BSFLP** | **Q4AR** | **QnA(S)** | **Q** | **Programsteg** |
|---|---|---|---|---|
| | ● | ● | ● | 3 |

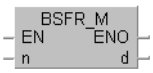
**Tillåtna operander**

| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| **D** | • | - | - | - | - | - | - | - | - | **SM0** |
| **n** | - | • | • | • | • | • | • | • | - | |

**MELSEC Instruktionslista**

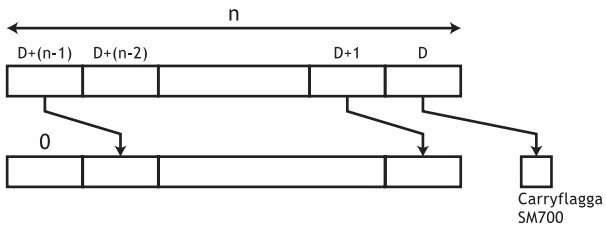
**Ladder**

**Funktionsblock**



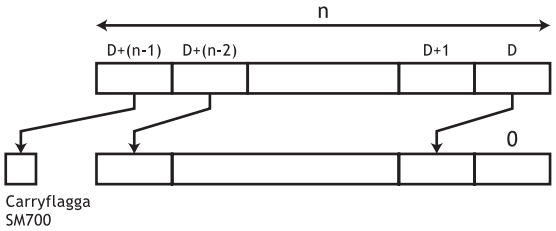

## BSFR

Datamängden specificerad av angiven startadress (D) och längden "n" bitar skiftas ett steg åt höger. Den utskiftade biten (D) överförs till carryflaggan SM700. Vid skiftning höger fylls nolla på från vänster.



# BSFL

Datamängden specificerad av angiven startadress (D) och längden "n" bitar skiftas ett steg åt vänster. Den utskiftade biten D+(n-1) överförs till carryflaggan SM700. Vid skiftning åt vänster fylls nolla på från höger.



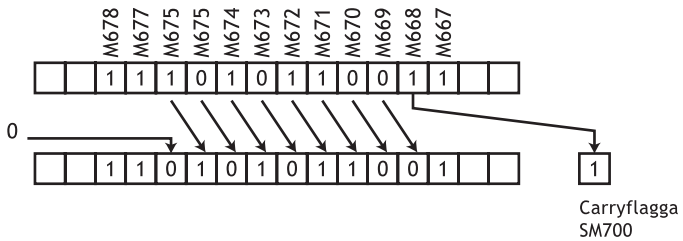

Skifte utförs så länge ingångsvillkoret är TILL om inget förreglingsprogram eller puls- tillägg (P) används.

**Observera!**
I GX IEC Developer skall D definieras och som en bitvariabel (t ex BOOL) och n som en enkelordsvariabel (t ex INT)

## Programexempel

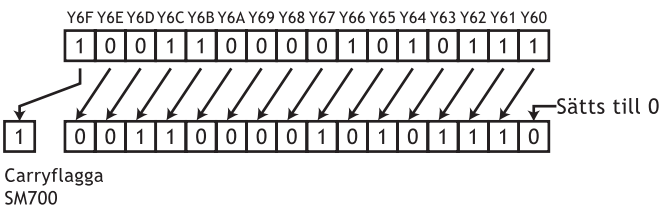
### BSFR

Program som efter aktivering av X8F flyttar minnescellerna M668 - M676 en bit åt höger. Status i första cellen M668 lagras i SM700.




### BSFL

Program som efter aktivering av X4 flyttar utgångarna Y60-Y6F en bit åt vänster. Status i sista cellen Y6F lagras i SM700.



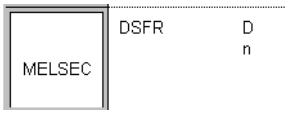
## 8.18.3 Skift av enkelord, ett steg, DSFR, DSFL

| DSFR, DSFRP, DSFL, DSFLP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | • | • | • | 3 |

**Tillåtna operander**

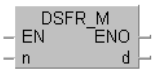
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| D | - | • | • | - | - | - | - | - | - | SM0 |
| n | • | • | • | • | • | • | • | • | - | |

**MELSEC Instruktionslista**



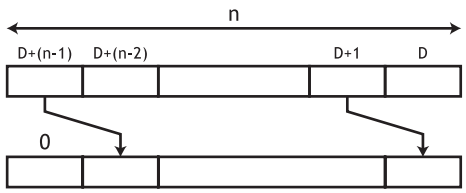

**Ladder**

**Funktionsblock**



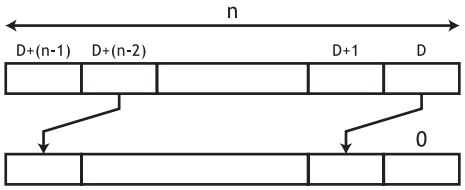

## DSFR

DSFR instruktionen skiftar ordoperandens innehåll från angiven startadress (D) och längden "n" ord skiftas ett ord åt höger. Det högsta ordet ändras till 0. För timers T och räknare C kan endast ärvärden skiftas.



# DSFL

DSFL instruktionen skiftar ordoperandens innehåll från angiven startadress (D) och längden "n" ord ett steg åt vänster. Det lägsta ordet ändras till 0. För timers T och räknare C kan endast ärvärden skiftas.



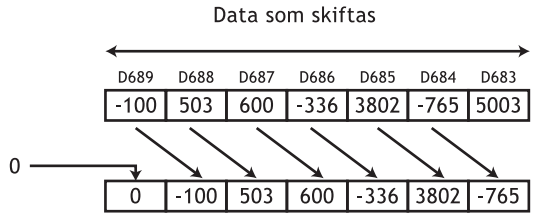

Skifte sker så länge ingångsvillkoret är TILL om inget förreglingsprogram eller puls-tillägg (P) används.

**Observera!**
I GX IEC Developer skall D och n definieras som en enkelordsvariabel (t ex INT)

## Programexempel

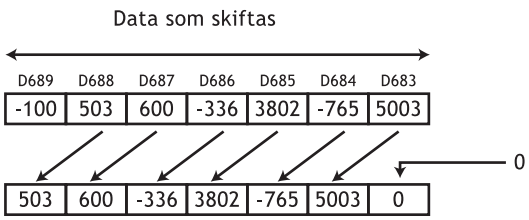
### DSFR

Program som efter aktivering av XB skiftar D683 till D689 ett ord åt höger. Status i D689 sätts till noll.




### DSFL

Program som efter aktivering av XB skiftar D683 till D689 ett ord åt vänster. Status i D683 sätts till noll.



# 8.19 Bitinstruktioner

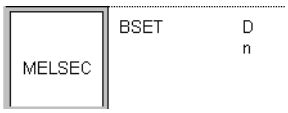
## 8.19.1 Påverkan av bit i enkelord, BSET, BRST

| BSET, BSETP, BRST, BRSTP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | • | • | • | 3 |

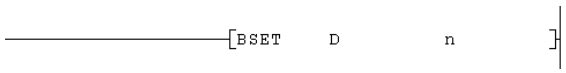
**Tillåtna operander**

| | Interna kanaler | | R, ZR | J** | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| D | • | • | • | - | • | • | • | - | - | - |
| n | • | • | • | • | • | • | • | • | - | - |

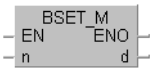
**MELSEC Instruktionslista**




**Ladder**

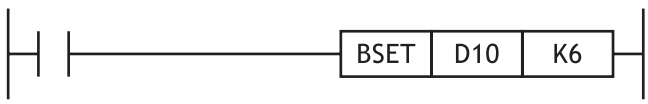


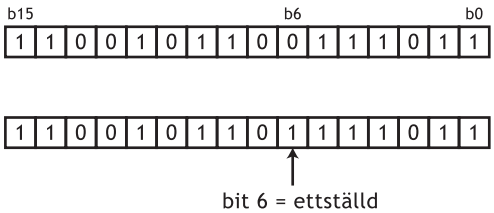

**Funktionsblock**




## BSET

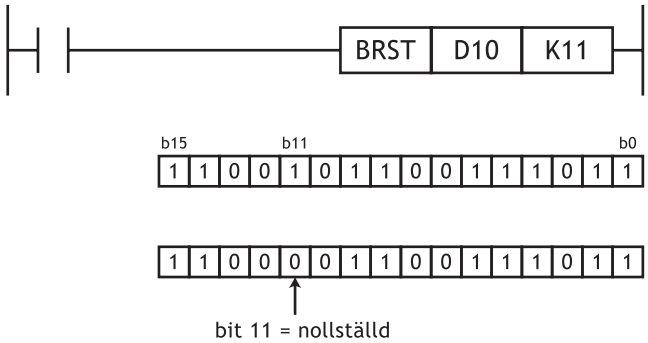
BSET-instruktionen ettställer bit n i ett enkelord definierad i (D), n kan anta ett värde mellan 0 och 15. Sätts värdet högre än 15 utförs instruktionen på värdet på de lägsta 4 bitarna i talet, (talet 17 sätter bit 1).



# BRST

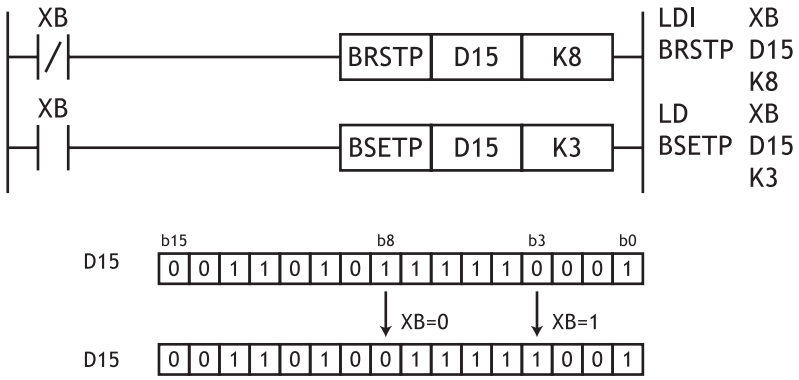
BRST-instruktionen nollställer bit n i ett enkelord definierad i (D), n kan anta ett värde mellan 0 och 15. Sätts värdet högre än 15 utförs instruktionen på de lägsta 4 bitarna i talet (talet 19 sätter bit 3).




## Programexempel

### BSET, BRST

Programmet ettställer tredje biten i D15 när XB ettställs. Den åttonde biten återställs när XB nollställs.




**Observera!**
I GX IEC Developer skall D och n definieras som en enkelordsvariabel (t ex INT)

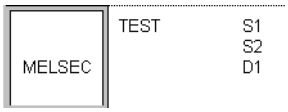
## 8.19.2 Kontroll av bitstatus i ord, TEST

| TEST, TESTP, DTEST, DTESTP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | • | • | • | 4 |

**Tillåtna operander**

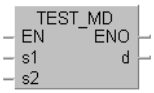
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S1 | • | • | • | • | • | • | • | - | - | - |
| S2 | • | • | • | • | • | • | • | • | - | - |
| D | • | - | - | • | - | - | - | - | - | - |

**MELSEC Instruktionslista**



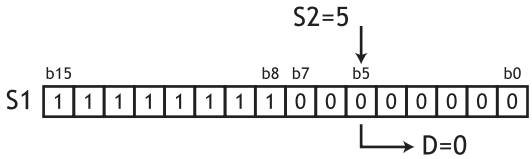

**Ladder**

**Funktionsblock**




# TEST

Instruktionen kontrollerar status hos en bit (S2) i en operand S1. Resultatet lagras resultatet i D. Värdet i S2 kan vara mellan 0 och 15.



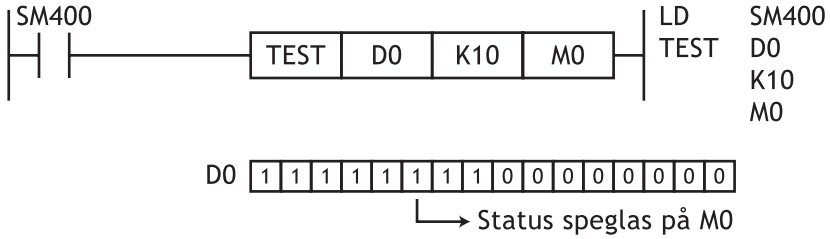

**Observera!**
I GX IEC Developer skall S1 och S2 definieras som en enkelordsvariabel (t ex INT) och D som en bitvariabel (t ex BOOL)

# DTEST

Instruktionen kontrollerar status hos en bit S2 i en ordoperand S1 och S1+1. Resultatet lagras i D. Värdet i S2 kan vara mellan 0 och 31.

**Observera!**
I GX IEC Developer skall S1 och S2 definieras som en dubbelordsvariabel (t ex DINT) och D som en bitvariabel (t ex BOOL)

## Programexempel




I exemplet speglas status på bit 10 i D0 ut på biten M0 när SM400 är ettställd.

# 8.20 Datainstruktioner

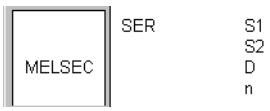
## 8.20.1 Sökning av data, SER

| SER, SERP, DSER, DSERP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | • | • | • | 5 |

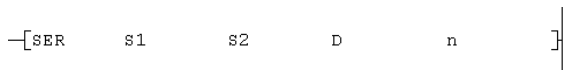
**Tillåtna operander**

| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | U | |
| S1 | • | • | • | • | • | • | • | • | - | SM0 |
| S2 | - | • | • | - | - | - | - | - | - | |
| D | - | • | • | - | • | • | • | - | - | |
| n | • | • | • | • | • | • | • | • | - | |

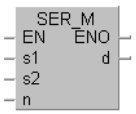
**MELSEC Instruktionslista**




**Ladder**




**Funktionsblock**




## SER

SER-instruktionen ger möjlighet att söka data inom ett definierat område. Detta sker genom att i (S2) ange en startadress. I (S1) anger man den data som skall användas som ett sökbegrepp och med "n" anges längden av sökområdet. Efter genomförd sökning kommer den första träffadressen mellanlagras i D. D+1 innehåller antalet träffar. Om "n" är noll eller negativt kommer ingen sökning att ske. Skulle "n" vara större än antalet kvarvarande adresser för den operandtyp som har startadress (S2), kommer felflagga SM0 att sättas.

**Observera!**
I GX IEC Developer skall S1, S2 och n definieras som en enkelordsvariabel (t ex INT) och D som en vektor med två enkelord (t ex INT)

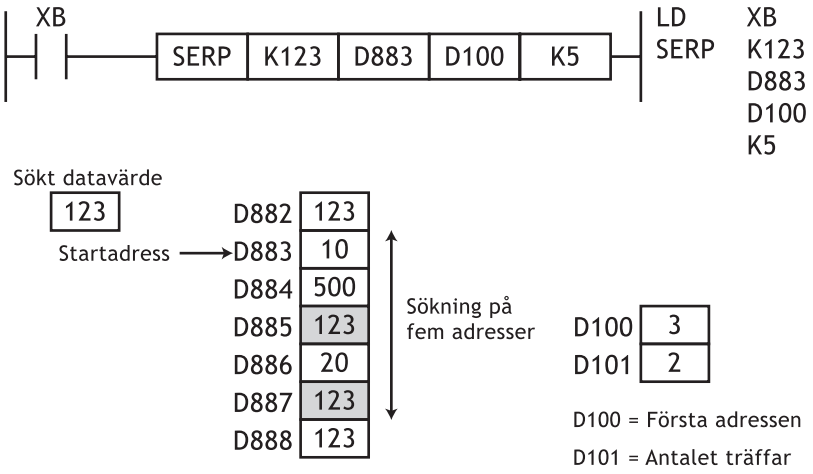
## DSER

På samma sätt utförs DSER men med 32 bitar

**Observera!**
I GX IEC Developer skall S1 och S2 definieras som en dubbelordsvariabel, n skall definieras som en enkelordsvariabel (t ex INT) och D som en vektor med två enkelord (t ex INT)

## Programexempel

Program som efter aktivering av XB söker data i D883 till D887 som stämmer överens med värdet 123.



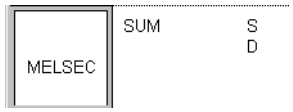
## 8.20.2 Summering av ettställda bitar, SUM

| SUM, SUMP, DSUM, DSUMP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | • | • | • | 3 |

**Tillåtna operander**

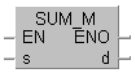
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | • | • | • | • | • | - | - |
| D | • | • | • | • | • | • | • | - | - | |

**MELSEC Instruktionslista**



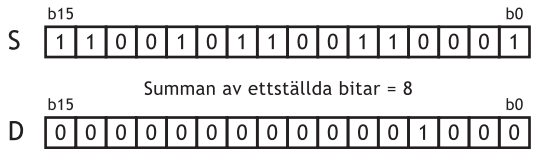

**Ladder**

**Funktionsblock**




## SUM

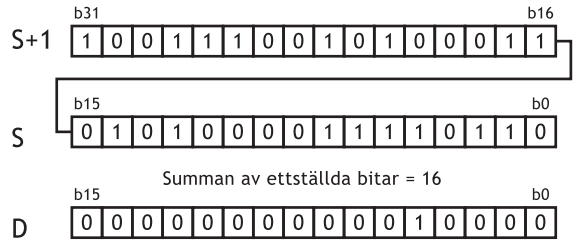
Med hjälp av SUM-instruktionen kan man summera antalet ettställda bitar i ett enkelord. Adressen som skall kontrolleras anges i (S). Resultatet lagras som ett binärt tal i operanden (D).




**Observera!**
I GX IEC Developer skall S och D definieras som en enkelordsvariabel (t ex INT)

# DSUM

Med hjälp av DSUM-instruktionen kan man summera antalet ettställda bitar i ett dubbelord. Adressen som skall kontrolleras anges i (S). Resultatet lagras som ett binärt tal i operanden (D).

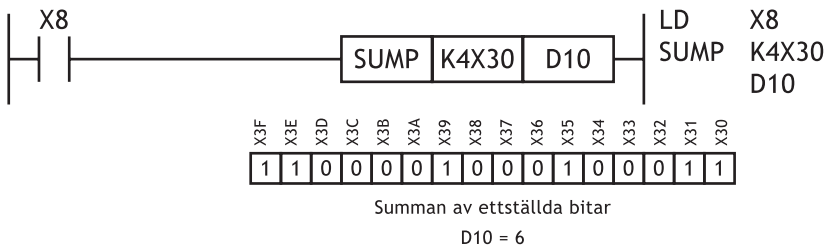



**Observera!**
I GX IEC Developer skall S definieras som en dubbelordsvariabel (t ex DINT) och D definieras som en enkelordsvariabel (t ex INT)

## Programexempel

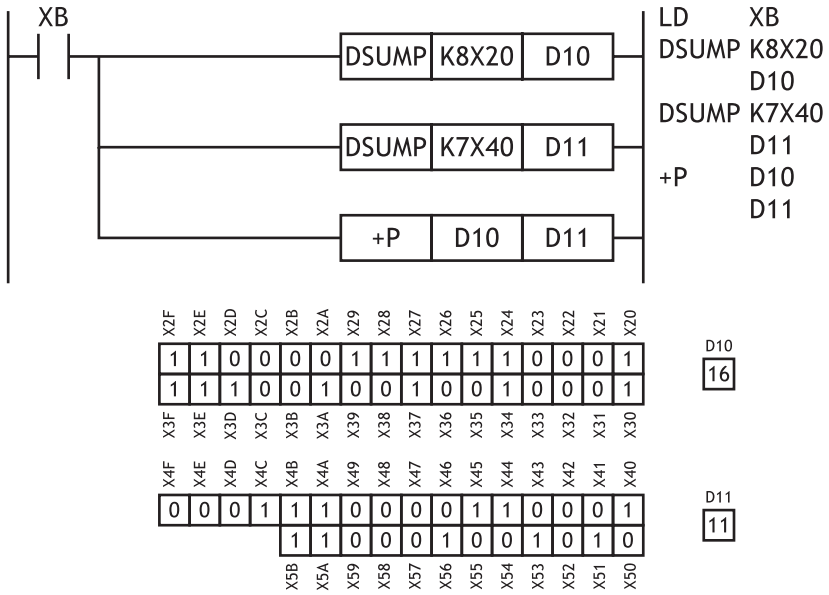
### SUM

Program som efter aktivering av X8 summerar ettställda ingångar och lagrar resultatet i D10.



## DSUM

Program som summerar ettställda ingångar mellan X20 och X3F och lagrar resultatet i D10. Sedan summeras de ettställda ingångarna mellan X40 och X5B och resultatet lagras i D11. Sedan adderas resultaten och summan lagras D11.




| X2F | X2E | X2D | X2C | X2B | X2A | X29 | X28 | X27 | X26 | X25 | X24 | X23 | X22 | X21 | X20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| X3F | X3E | X3D | X3C | X3B | X3A | X39 | X38 | X37 | X36 | X35 | X34 | X33 | X32 | X31 | X30 |

D10 = 16

| X4F | X4E | X4D | X4C | X4B | X4A | X49 | X48 | X47 | X46 | X45 | X44 | X43 | X42 | X41 | X40 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| | | | | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| | | | | X5B | X5A | X59 | X58 | X57 | X56 | X55 | X54 | X53 | X52 | X51 | X50 |

D11 = 11

Antalet ettställda ingångar mellan X20 och X5B: D11 = 27

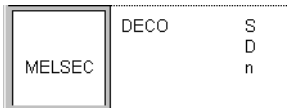
## 8.20.3 Avkodning, DECO

| DECO, DECOP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 4 |

**Tillåtna operander**

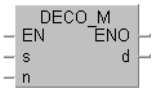
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | • | • | • | • | • | - | SM0 |
| D | • | • | • | - | - | - | - | - | - | |
| n | • | • | • | • | • | • | • | • | - | |

**MELSEC Instruktionslista**




**Ladder**

**Funktionsblock**




# DECO

Med hjälp av DECO-instruktionen kan 8 bitar data avkodas till 256 bitar. Startadressen för operanden som skall avkodas anges i (S) och antalet adresser anges i "n". Resultatet av binäravkodningen lagras i (D). Talet "n" skall vara mellan 1 och 8.

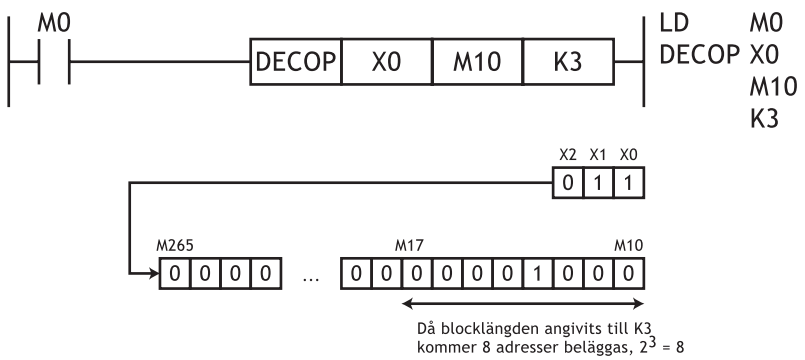
Fel som kan aktivera felflaggorna.

– Värdet n ligger inte mellan 0 och 8 (värdet 0 innebär ingen avkodning).

**Observera!**
I GX IEC Developer skall S och D definieras som en bit- eller enkelordsvariabel (t ex BOOL eller INT)) och n som en enkelordsvariabel (t ex INT)

### Programexempel

Data från X0 till X2 dekodas och lagras i M10 till M17. Då X0 och X1 är lika med ett kommer M13 (tredje positionen från M10) att ettställas.

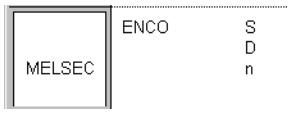


## 8.20.4 Kodning, ENCO

| ENCO, ENCOP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 4 |

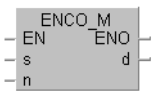
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J** | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | - | - | - | - | • | - | SM0 |
| D | • | • | • | • | • | • | • | - | - | |
| n | • | • | • | • | • | • | • | • | - | |

**MELSEC Instruktionslista**




**Ladder**

**Funktionsblock**




# ENCO

Med hjälp av ENCO-instruktionen kan 256 bitars data kodas i 8 bitar. Startadressen för operanden som skall kodas anges i (S) och antalet adresser anges i n. Resultatet av binärkodningen* lagras i (D). Talet n skall vara mellan 1 och 8. En bitoperand hanteras som 1 bit och en enkelordoperand hanteras som 16 bitar. Om mer än en bit är satt kodas den med lägst adress.

*) Fel som aktivera felflaggan:

– Värdet n ligger inte mellan 0 och 8 (0 innebär ingen förändring)

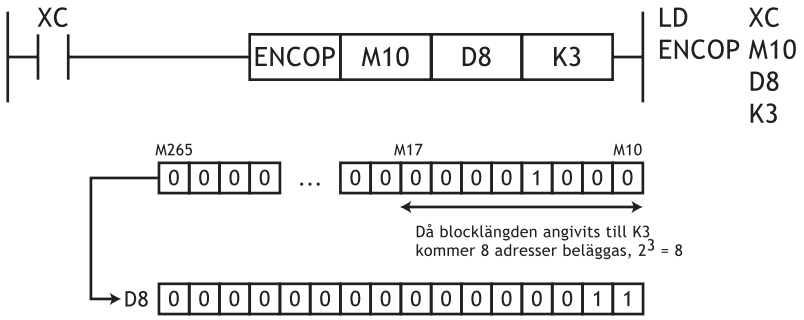
– Alla bitar i (S) är noll

**Observera!**
I GX IEC Developer skall S och D definieras som en bit- eller enkelordsvariabel (t ex BOOL eller INT)) och n som en enkelordsvariabel (t ex INT)

## Programexempel

### ENCO

Data i M10 till M17 läses och kodas och lagras som binärvärde i D8.



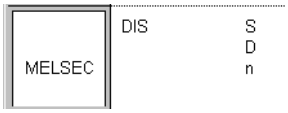
# 8.20.5 Uppackning av data, DIS

| DIS, DISP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 4 |

**Tillåtna operander**

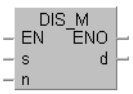
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S1 | • | • | • | • | • | • | • | • | - | SM0 |
| D | - | • | • | - | - | - | - | - | - | |
| n | • | • | • | • | • | • | • | • | - | |

**MELSEC Instruktionslista**



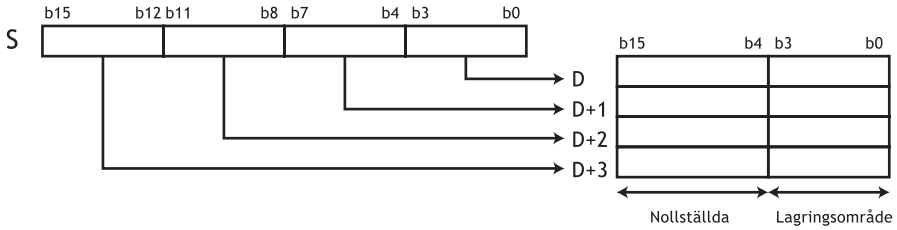

**Ladder**

**Funktionsblock**




## DIS

Uppackning av upp till fyra enkelord till grupper om 4 bitar. Med hjälp av DIS-instruktionen kan grupper om 4 bitar i ett enkelord packas upp och lagras. S innehåller de bitar som ska packas upp. Dessa lagras i D i grupper om 4 bitar. I n anger man antalet grupper som skall behandlas (1-4). Om n=0 kommer ingen förändring att ske. De övre 12 bitarna i D sätts till 0.

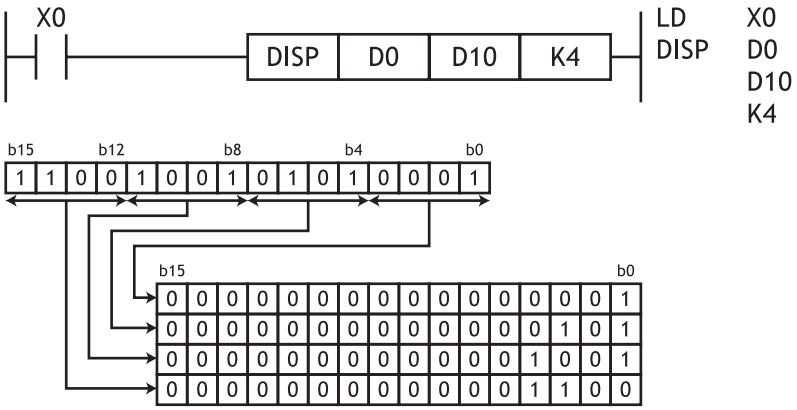



**Observera!**
I GX IEC Developer skall S, D och n definieras som en enkelordsvariabel (t ex INT)

## Programexempel

### DIS

Programmet delar upp enkelord i grupper om 4 bitar och lagrar dessa i D10 till D13.



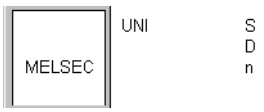
# 8.20.6 Nerpackning av data, UNI

| UNI, UNIP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 4 |

**Tillåtna operander**

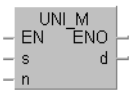
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | - | - | - | - | - | - | SM0 |
| D | - | • | • | • | • | • | • | - | - | |
| n | • | • | • | • | • | • | • | • | - | |

**MELSEC Instruktionslista**



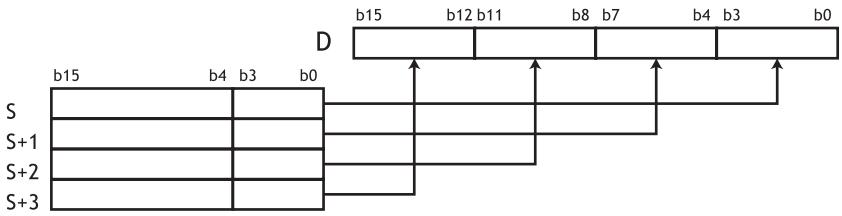

**Ladder**

**Funktionsblock**




## UNI

Nerpackning av grupper om 4 bitar till upp till 4 enkelords. UNI-instruktionen sätter samman och lagrar de lägsta 4 bitarna i ett enkelord. S innehåller de bitar som ska packas ner. De lagras i D nerpackade i grupper om 4 bitar. I n anger man antalet grupper som skall sättas samman (1-4). De övre bitarna i D som ej berörs av nedpackningen sätts till 0. Om n=0 kommer ingen förändring att ske. Om n inte ligger i intervallet 0-4 kommer programmet avbrytas och felflaggan sättas.

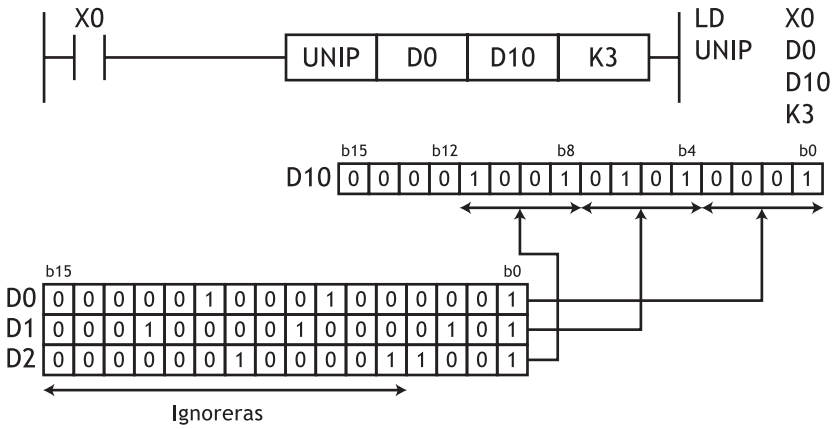



**Observera!**
I GX IEC Developer skall S, D och n definieras som en enkelordsvariabel (t ex INT)

## Programexempel

### UNI

De lägsta 4 bitarna i D0 till D2 överförs till D10 i grupper om 4 bitar.



# 8.21 Strukturerade programinstruktioner

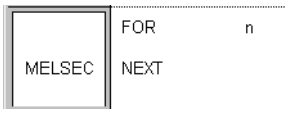
## 8.21.1 Upprepande instruktioner, FOR, NEXT

| FOR, NEXT | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | FOR: 2<br>NEXT: 1 |

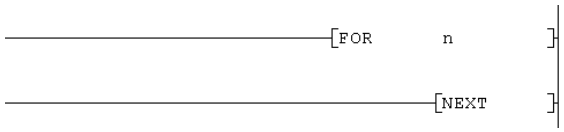
**Tillåtna operander**

| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| n | • | • | • | • | • | • | • | • | - | SM0 |

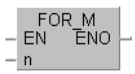
**MELSEC Instruktionslista**




**Ladder**




**Funktionsblock**

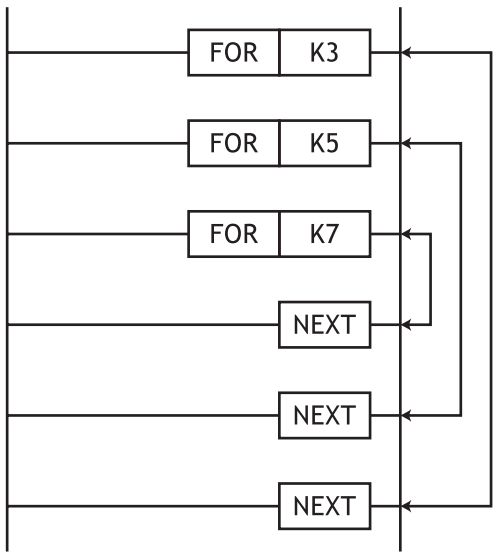



## FOR

Med hjälp av-instruktionerna kan en programdel som ligger mellan FOR- och NEXT-instruktionerna upprepas n gånger där n kan vara ett tal mellan 1 och 32 767. Om programdelen mellan FOR och NEXT inte skall utföras måste den hoppas över med CJ eller SCJ. Maximalt 5 nivåer av FOR och NEXT loopar är tillåtna.

### Programexempel

Exempel på 3 nivåer FOR-NEXT loopar.



## Felkällor

Fel i programbearbetningen inträffar när:

– FEND eller END finns mellan FOR och NEXT.

– NEXT kommer före FOR.

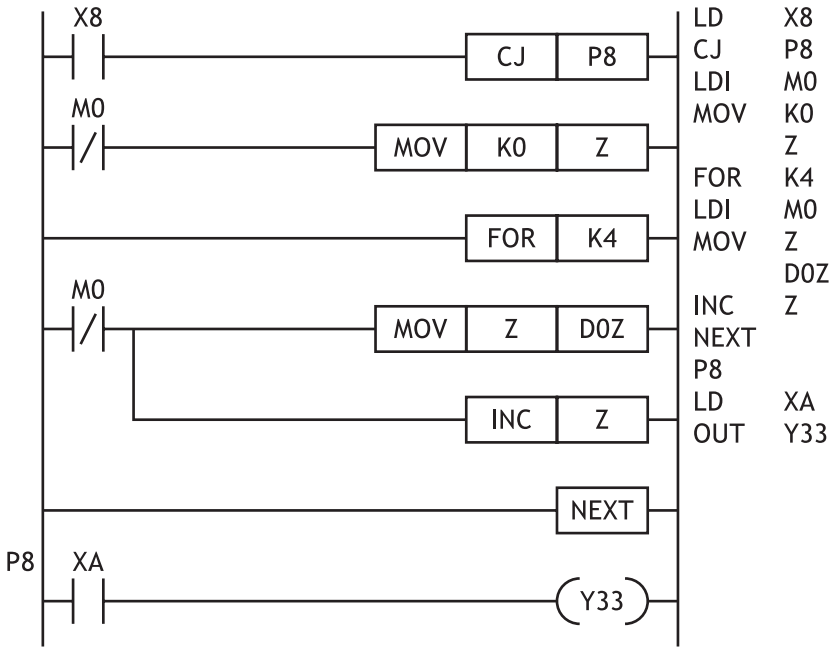
– Antalet FOR och NEXT inte är lika.

**Observera!**
I GX IEC Developer skall n definieras som en enkelordsvariabel (t ex INT)

## Programexempel

### FOR, NEXT

Program som utför FOR-/NEXT-instruktioner när X8 är nollställd och hoppar över dessa när X8 är ettställd.



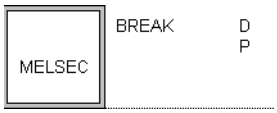
## 8.21.2 Avbrytning av upprepande instruktion, BREAK

| BREAK, BREAKP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 3 |

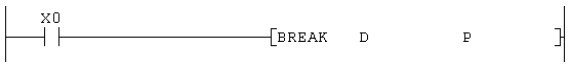
**Tillåtna operander**

| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | P | |
| D | • | • | • | • | • | • | • | - | - | SM0 |
| P | - | - | - | - | - | - | - | - | • | |

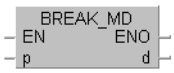
**MELSEC Instruktionslista**




**Ladder**




**Funktionsblock**




# BREAK

Instruktionen avbryter en FOR-/NEXT-loop och hoppar till en pekaradress som anivits i P. Antalet återstående FOR-/NEXT-loopar anges i D.

BREAK kan endast användas en gång under en FOR-/NEXT-loop.

BREAK kan endast användas i en nivå, för flera nivåer måste flera BREAK-instruktioner exekveras.

I följande fall sätts felflaggan:

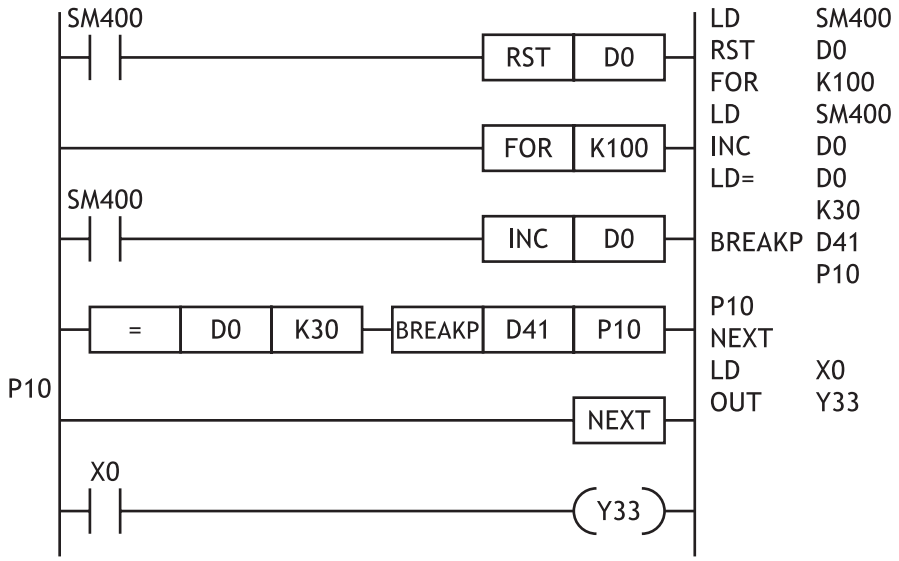
– BREAK utfördes utanför en FOR-/NEXT-loop.

– Det finns inget subprogram lagrat vid pekaradressen som angivits i P.

**Observera!**
I GX IEC Developer skall n definieras som en enkelordsvariabel (t ex INT) och P som pekare eller Label

## Programexempel

Föjande program avslutar exekvereringen i den trettionde (30) FOR/NEXT-loopen och hoppar till programmet vid pekare P10. Antalet återstående loopar (70) lagras i D41.



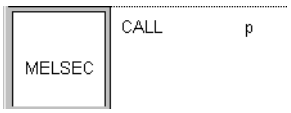
## 8.21.3 Anrop av subrutinprogram, CALL

| CALL, CALLP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 2 |

**Tillåtna operander**

| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | P | |
| P | - | • | • | - | - | - | - | - | • | SM0 |

**MELSEC Instruktionslista**

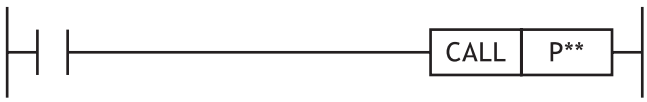



**Ladder**

**Funktionsblock**

## CALL(P)

Instruktionen anropar en subrutin med pekaradress P.
P har ett värde mellan P0 och P4095 (254 för MELSEC QnA(S)). Upp till fem nivåer av CALL(P)-instruktioner kan användas. Detta innebär att en subrutin kan anropas från en annan subrutin. CALLP-instruktionen utförs en gång vid positiv flank.

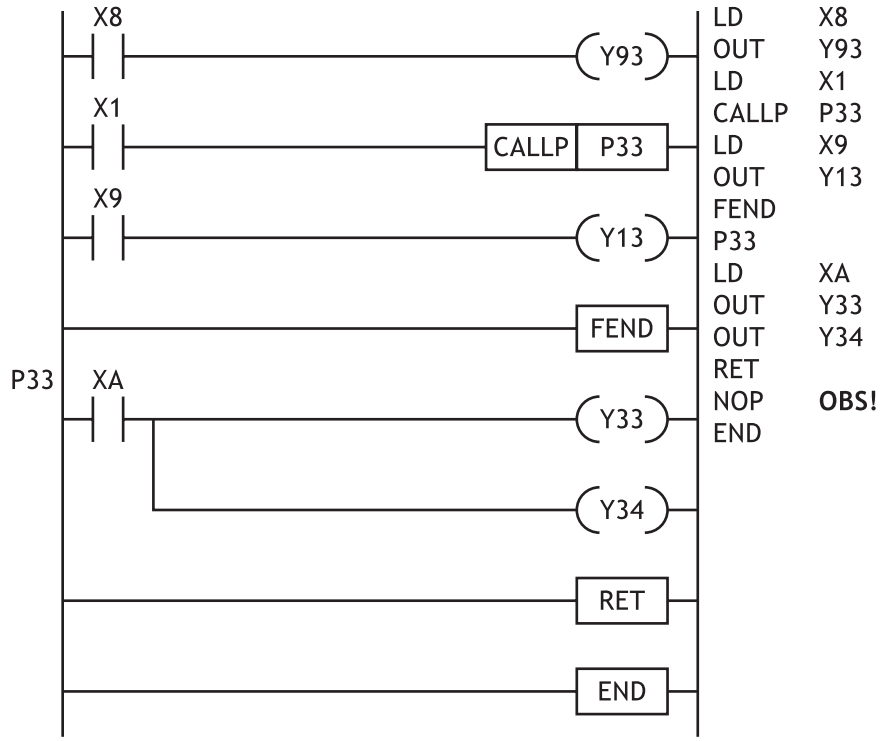



**Observera!**
I GX IEC Developer kan CALL (P)_M inte användas i IEC-editorerna. Använd egendefinierade funktioner och funktionsblock istället.

## Programexempel

### CALL, RET

Program som anropar en subrutin på pekaradress P33 när X1 aktiveras.



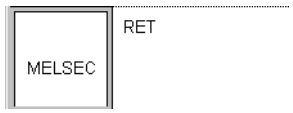

**OBS!**
NOP-instruktionen skall alltid finnas precis före END-instruktionen.
GX IEC Developer lägger själv till NOP före END vid kompilering.

## 8.21.4 Slut på subrutinprogram, RET

| RET | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 1 |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| P | - | - | - | - | - | - | - | - | - | SM0 |

**MELSEC Instruktionslista**




**Ladder**

**Funktionsblock**

# RET

Betecknar slutet på en subrutin.
Alla subrutiner måste avslutas med en RET-instruktion. Återhopp till huvudprogrammet sker till motsvarande CALL-instruktion. Mellan en RET-instruktion i subrutinen och END-instruktionen i huvudprogrammet måste en NOP-instruktion läggas in för att PLC-systemet skall fungera felfritt.

När ett program använder PLS- och PLF-instruktionerna i en subrutin och subrutinen endast utförs en gång kan PLS-/PLF-instruktionernas operander bli ettställda mer än ett programvarv.

Exempel på felkällor

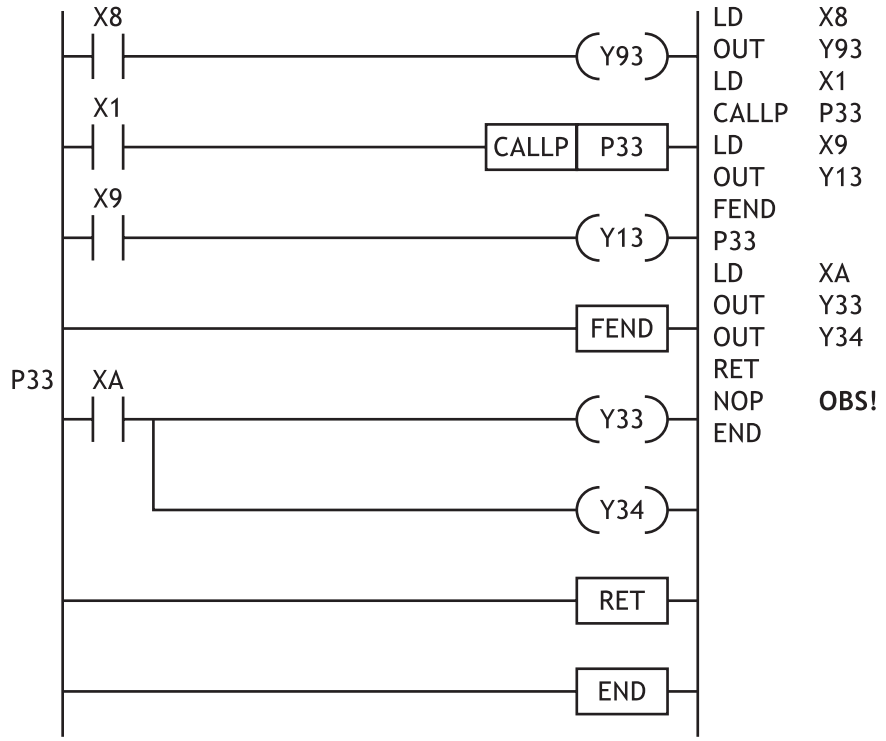
– FEND-instruktion utförs mellan en CALL(P)- och en RET-instruktion.
– RET-instruktion har utförts före en CALL(P)-instruktion.
– RET får endast användas som slut på en subrutin.
– Fler än fem subrutinnivåer har använts.

**Observera!**
I GX IEC Developer kan RET_M inte användas i IEC-editorerna. Använd egendefinierade funktioner och funktionsblock istället.

## Programexempel

### CALL, RET

Program som anropar en subrutin på pekaradress P33 när X1 aktiveras.




**OBS!**
NOP-instruktionen skall alltid finnas precis före END-instruktionen.
GX IEC Developer lägger själv till NOP före END vid kompilering.

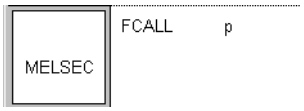
## 8.21.5 Nollställning av utgångar i subrutinprogram, FCALL

| FCALL, FCALLP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 2 |

**Tillåtna operander**

| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| P | - | • | • | - | - | - | - | - | - | SM0 |

**MELSEC Instruktionslista**

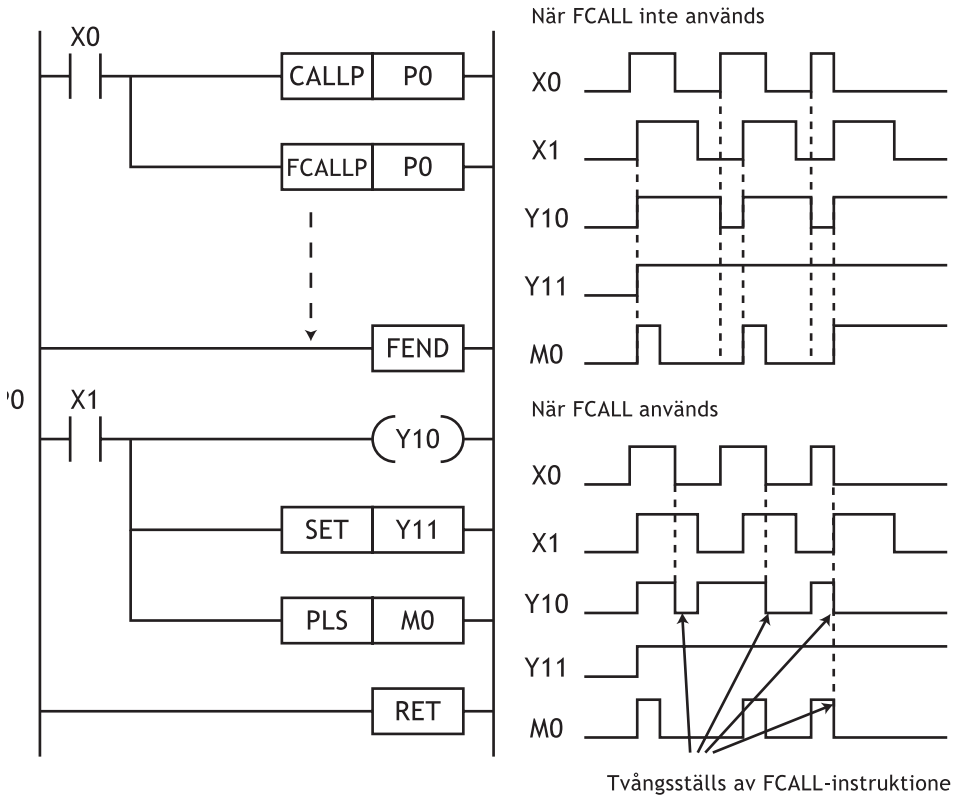



**Ladder**

**Funktionsblock**

# FCALL

Användes tillsammans med CALL-instruktionen. Nollställer utgångar i ett subrutinprogram. Endast de utgångar som är definierade i subrutinprogrammet i Pn påverkas.




Tvångsställs av FCALL-instruktione

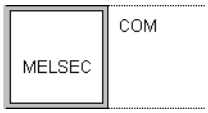
**Observera!**
I GX IEC Developer kan FCALL(P) inte användas i IEC-editorerna. Använd egendefinierade funktioner och funktionsblock istället.

## 8.21.6 Uppdatera nätverkskommunikation, COM

| COM | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 1 |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| - | - | - | - | - | - | - | - | - | - | - |

**MELSEC Instruktionslista**




**Ladder**

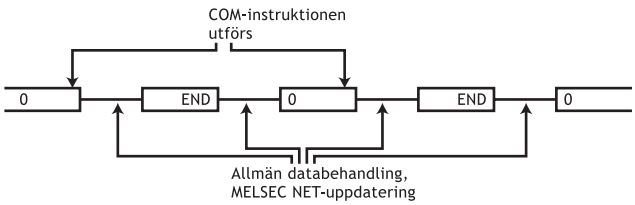
**Funktionsblock**

# COM

COM-instruktionen används för att göra kommunikationen snabbare med en remote I/O-station eller för att få en snabbare uppdatering av MELSEC NET när masterstationens programcykeltid är längre än lokalstationens.

När en COM-instruktion utörs avbryts bearbetningen av PLC programmet tillfälligt och en allmän databehandling utförs.

COM-instruktionen kan användas valfritt antal gånger i PLC programmet, dock kommer cykeltiden att förlängas på grund av den allmänna databehandlingen och MELSEC NET-uppdateringen.




I den allmänna databehandlingen ingår:

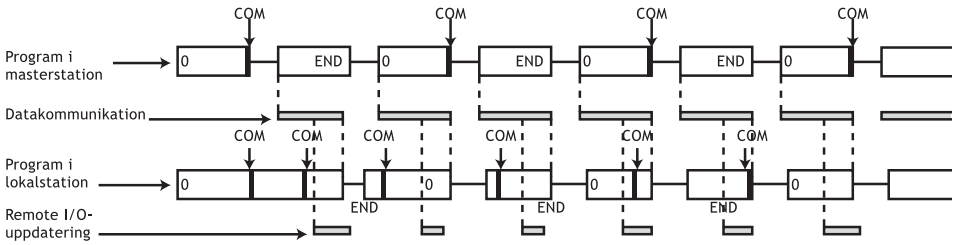
– Kommunikation mellan CPU och yttre enheter

– Övervakning och kontroll av övriga stationer i MELSEC NET-slingan

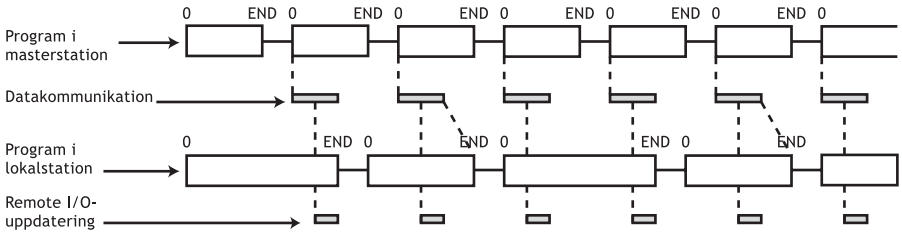
– Läsa buffertminnen i andra funktionsmoduler som är anslutna via en kommunikationsmodul.

## Exempel på villkor för utförande

Med COM-instruktion




Utan COM-instruktion




Genom att använda COM-instruktionen i masterstationen kan datakommunikationen utföras snabbare eftersom antalet kommunikationstillfällen blir fler. Om man dessutom använder COM-instruktionen i lokalstationerna förbättras säkerheten i mottagningen även om cykeltiden i lokalstationen är längre än i masterstationen. COM-instruktionen medför också att uppdatering av MELSEC NET sker varje gång lokalstationen får kommandon från mastern mellan:

– Steg 0 och COM-instruktion

– COM-instruktion och COM-instruktion

– COM-instruktion och END-instruktion

Om cykeltiden för MELSEC NET är längre än masterstationens programcykeltid kan kommunikationen inte snabbas upp med COM-instruktionen.

Remote I/O uppdatering = uppdatering av in/utgångar i remote I/O station.

# 8.22 Datatabellinstruktioner

## 8.22.1 Skriv till FIFO-register, FIFW

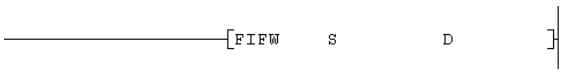
| FIFW, FIFWP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 3 |

**Tillåtna operander**

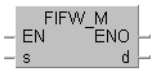
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | • | • | • | • | - | - | SM0 |
| D | - | • | • | - | - | - | - | - | - | |

**MELSEC Instruktionslista**

**Ladder**

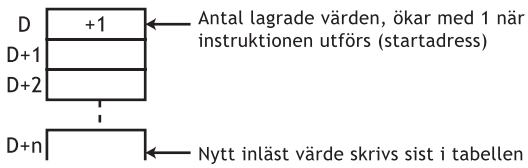



**Funktionsblock**




# FIFW

FIFO-tabellen används för att buffra data och fungerar enligt principen först in-först ut. Hanteringen sker med två instruktioner, läsning till och skrivning från tabellen. Med FIFW-instruktionen kan man skriva in 16-bitarsord i FIFO-tabellen. Värdet hämtar man i (S) och FIFO-tabellens startadress anger man i (D).




**Observera!**
I GX IEC Developer skall S och D definieras som en enkelordsvariabel (t ex INT)

### Programexempel

Se FIFR-instruktionen i *kapitel 8.22.2*.

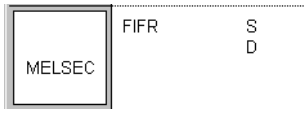
## 8.22.2 Läs från FIFO-register, FIFR

| FIFR, FIFRP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 3 |

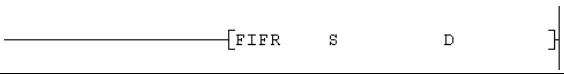
**Tillåtna operander**

| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | • | • | • | • | - | - | SM0 |
| D | - | • | • | - | - | - | - | - | - | |

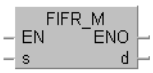
**MELSEC Instruktionslista**




**Ladder**



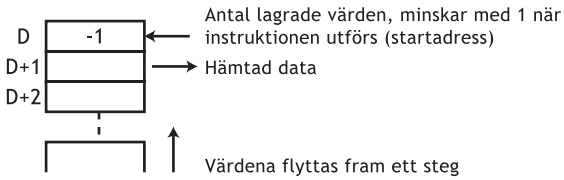

**Funktionsblock**




# FIFR

Med hjälp av FIFR-instruktionen kan värden läsas från FIFO tabellen.

Det hämtade värdet lagras i S och D anger startadressen till tabellen. Det hämtade värdet finns alltid på andra plats efter registret som anger antalet värden. Vid hämtning flyttas efterföljande värden upp ett steg och minskar antalet värden med ett. Frilagda register nollställs.

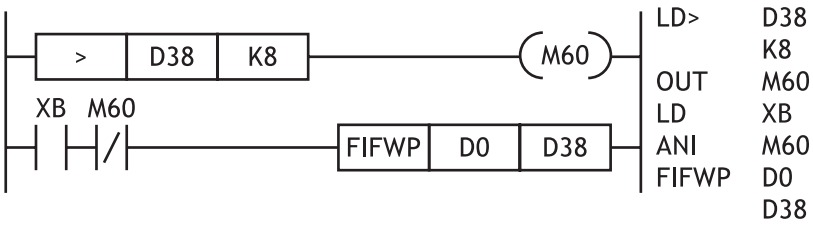


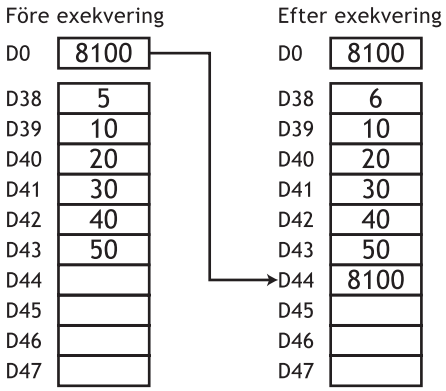

**Observera!**
I GX IEC Developer skall S och D definieras som en enkelordsvariabel (t ex INT)

## Programexempel

### FIFW

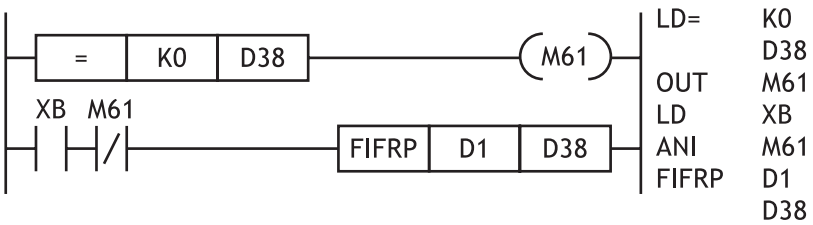
En FIFO-tabell upprättas i dataregistren D38-D47. Startadressen innehåller positionspekaren som visar antalet lagrade värden. Efter ettställning av XB kommer datan i D0 att belägga nästa fria minnesposition i FIFO-listan (ex D44). I den aktuella FIFO-listan kan maximalt 9 adresser beläggas. M60 programmeras här som en förregling för FIFW-instruktionen.
Minnescellen aktiveras om D38 är större eller lika med 9.

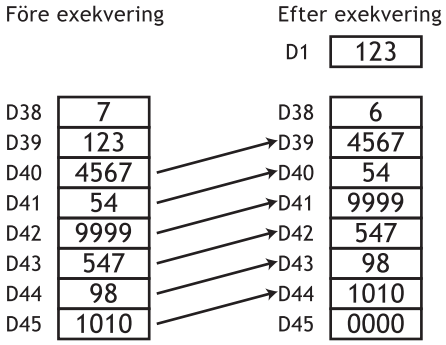



### FIFR

Program som läser data från D38 till D45 i FIFO tabellen och överför dessa till register D1.



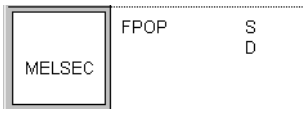



## 8.22.3 Läs data från slutet av FIFO-register, FPOP

| FPOP, FPOPP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 3 |

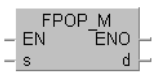
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | • | • | • | • | • | - | SM0 |
| D | - | • | • | - | - | - | - | - | - | |

**MELSEC Instruktionslista**




**Ladder**

**Funktionsblock**

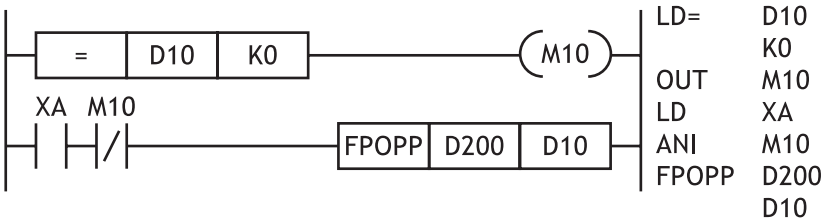



# FPOP

Läser ut data från tabell enligt principen sist in först ut. Läsningen börjar med sista adressen D+n i datatabellen. Datan överförs till lagerområdet definierad i S. Läsadressen sättes till 0. Efter en läsning minskas positionspekaren (D) med 1.
FPOP-instruktionen får ej utföras om positionspekaren (D) är lika med noll, då aktiveras felflaggan.

**Observera!**
I GX IEC Developer skall S och D definieras som en enkelordsvariabel (t ex INT)

## Programexempel




Före exekvering

| | |
|---|---|
| D10 | 7 |
| D11 | 10 |
| D12 | 20 |
| D13 | 30 |
| D14 | 40 |
| D15 | 50 |
| D16 | 60 |
| D17 | 70 |

Efter exekvering

| | |
|---|---|
| D200 | 70 |

| | |
|---|---|
| D10 | 6 |
| D11 | 10 |
| D12 | 20 |
| D13 | 30 |
| D14 | 40 |
| D15 | 50 |
| D16 | 60 |
| D17 | 0 |

Så länge positionspekaren (D10) ej är lika med noll läses det sista värdet i datatabellen och lagras i D200 varje gång XA går till (positiv flank).

# 8.23 Buffertminnesinstruktioner

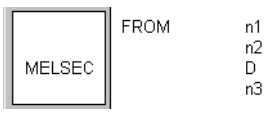
## 8.23.1 Läs från funktionsmodul, FROM

| FROM, FROMP, DFRO, DFROP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | • | • | • | 5 |

**Tillåtna operander**

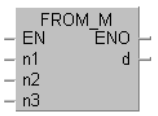
| | Interna kanaler | | R, ZR | J** | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | U | |
| n1 | • | • | • | • | • | • | • | • | • | SM0 |
| n2 | • | • | • | • | • | • | • | • | - | |
| D | • | • | • | - | - | - | - | - | - | |
| n3 | • | • | • | • | • | • | • | • | - | |

**MELSEC Instruktionslista**



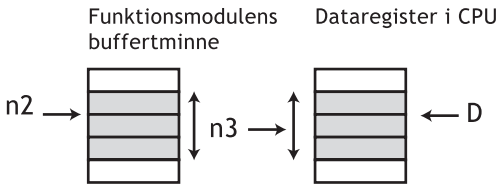

**Ladder**

**Funktionsblock**




## FROM

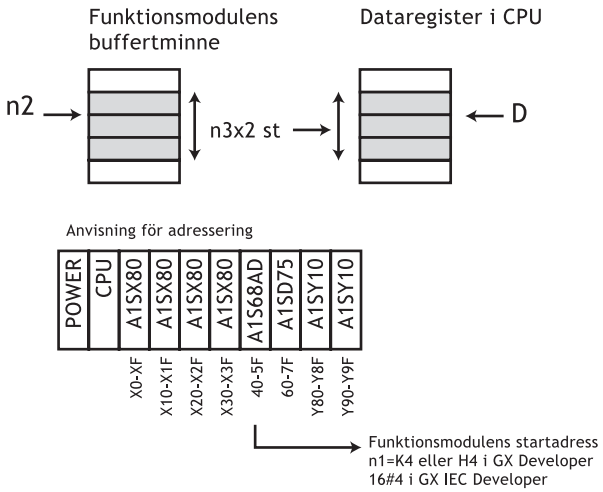
FROM-instruktionen läser enkelord från en funktionsmoduls buffertminne och lagrar det i ett definierat register i PLC-systemet. Startadressen för läsning anges i n2 och antalet buffertminnen som skall läsas anges i n3 och modulens grundadress anges i n1. Lagringsadressen anges i (D).




**Observera!**
I GX IEC Developer skall n1, n2, n3 och D definieras som en enkelordsvariabel (t ex INT)

# DFRO

DFRO-instruktionen läser dubbelord (2 buffertminnen) från en funktionsmoduls buffertminne och lagrar det i ett definierat register i PLC-systemet. Startadressen för läsning anges i n2 och antalet dataord som skall läsas anges i n3 och modulens grundadress anges i n1. Lagringsadressen anges i (D).




**Observera!**
I GX IEC Developer skall n1, n2, n3 definieras som en enkelordsvariabel (t ex INT) och D definieras som en dubbelordsvariabel (t ex DINT)

Fel i programbearbetningen inträffar när:

– På angiven I/O adress (n1) sitter fel typ av modul.

– Funktionsmodulen är ej tillgänglig.

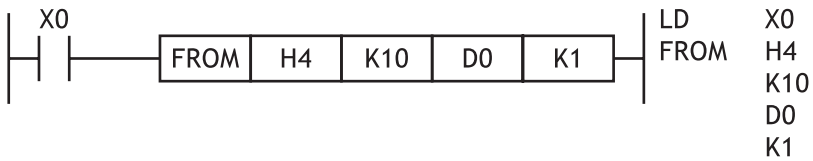
– Antalet definierade dataord (n3) överstiger det i D angivna minnesområdet.

I samtliga fall kommer felflagga att sättas.

## Programexempel

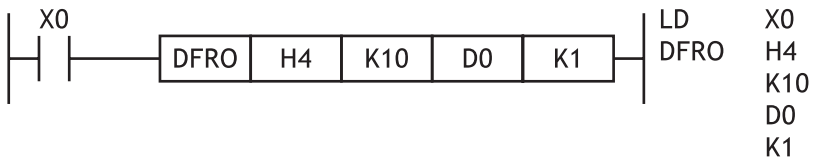
### FROM

Följande program läser ett dataord från buffertadress 10 i en funktionsmoduls buffertminne. Modulens adressområde ligger mellan 040 och 05F. Den lästa datan lagras i D0.




### DFRO

Följande program läser två dataord från adress 10 och 11 i en funktionsmoduls buffertminne. Modulens adressområde ligger mellan 040 och 05F. Den lästa datan lagras i D0 och D1.



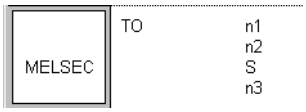
## 8.23.2 Skriv till funktionsmodul, TO

| TO, TOP, DTO, DTOP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 5 |

**Tillåtna operander**

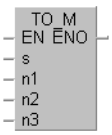
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | U | |
| n1 | • | • | • | • | • | • | • | • | • | SM0 |
| n2 | • | • | • | • | • | • | • | • | - | |
| D | • | • | • | - | - | - | - | • | - | |
| n3 | • | • | • | • | • | • | • | • | - | |

**MELSEC Instruktionslista**



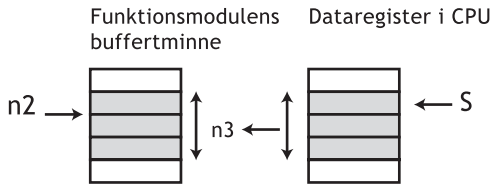

**Ladder**

**Funktionsblock**




# TO

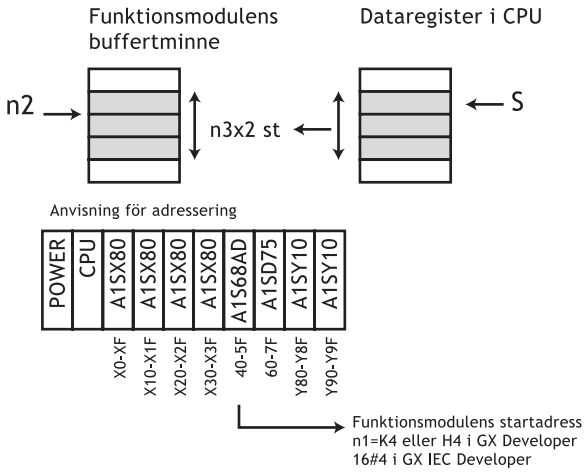
TO-instruktionen läser ett enkelord i ett definierat register i centralenheten och skriver det till en funktionsmoduls buffertminne. Startadressen för buffertminnet anges i n2 och antalet buffertminnen som skall skrivas till anges i n3. Modulens grundadress anges i n1. Källadressen från vilken datan skall läsas anges i (S).




**Observera!**
I GX IEC Developer skall n1, n2, n3 och S definieras som en enkelordsvariabel (t ex INT)

# DTO

DTO-instruktionen läser dubbelord (2 register) i ett definierat register i centralenheten och skriver det till en funktionsmoduls buffertminne. Startadressen för överföringen anges i n2 och antalet buffertminnen som skall skrivas anges i n3 och modulens grundadress anges med n1. Källadressen från vilken datan skall läsas anges i S.




**Observera!**
I GX IEC Developer skall n1, n2, n3 definieras som en enkelordsvariabel (t ex INT) och S definieras som en dubbelordsvariabel (t ex DINT)

Fel i programbearbetningen inträffar när:

– Fel typ av modul sitter på angiven I/O adress n1 .

– Funktionsmodulen är ej tillgänglig.

– Antalet definierade dataord n3 överstiger det i D angivna minnesområdet.

– Buffertminnesadressen angiven i n2 är utanför tillåtet område

I samtliga fall kommer felflagga att sättas.

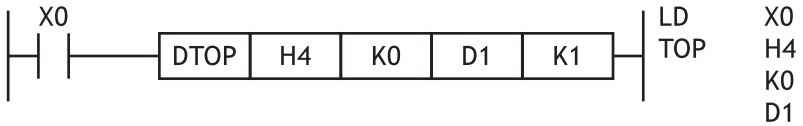
## Programexempel

### TO

Följande program skriver in talet 3 i buffertminne 0 i en funktionsmodul när X0 aktiveras. Modulen finns på adressplats 040 till 05F.

### DTO

Programmet skriver in två dataord från D1 och D2 till en funktionsmoduls buffertminnesadress 0 och 1 när X10 aktiveras. Modulen finns på adressplats mellan 040 till 05F.



# 8.24 Teckensträninstruktioner

## 8.24.1 Omvandla till ASCII-format, ASC

| ASC, ASCP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ●1) | 4 |

**Tillåtna operander**

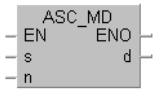
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | - | • | • | - | - | - | - | - | - | SM0 |
| D | - | • | • | - | - | - | - | - | - | |
| n | • | • | • | • | • | • | • | • | - | |

1) Instruktionen kan ej användas i Q00(J) eller Q01

**MELSEC Instruktionslista**

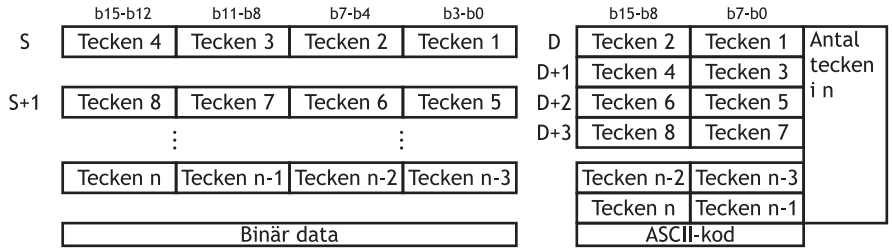
**Ladder**

**Funktionsblock**



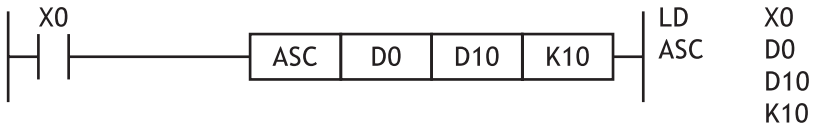

## ASC

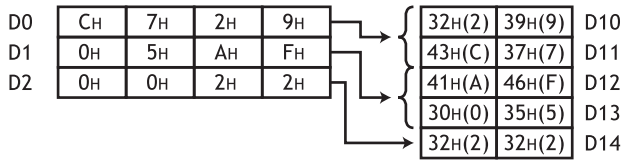
Omvandling av BIN 16-bitar till ASCII-kod. ASCII-instruktionen omvandlar datan i S och framåt till hexadecimalt ASCII-format och resultatet lagras i antal tecken specificerat av n från D och framåt.



## Programexempel

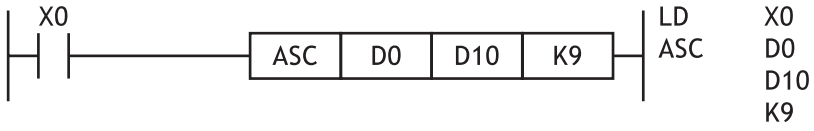
### ASC

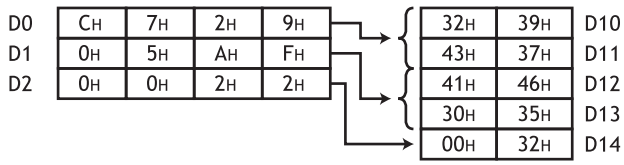



När X0 aktiveras omvandlas 10 tecken från binärkod till ASCII-kod. Binärkoden ligger i D0 och resultatet lagras i D10 och framåt.

Om n är ett udda tal kommer det övre tecknet i sista avlämningsadressen nollställas.




Även om avlämningsadresserna överlappar källdatan kommer instruktionen att utföras korrekt. Om värdet i n är 0 utförs omvandlingen inte.

I följande fall sätts felflaggan:

– Om antalet tecken är fler än vad som är angivet i n och kräver fler register för (S) och (D).

---

**Observera!**
I GX IEC Developer skall S, D, och n definieras som en enkelordsvariabel (t ex INT).

---

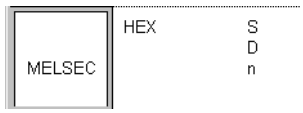
## 8.24.2 Omvandla till HEX-format, HEX

| HEX, HEXP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ●1) | |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J** | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | - | • | • | - | - | - | - | - | - | SM0 |
| D | - | • | • | - | - | - | - | - | - | |
| n | • | • | • | • | • | • | • | • | - | |

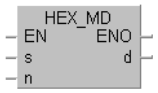
1) Instruktionen kan ej användas i Q00(J) eller Q01

**MELSEC Instruktionslista**




**Ladder**

**Funktionsblock**




# HEX

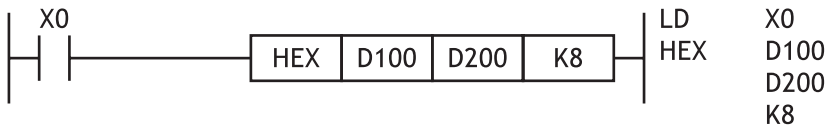
Omvandling av ASCII-kod till binärkod. HEX-instruktionen omvandlar teckensträngen i (S) och n tecken framåt till binärkod som lagras med början i (D).

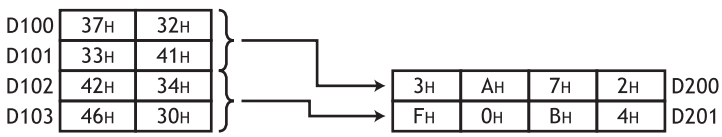
**Observera!**
I GX IEC Developer skall S, D, och n definieras som en enkelordsvariabel (t ex INT).

### Programexempel

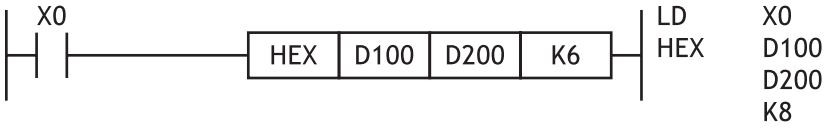
### HEX

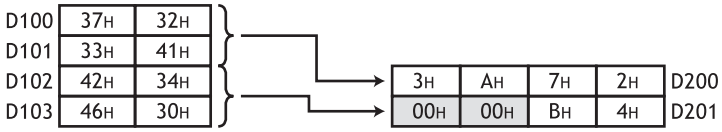
När X0 aktiveras omvandlas 8 tecken från ASCII-kod till binärkod. ASCII-koden som skall omvandlas ligger i D100 och resultatet lagras i D200.



Om n inte är jämt delbart med 4 kommer de högre bitarna i avlämningsadresserna att nollställas.




Instruktionen kommer att utföras korrekt även om avlämningsadressen överlappar källdatan.

Om värdet i n är 0 utförs omvandlingen inte.

I följande fall sätts felflaggan:

– Om källdatan innehåller andra tecken än “30H - ”39H” eller “41H” - “46H”, dvs 0-9 samt A-F.

– Om n är negativt.

– Om antalet tecken enligt n kräver fler register för (S) eller (D) än vad som tillgängligt enligt parametrarna.

# 8.25 Specialfunktioner

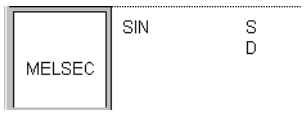
## 8.25.1 Sinusomvandling av flyttal, SIN

| SIN, SINP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ●1) | 3 |

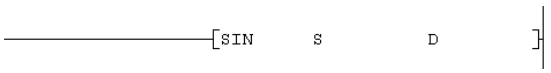
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | E | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | - | • | • | - | • | • | - | • | - | SM0 |
| D | - | • | • | - | • | • | - | - | - | |

1) Kräver revision B av Q00(J) och Q01

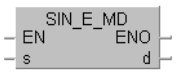
**MELSEC Instruktionslista**




**Ladder**

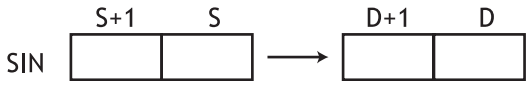



**Funktionsblock**




## SIN

Instruktionen utför en sinusberäkning på ett flyttal i (S) och (S)+1 och lagrar resultatet i (D) och (D)+1.




Värdet i (S) och (S)+1 måste vara en vinkel mellan 0 och 360 grader i radianer (grader x π / 180).

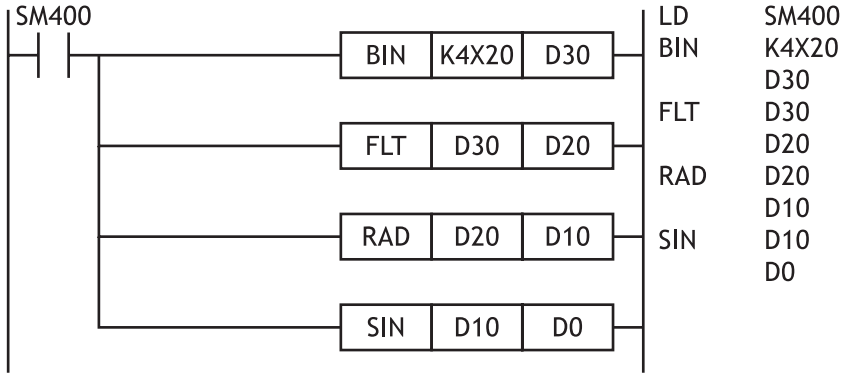
För att omvandla grader till radianer se instruktionerna RAD ooch DEG.

**Observera!**
I GX IEC Developer skall S och D definieras som en flyttalsvariabel (REAL).

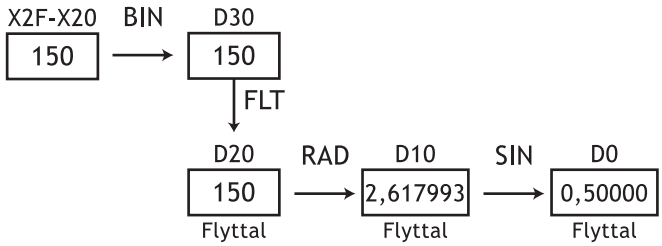
## Programexempel

### SIN

Programmet omvandlar ingångarna X20-X2F till ett binärvärde, binärvärdet omvandlas till ett flyttal. Efter omvandling till radianer kan SIN-instruktionen utföras.




Beräkning av SIN 150




Felflaggan sätts som (S), (S)+1 är 0 (endast Q4AR).

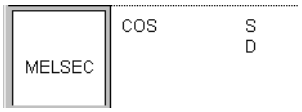
## 8.25.2 Cosinusomvandling av flyttal, COS

| COS, COSP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ●1) | 3 |

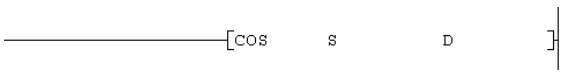
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | E | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | - | • | • | - | • | • | - | • | - | SM0 |
| D | - | • | • | - | • | • | - | - | - | |

1) Kräver revision B av Q00(J) och Q01

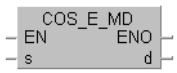
**MELSEC Instruktionslista**




**Ladder**

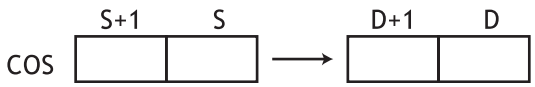



**Funktionsblock**




# COS

Denna instruktion utför en cosinusberäkning på ett flyttal i (S) och (S)+1 och lagrar resultatet i (D) och (D+1).




Värdet i S och S+1 måste vara en vinkel mellan 0 och 360 grader i radianer

(grader x π / 180).

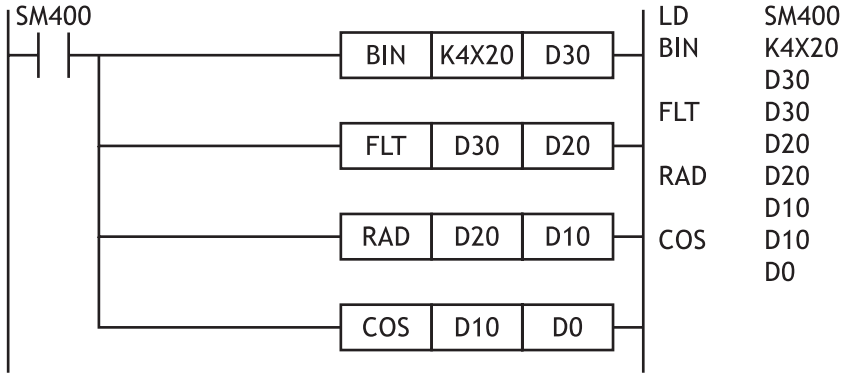
För att omvandla grader till radianer se instruktionerna RAD ooch DEG.

**Observera!**
I GX IEC Developer skall S och D definieras som en flyttalsvariabel (REAL).

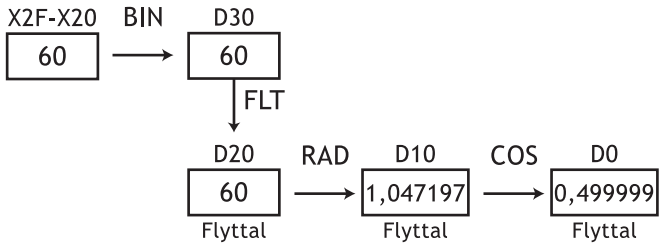
## Programexempel

### COS

Programmet omvandlar ingångarna X2F-X20F till ett binärvärde, binärvärdet omvandlas till ett flyttal. Efter omvandling till radianer kan COS-instruktionen utföras.




Beräkning av COS 60




Felflaggan sätts som (S), (S)+1 är 0 (endast Q4AR).

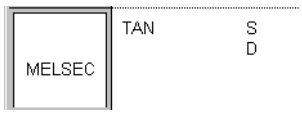
## 8.25.3 Tangensomvandling av flyttal, TAN

| TAN, TANP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● 1) | 3 |

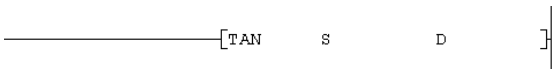
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | E | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | - | • | • | - | • | • | - | • | - | SM0 |
| D | - | • | • | - | • | • | - | - | - | |

1) Kräver revision B av Q00(J) och Q01

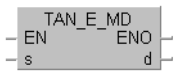
**MELSEC Instruktionslista**




**Ladder**

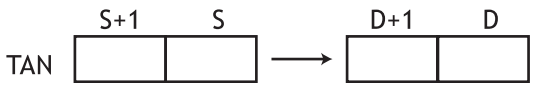



**Funktionsblock**




## TAN

Denna instruktion utför en tangensberäkning på ett flyttal i (S) och (S)+1 och lagrar resultatet i (D) och (D+1).




Värdet i (S) och (S)+1 måste vara en vinkel mellan 0 och 360 grader i radianer (grader x π / 180).

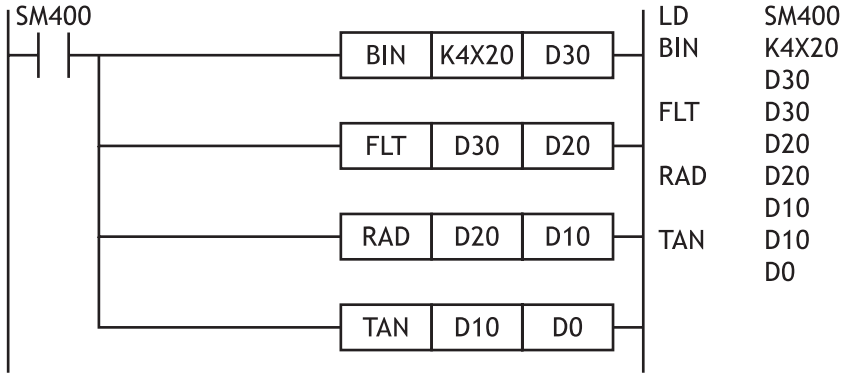
För att omvandla grader till radianer se instruktionerna RAD ooch DEG.

**Observera!**
I GX IEC Developer skall S och D definieras som en flyttalsvariabel (REAL).

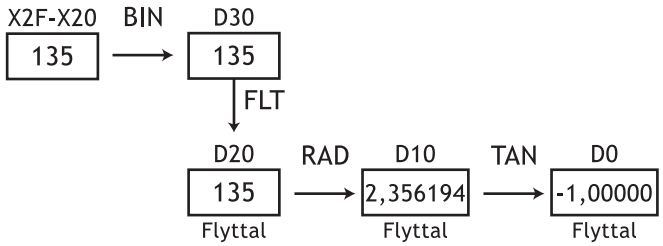
## Programexempel

### TAN

Programmet omvandlar ingångarna X20-X2F till ett binärvärde, binärvärdet omvandlas till ett flyttal. Efter omvandling till radianer kan TAN-instruktionen utföras.




Beräkning av TAN 135




Felflaggan sätts om:

– (S), (S)+1 är 0 (endast Q4AR)

– Om resultat av beräkningen är 0 eller utanför området $\pm 2^{-127} < (D), 0 < \pm 2^{129}$

## 8.25.4 Arcussinusomvandling av flyttal, ASIN

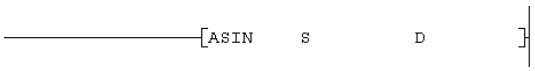
| ASIN, ASINP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● 1) | 3 |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | E | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | - | • | • | - | • | • | - | • | - | SM0 |
| D | - | • | • | - | • | • | - | - | - | |

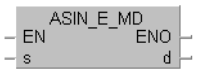
1) Instruktionen kan ej användas i Q00(J) eller Q01

**MELSEC Instruktionslista**

**Ladder**

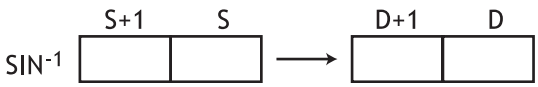



**Funktionsblock**




## ASIN

Denna instruktion utför en arcussinusberäkning på ett flyttal i (S) och (S)+1 och lagrar resultatet i (D) och (D+1).




Värdet i S och S+1 måste vara mellan -1 och +1. Resultatet i (D), (D)+1 lagras som radianer.

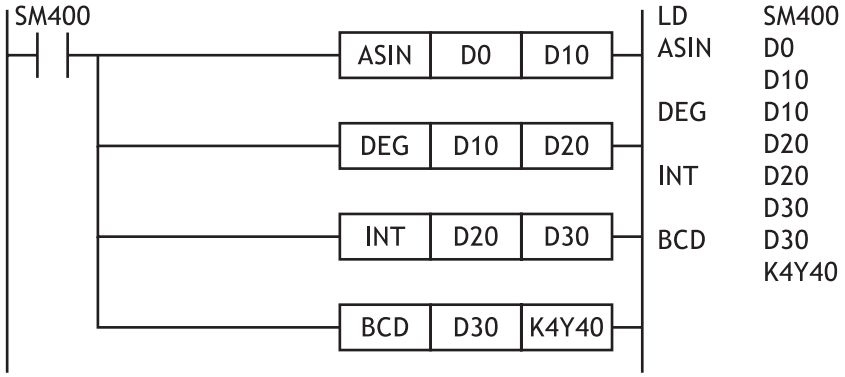
För att omvandla radianer till grader se instruktionerna RAD ooch DEG.

**Observera!**
I GX IEC Developer skall S och D definieras som en flyttalsvariabel (REAL).

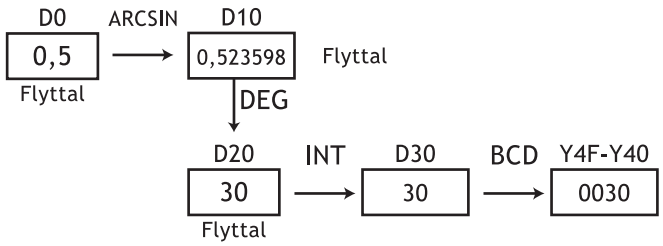
## Programexempel

### ASIN

Programmet omvandlar arcsinvärdet i D0 till grader för presentation i BCD-kod på utgångarna Y40-Y4F.




Beräkning av ASIN 0,5



## 8.25.5 Arcuscosinusomvandling av flyttal, ACOS

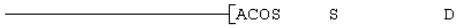
| ACOS, ACOSP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ●1) | 3 |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | E | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | - | • | • | - | • | • | - | • | - | SM0 |
| D | - | • | • | - | • | • | - | - | - | |

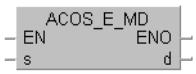
1) Instruktionen kan ej användas i Q00(J) eller Q01

**MELSEC Instruktionslista**

**Ladder**

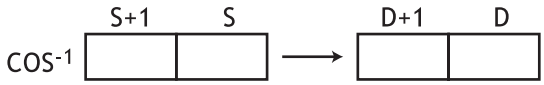



**Funktionsblock**




# ACOS

Denna instruktion utför en arcuscosinusberäkning på ett flyttal i (S) och (S)+1 och lagrar resultatet i (D) och (D+1).




Värdet i (S) och (S)+1 måste vara mellan -1 och +1. Resultatet i (D), (D)+1 lagras som radianer.

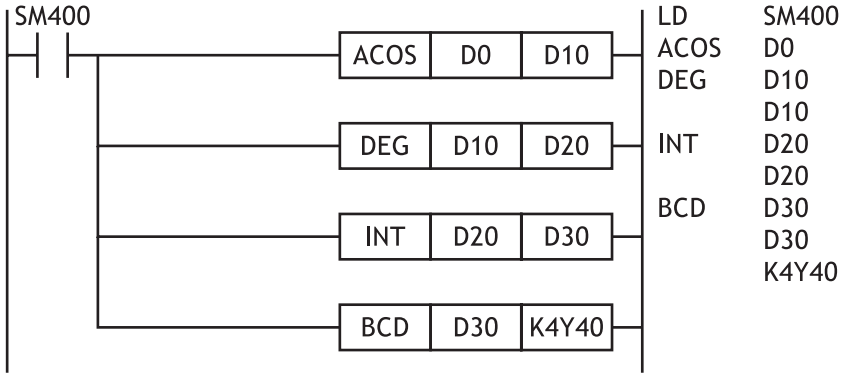
För att omvandla radianer till grader se instruktionerna RAD och DEG.

**Observera!**
I GX IEC Developer skall S och D definieras som en flyttalsvariabel (REAL).

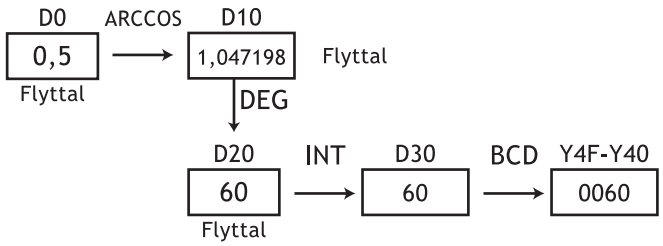
## Programexempel

### ACOS

Programmet omvandlar arccosvärdet i D0 till grader för presentation i BCD-kod på utgångarna Y40-Y4F.




Beräkning av ACOS 0,5



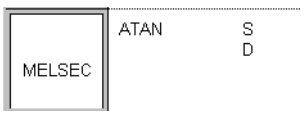
## 8.25.6 Arcustangensomvandling av flyttal, ATAN

| ATAN, ATANP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● 1) | 3 |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | E | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | - | • | • | - | • | • | - | • | - | SM0 |
| D | - | • | • | - | • | • | - | - | - | |

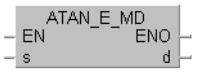
1) Instruktionen kan ej användas i Q00(J) eller Q01

**MELSEC Instruktionslista**



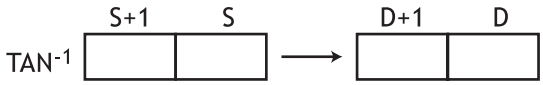

**Ladder**

**Funktionsblock**




## ATAN

Denna instruktion utför en arcustangensberäkning på ett flyttal i (S) och (S)+1 och lagrar resultatet i (D) och (D)+1.




Värdet i (S) och (S)+1 måste vara mellan -1 och +1. Resultatet i (D), (D)+1 lagras som radianer.

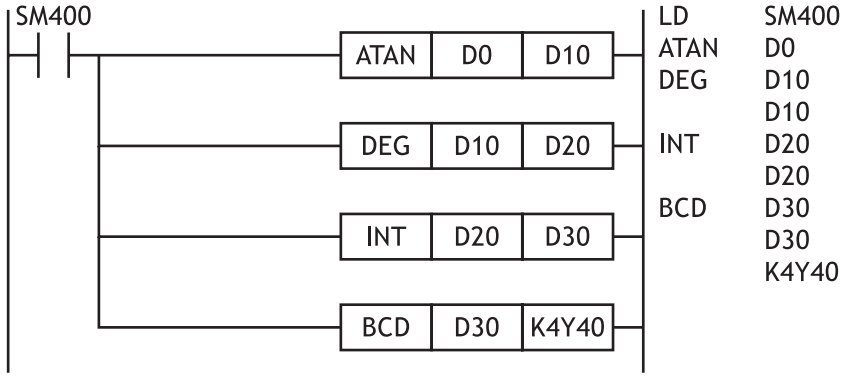
För att omvandla radianer till grader se instruktionerna RAD och DEG.

**Observera!**
I GX IEC Developer skall S och D definieras som en flyttalsvariabel (REAL).

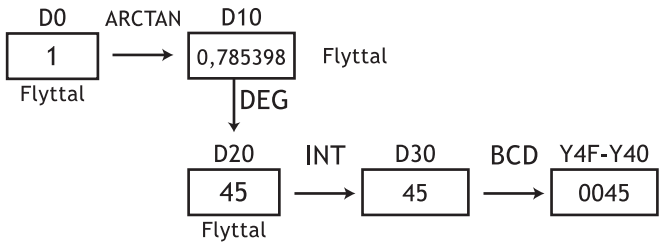
## Programexempel

### ATAN

Programmet omvandlar arctanvärdet i D0 till grader för presentation i BCD-kod på utgångarna Y40-Y4F.




Beräkning av ATAN 1



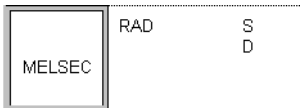
## 8.25.7 Radianomvandling av flyttal, RAD

| RAD, RADP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ●1) | 3 |

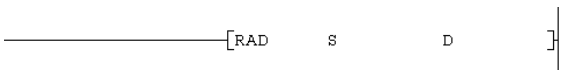
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | E | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | - | • | • | - | • | • | - | • | - | - |
| D | - | • | • | - | • | • | - | - | - | |

1) Kräver revision B av Q00(J) och Q01

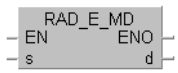
**MELSEC Instruktionslista**




**Ladder**

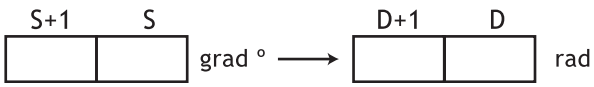



**Funktionsblock**




# RAD

Denna instruktion omvandlar ett flyttal i (S) och (S)+1 från grader till radianer och lagrar resultatet i (D) och (D)+1.



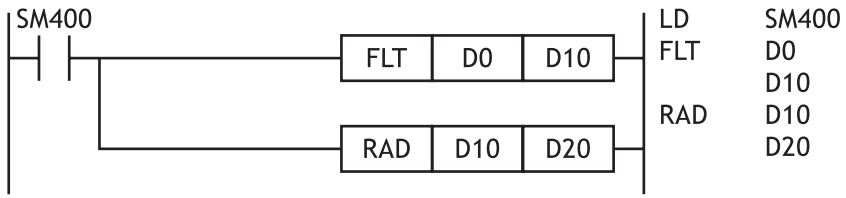

Omvandling från grader till radianer sker enligt formeln: grader x 180/ π.

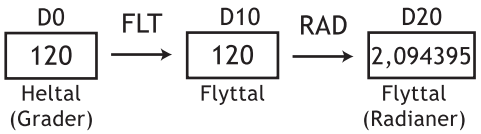
**Observera!**
I GX IEC Developer skall S och D definieras som en flyttalsvariabel (REAL).

### Programexempel

#### RAD

Programmet omvandlar gradtalet i D0 till flyttal och radianer.



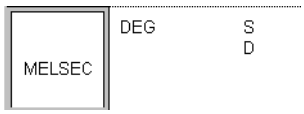
## 8.25.8 Gradomvandling av flyttal, DEG

| DEG, DEGP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ●1) | 3 |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | E | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | - | • | • | - | • | • | - | • | - | - |
| D | - | • | • | - | • | • | - | - | - | |

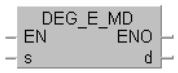
1) Kräver revision B av Q00(J) och Q01

**MELSEC Instruktionslista**




**Ladder**

**Funktionsblock**




# DEG

Denna instruktion omvandlar ett flyttal i (S) och (S)+1 från radianer till grader och lagrar resultatet i (D) och (D)+1.

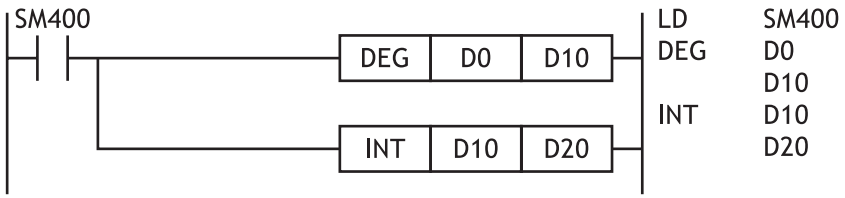
Omvandling från grader till radianer sker enligt formeln: grader= radianer x 180/ π.

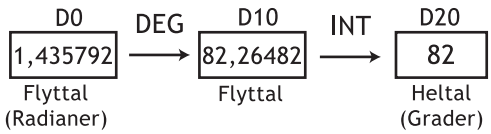
**Observera!**
I GX IEC Developer skall S och D definieras som en flyttalsvariabel (REAL).

### Programexempel

#### DEG

Programmet omvandlar gradtalet i D0 till flyttal och radianer.



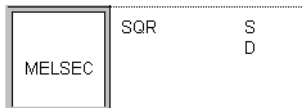
## 8.25.9 Kvardatrot ur flyttal, SQR

| SQR, SQRP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ●1) | 3 |

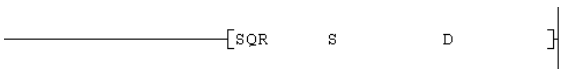
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | E | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | - | • | • | - | • | • | - | • | - | SM0 |
| D | - | • | • | - | • | • | - | - | - | |

1) Kräver revision B av Q00(J) och Q01

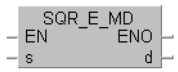
**MELSEC Instruktionslista**




**Ladder**

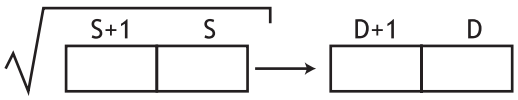



**Funktionsblock**




# SQR

Instruktionen beräknar kvdratroten ur flyttalet i (S) och (S)+1 och lagrar resultatet i (D) och (D+1).



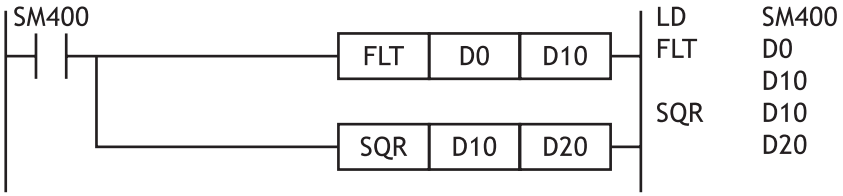

Felflaggan sätts om värdet i (S), (S)+1 är negativt.

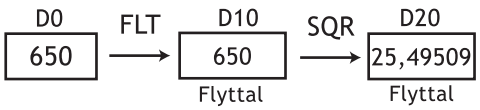
**Observera!**
I GX IEC Developer skall S och D definieras som en flyttalsvariabel (REAL).

### Programexempel

### SQR

Programmet omvandlar binärvärdet i D0 till flyttal och beräknar talets kvadratrot.



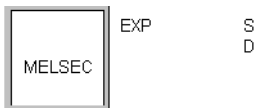
# 8.25.10 Logaritmomvandling av flyttal, EXP

| EXP, EXPP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● 1) | 3 |

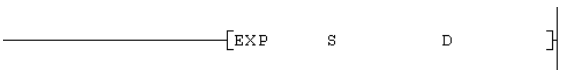
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | E | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | - | • | • | - | • | • | - | • | - | SM0 |
| D | - | • | • | - | • | • | - | - | - | |

1) Kräver revision B av Q00(J) och Q01

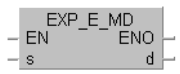
**MELSEC Instruktionslista**




**Ladder**




**Funktionsblock**



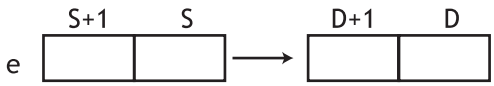

## EXP

Instruktionen använder flyttalet i (S), (S)+1 för att beräkna värdet av e $^{(S),\ (S)+1}$. Resultatet lagras i (D), (D)+1.

Felflaggan sätts om:

– Exponentvärdet är större än 89.

– Resultatet får ej vara utanför gränserna $2^{-127}$ - $2^{129}$

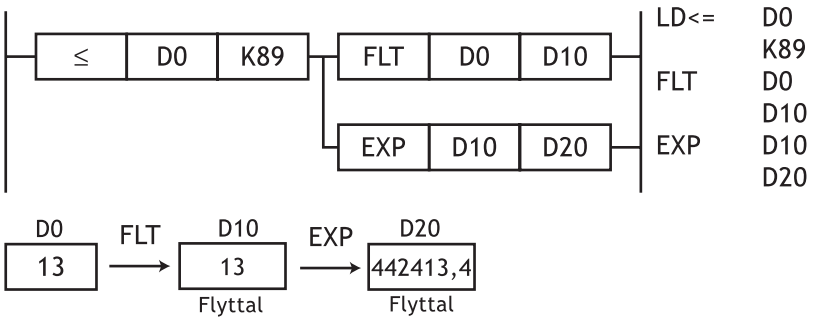



**Observera!**
I GX IEC Developer skall S och D definieras som en flyttalsvariabel (REAL).

## Programexempel

### EXP

Programmet kontrollerar att exponenten ej är större än 89 innan beräkningen sker.



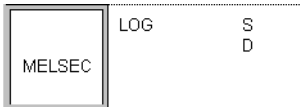
# 8.25.11 Logaritmomvandling av flyttal, LOG

| LOG, LOGP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● 1) | 3 |

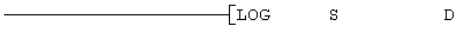
| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | E | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | - | • | • | - | • | • | - | • | - | SM0 |
| D | - | • | • | - | • | • | - | - | - | |

1) Kräver revision B av Q00(J) och Q01

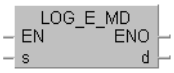
**MELSEC Instruktionslista**




**Ladder**




**Funktionsblock**



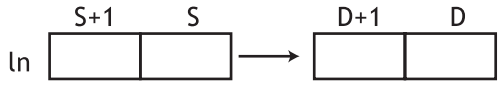

## LOG

Instruktionen beräknar logaritmen ur ett flyttal i (S) och (S)+1, (ln(S,S+1). Resultatet lagras (D) och (D)+1.

Felflaggan sätts om:

– Värdet i (S) och (S)+1 är negativt.

– Eller om resultatet av beräkningen överskrider gränserna $2^{-127}$ - $2^{129}$

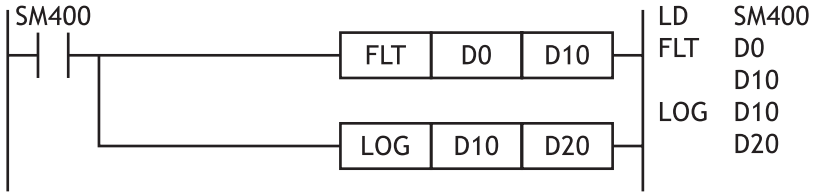


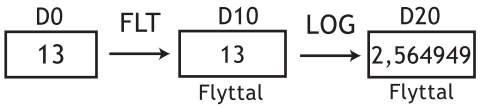

**Observera!**
I GX IEC Developer skall S och D definieras som en flyttalsvariabel (REAL).

## Programexempel

### LOG

Programmet omvandlar heltalet i D10 till flyttal och beräknar logaritmen och lagrar resultatet i D30.



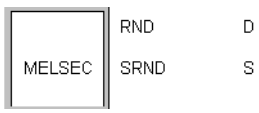
# 8.25.12 Generering av slumptal, RND, SRND

| RND, RNDP, SRND, SRNDP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ●1) | 3 |

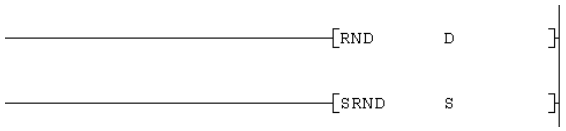
| Tillåtna operander | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | | | | | | | - | - | - |

1) Kräver revision B av Q00(J) och Q01

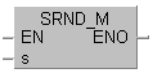
**MELSEC Instruktionslista**




**Ladder**




**Funktionsblock**




## RND

RND-instruktionen genererar ett slumpvärde inom området 0 till 32 767 och lagrar resultatet i D.

**Observera!**
I GX IEC Developer skall D definieras som en enkelordsvariabel (t ex INT).

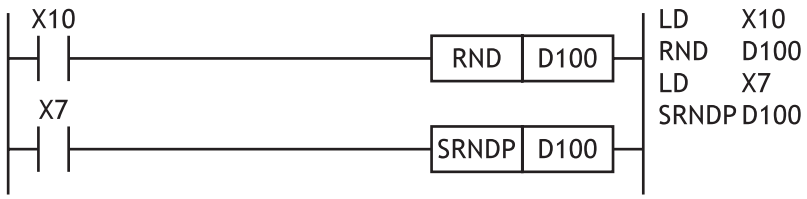
## SRND

Uppdaterar serien av slumpvärden lagrade i S.

**Observera!**
I GX IEC Developer skall S definieras som en enkelordsvariabel (t ex INT).

### Programexempel

När X10 är ettställd lagras ett slumptal i D100, på positiv flank på X7 genereras en ny slumptalsrerie i D100.



# 8.26 Datakontrollinstruktioner

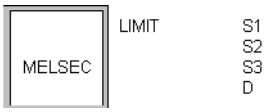
## 8.26.1 Begränsning av utgångsvärden, LIMIT

| LIMIT, LIMITP, DLIMIT, DLIMITP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 5 |

**Tillåtna operander**

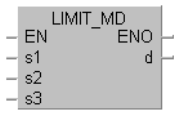
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S1 | • | • | • | • | • | • | • | • | - | SM0 |
| S2 | • | • | • | • | • | • | • | • | - | |
| S3 | • | • | • | • | • | • | • | • | - | |
| D | • | • | • | • | • | • | • | - | - | |

**MELSEC Instruktionslista**




**Ladder**

**Funktionsblock**




## LIMIT

Begränsning av binära utgångsvärden, enkelord.

Instruktionen kontrollerar om data i operanden specificerad i S3 är inom området för undre gränsvärdet S1 och för övre gränsvärdet S2.

| Operand | Funktion | Datatyp |
|---|---|---|
| S1 | Undre gränsvärde (utgång) | BIN 16 bit |
| S2 | Övre gränsvärde (utgång) | |
| S3 | Ingångsvärde som skall begränsas | |
| D | Första lagrade utgångs begränsningsvärdet | |

Om värdet i S3 är större än värdet i S2 skickas S2 ut på D.
Om värdet i S3 är mindre än värdet i S1 skickas S1 ut på D.
S1, S2 och S3 måste vara inom området -32 768 - 32 767.
Om endast den övre gränsen skall kontrolleras skall S1 sättas till -32 768.
Om endast den undre gränsen skall kontrolleras skall S2 sättas till 32 767.

Felflaggan sätts om:

– Värdet i S1 är större än värdet i S2.

**Observera!**
I GX IEC Developer skall S1, S2, S3 och D definieras som en enkelordsvariabel (t ex INT).

# DLIMIT

Begränsning av binära utgångsvärden, dubbelord.

Instruktionen kontrollerar om data i operanden specificerad i (S3+1, S3) är inom området för undre gränsvärdet (S1+1, S1) och för övre gränsvärdet (S2+1, S2).

| Operand | Funktion | Datatyp |
|---|---|---|
| S1 | Undre gränsvärde (utgång) | BIN 32 bit |
| S2 | Övre gränsvärde (utgång) | |
| S3 | Ingångsvärde som skall begränsas | |
| D | Första lagrade utgångs begränsningsvärdet | |

Felflaggan sätts om:

– Värdet i (S1+1, S1) är större än värdet i (S2+1, S2).

Om värdet i (S3+1, S3) är större än värdet i (S2+1, S2) skickas (S2+1, S2) ut på (D+1, D).

Om värdet i (S3+1, S3) är mindre än värdet i (S1+1, S1) skickas (S1+1, S1) ut på (D+1, D).

(S1+1, S1) , (S2+1, S2) och (S3+1, S3) måste vara inom området -2 147 483 648 - 2 147 483 647.

Om endast den övre gränsen skall kontrolleras skall (S1+1, S1) sättas till -2 147 483 648.

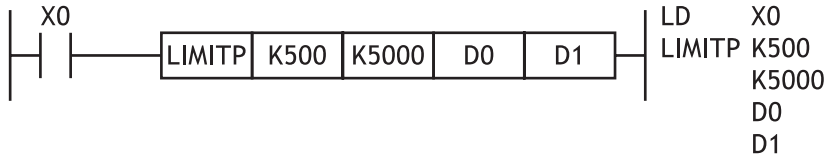
Om endast den undre gränsen skall kontrolleras skall (S2+1, S2) sättas till 2 147 483 647

**Observera!**
I GX IEC Developer skall S1, S2, S3 och D definieras som en dubbelordsvariabel (t ex DINT).

## Programexempel

### LIMIT

På positiv flank på X0 kontrollerar programmet om värdet i D0 är inom området 500-5000.




Om D0 är mindre än 500 lagras 500 i D1.

Om D0 är större än 5000 lagras 5000 i D1.

Om D0 är inom gränserna lagras D0 i D1.

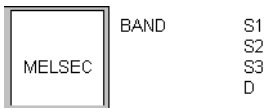
## 8.26.2 Dödbandskontroll, BAND

| BAND, BANDP, DBAND, DBANDP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | • | • | • | 5 |

**Tillåtna operander**

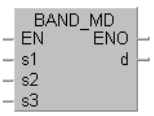
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S1 | • | • | • | • | • | • | • | • | - | SM0 |
| S2 | • | • | • | • | • | • | • | • | - | |
| S3 | • | • | • | • | • | • | • | • | - | |
| D | • | • | • | • | • | • | • | - | - | |

**MELSEC Instruktionslista**




**Ladder**

**Funktionsblock**




## BAND

Dödbandkontroll av enkelord.

Instruktionen subtraherar ett negativt och ett positivt gränsvärdet från det binära ingångsvärdet i S3. Om värdena i S3 är inom området, lagras det i D=0.

| Operand | Funktion | Datatyp |
|---|---|---|
| S1 | Undre gränsvärde dödband (utgångsvärde=0) (negativt) | BIN 16 bit |
| S2 | Övre gränsvärde dödband (utgångsvärde=0) (positivt) | |
| S3 | Ingångsvärde som skall begränsas via dödbandskontroll | |
| D | Första lagrade resultatet av subtrakraktionen av ingångsvärdet minus gränsvärdet | |

Om värdet i S3 är mindre än det undre gränsvärdet i S1 innehåller D värdet S3-S1.
Om värdet i S3 är större än det övre gränsvärdet i S2 innehåller D värdet S3-S2.
Om värdet i S3 är inom gränserna innhåller D värdet noll.
Värdena S1, S2 och S3 måste vara inom gränserna -32 768-32 767
Om resultatet efter subtraktionen hamnar utanför gränserna -32 768 - 32 767 justeras värdet enligt följande regler:
Om S3 innehåller värdet -32 768 och värdet i S2, 10 blir resultatet efter subtraktion:
-32 768-10=H8000-HA =H7FF6=32758

Felflaggan sätts om S1 är större än S2

**Observera!**
I GX IEC Developer skall S1, S2, S3 och D definieras som en enkelordsvariabel (t ex INT).

# DBAND

Dödbandkontroll av dubbelord.

Instruktionen subtraherar ett negativt och ett positivt gränsvärdet från det binära ingångsvärdet i S3. Om värdena i S3 är inom området, lagras det i D=0.

| Operand | Funktion | Datatyp |
|---|---|---|
| S1 | Undre gränsvärde dödband (utgångsvärde=0) (negativt) | BIN 32 bit |
| S2 | Övre gränsvärde dödband (utgångsvärde=0) (positivt) | |
| S3 | Ingångsvärde som skall begränsas via dödbandskontroll | |
| D | Första lagrade resultatet av subtrakraktionen av ingångsvärdet minus gränsvärdet | |

Om värdet i (S3+1, S3) är mindre än det undre gränsvärdet i (S1+1, S1), innehåller (D+1, D) värdet (S3+1, S3) - (S1+1, S1).
Om värdet i (S3+1, S3) är större än det övre gränsvärdet i (S2+1, S2) innehåller (D+1, D) värdet (S3+1, S3) - (S2+1, S2).
Om värdet i (S3+1, S3) är inom gränserna innhåller (D+1, D) värdet noll.
Värdena i (S1+1, S1), (S2+1, S2) och (S3+1, S3) måste vara inom gränserna -2 147 483 648 - 2 147 483 647.
Om resultatet efter subtraktionen hamnar utanför gränserna -2 147 483 648 - 2 147 483 647 justeras värdet enligt följande regler:
Om (S3+1, S3) innehåller värdet -2 147 483 648 och värdet i (S2+1, S2) är 1000 blir resultatet efter subtraktionen:
-2 147 483 648 -1000=H80000000-H3E8=H7FFFFC18=2 147 482 648

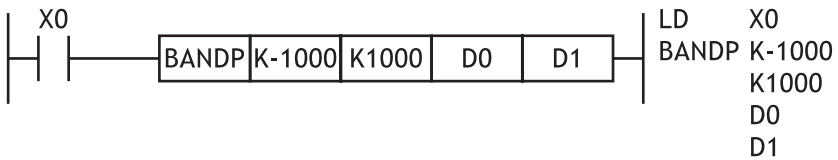
Felflaggan sätts om (S1+1, S1) är större än (S2+1, S2)

**Observera!**
I GX IEC Developer skall S1, S2, S3 och D definieras som en dubbelordsvariabel (t ex DINT).

## Programexempel

### BAND

På positiv flank på X0 gör programmet en dödbandskontroll på dataregister D0.




Om D0 är större än 1000 lagras värdet i D0-1000 i D1.

Om D0 är mindre än -1000 lagras värdet i D0-(-1000) i D1.

Om D0 är inom gränserna -1000-1000 lagras värdet 0 i D1.

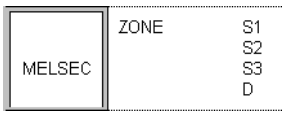
## 8.26.3 Zonkontroll, ZONE

| ZONE, ZONEP, DZONE, DZONEP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | • | • | • | 5 |

**Tillåtna operander**

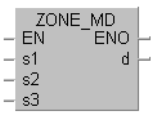
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S1 | • | • | • | • | • | • | • | • | - | SM0 |
| S2 | • | • | • | • | • | • | • | • | - | |
| S3 | • | • | • | • | • | • | • | • | - | |
| D | • | • | • | • | • | • | • | - | - | |

**MELSEC Instruktionslista**




**Ladder**

**Funktionsblock**




## ZONE

Instruktionen adderar ett negativt och ett positivt värde till ett binärt enkelord i S3. Det negativa värdet lagras i S1 och det positiva värdet lagras i S2. Resultatet lagras i D.

Om värdet i S3 är mindre än noll, innehåller D värdet S3+S1.
Om värdet i S3 är större än noll, innehåller D värdet S3+S2.
Om värdet i S3 är lika med noll, innehåller D värdet noll.

| Operand | Funktion | Datatyp |
|---|---|---|
| S1 | Negativt zonkontrollvärde, adderas till S3 | BIN 16 bit |
| S2 | Positivt zonkontrollvärde, adderas till S3 | |
| S3 | Börvärde | |
| D | Börvärdet + zonkontrollvärdet | |

Värdet i S1, S2, och S3 måste vara inom gränserna -32 768 - 32 767

Om resultatet efter additionen hamnar utanför -32 768 - 32 767 justeras värdet enligt följande:
Om S3 innehåller värdet -32 768 och värdet i S1 är -100 bli resultatet efter additionen:
-32 768+ (-100)=H8000+HFF9C=H7F9C=H7F9C=32 668
Felflaggan sätts om S1 är större än S2.

**Observera!**
I GX IEC Developer skall S1, S2, S3 och D definieras som en enkelordsvariabel (t ex INT).

# DZONE

Instruktionen adderar ett negativt och ett positivt värde till ett binärt dubbelord i (S3+1, S3). Det negativa värdet lagras i (S1+1, S1) och det positiva värdet lagras i (S2+1, S2). Resultatet lagras i (D+1, D)

Om värdet i (S3+1, S3) är mindre än noll, innehåller (D+1, D) värdet (S3+1, S3) + (S1+1, S1).

Om värdet i (S3+1, S3) är större än noll, innehåller (D+1, D) värdet (S3+1, S3)+(S2+1, S2).

Om värdet i (S3+1, S3) är lika med noll, linnehåller (D+1, D) värdet noll.

| Operand | Funktion | Datatyp |
|---|---|---|
| S1 | Negativt zonkontrollvärde, adderas till S3 | BIN 32 bit |
| S2 | Positivt zonkontrollvärde, adderas till S3 | |
| S3 | Börvärde | |
| D | Börvärdet + zonkontrollvärdet | |

## 32 bitar DZONE

Värdena i (S1+1, S1), (S2+1, S2) och (S3+1, S3) måste vara inom gränserna -2 147 483 648 - 2 147 483 647.

Om resultatet efter additionen hamnar utanför gränserna -2 147 483 648 - 2 147 483 647 justeras värdet enligt följande regler:

Om (S3+1, S3) innehåller värdet -2 147 483 648 och värdet i (S1+1, S!) är -1000 blir resultatet efter additionen:

-2 147 483 648 +( -1000)=H80000000+HFFFFF18H=7FFFFC18=2 147 482 648

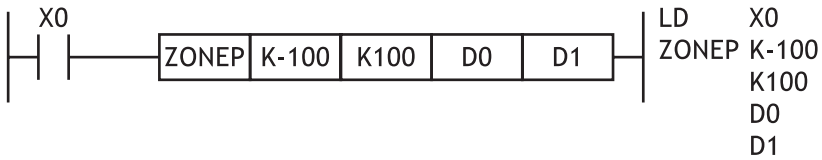
Felflaggan sätts om (S1+1, S1) är större än (S2+1, S2).

**Observera!**
I GX IEC Developer skall S1, S2, S3 och D definieras som en dubbelordsvariabel (t ex DINT).

## Programexempel

### ZONE

På positiv flank på X0 gör programmet en zonkontroll på dataregister D0.




Om D0 är större än noll, lagras värdet i D0+100 i D1.

Om D0 är mindre än noll, lagras värdet i D0+(-100) i D1.

Om D0 är lika med noll, lagras värdet noll i D1.

# 8.27 Filregister, växlingsinstruktion

## 8.27.1 Val av filregisterblock, RSET

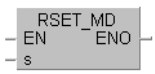
| RSET, RSETP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 2 |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| S | • | • | • | • | • | • | • | • | - | SM0 |

**MELSEC Instruktionslista**

**Ladder**

**Funktionsblock**




## RSET

Val av filregisterblock.

RSET-instruktionen växlar från ett filregisterblock som används av ett program till ett filregisterblock med adress S. Efter växlingen tar sekvensprogrammet över filregisterna i det adresserade blocket (R0-R32767).

Felflaggan sätts i följande fall:

– Om blocknumret i S ej existerar.

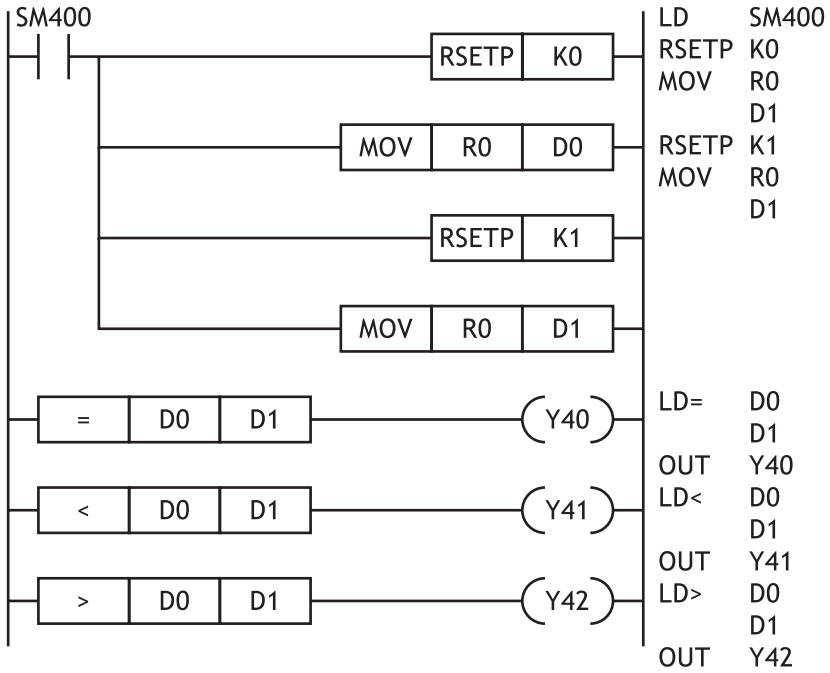
– Om det inte finns några filregister i block S.

**Observera!**
I GX IEC Developer skall S definieras som en enkelordsvariabel (t ex INT).

## Programexempel

### RSET

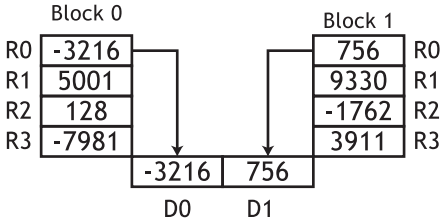
På positiv flank på SM400 jämförs filregister R0 i registerblock 0 med filregister R0 i registerblock 1. Båda registerna läses ut till vanliga dataregister med MOV-instruktionen.




Om R0 (block 0) = R0 (block 1) aktiveras Y40

Om R0 (block 0) < R0 (block 1) aktiveras Y41

Om R0 (block 0) > R0 (block 1) aktiveras Y42




I exemplet aktiveras Y41 eftersom D0 < D1

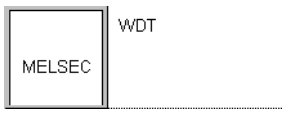
# 8.28 Andra instruktioner

## 8.28.1 Cykeltidsövervakning, WDT

| WDT, WDTP | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | 1 |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| - | - | - | - | - | - | - | - | - | - | - |

**MELSEC Instruktionslista**



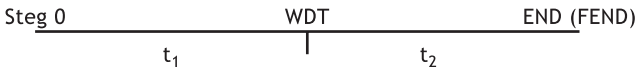

**Ladder**

**Funktionsblock**

## WDT

WDT-instruktionen återställer Watch-dog timer (övervakar programcykeltid). Instruktionen används då PLC programmets bearbetningstid riskerar överskrida det inställda WDT värdet. Sker detta regelbundet måste man förlänga tiden för WDT i parametrarna, utförs med en programmeringsenhet.

WDT tiden skall överstiga tiden från steg 0 till WDT och WDT till END.

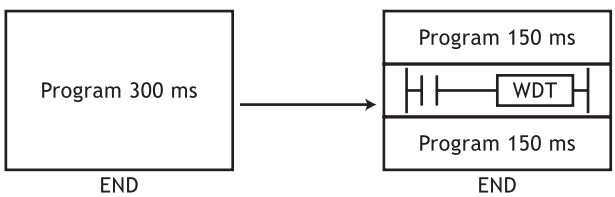



WDT-instruktionen kan användas flera gånger under ett programvarv. Man måste emellertid ta hänsyn till att då fel inträffar kan utgångarna inte nollställas förrän WDT-tiden har löpt ut. Dessutom kommer övervakningsfunktionen att förlängas i samband med många WDT-instruktioner.
Dessa återställs genom WDT-instruktionen. Specialregistervärdena måste därför vara större än WDT:s parametervärde.

### Programexempel

#### WDT

Programexempel som kan användas när inställd tid i WDT är 200 ms och cykeltiden är 300 ms.



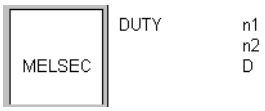
## 8.28.2 Programmerbart pulståg, DUTY

| DUTY | Q4AR | QnA(S) | Q | Programsteg |
|---|---|---|---|---|
| | ● | ● | ● | |

| Tillåtna operander | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Interna kanaler | | R, ZR | J*/* | | U*\G* | Zn | K, H(16#) | Övriga | Flaggor |
| | Bitar | Ord | | Bitar | Ord | | | | | |
| n1 | • | • | • | • | • | • | • | • | • | SM0 |
| n2 | • | • | • | • | • | • | • | • | • | |
| D | • 1) | - | - | - | - | - | - | - | - | |

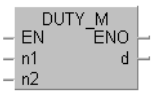
1) SM420 till SM424 och SM430 till SM434

**MELSEC Instruktionslista**




**Ladder**

**Funktionsblock**




# DUTY

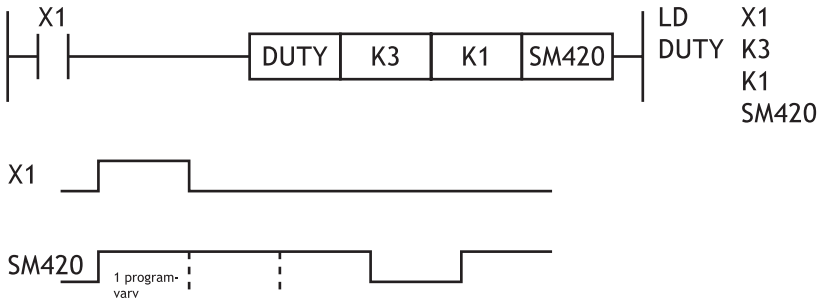
Med hjälp av DUTY-instruktionen kan pulståg med valfria ON/OFF-intervall programmeras. Specialminnescellerna SM420 till SM424 och SM430 till SM434 används för att generera pulstågen och definieras i (D). I n1 anges hur många programcykler som skall vara ettställda och i n2 anges hur många programcykler som skall vara nollställda.

**Observera!**
I GX IEC Developer skall n1 och n2 definieras som en enkelordsvariabel (t ex INT) och D som en bitvariabel (t ex BOOL)

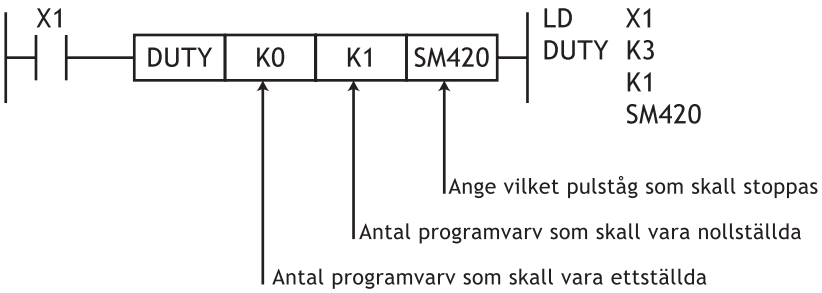
## Programexempel

### DUTY

När X1 blir aktiverad skapas ett pulståg med hjälp av SM4xx som innebär att specialminnescellen blir ettställd 3 programvarv och nollställd 1 programvarv.




Även om X1 nollställs fortsätter pulståget och kan avslutas med följande exempel.



# 9 Felkoder

## 9.1 Felkoder för MELSEC Q och QnA

En tabell över felkoderna finns i den engelska manualen för MELSEC Q och QnA.

# 10 Specialminnesceller

Interna specialminnesceller (SM) är bestämda till PLC-typen och kan inte användas som andra minnesceller i PLC-programmet. Däremot kan systemet ettställa och nollställa minnescellerna för att kontrollera PLC-systemet.

| | | |
|---|---|---|
| **Adress** | Specialminnescellernas adress | |
| **Namn** | Specialminnescellernas namn | |
| **Betydelse** | Specialminnescellernas betydelse | |
| **Beskrivning** | Detaljerad beskrivning av specialminnescellerna | |
| **Aktiveras av** | **Beskriver vem som aktiverar specialminnescellerna** | |
| | Systemet | S |
| | Aktiveras av användaren (via PLC-programmet eller programmeringsverktyget) | A |
| | Aktiveras av systemet eller användaren | S/A |
| | **Ytterligare beskrivning om specialminnescellerna sätts av systemet** | |
| | Aktiveras efter begäran från användaren | Begäran |
| | Aktiveras under instruktionsbearbetning | Instruktion |
| | Aktiveras efter att ett fel inträffat | Fel |
| | Aktiveras efter END-bearbetningen | END |
| | Aktiveras under initieringen | Initiering |
| | Aktiveras efter en statusförändring | Förändring |

# 10.1 Tabell specialminnesceller

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SM0 | Diagnostikfel | Från: Inget fel<br>Till: Fel | Ettställd om diagnosresultatet visar fel. Ligger till även om normaltillstånd uppnås. | S<br>(Fel) |
| SM1 | Självdiagnosfel | Från: Inget självdiagnosfel<br>Till: Självdiagnosfel | Ettställs då någon funktion som övervakas av sjävdiagnostiken är felaktig. | |
| SM5 | Allmän felinformation | Från: Ingen allmän information<br>Till: Allmän information finns | Om det finns allmän information och SM0 är till. | |
| SM16 | Individuell felinformation | Från: Ingen individuell felinformation<br>Till: Individuell felinformation finns | Om det finns individuell information och SM0 är till. | |
| SM50 | Felreset | Från>Till<br>Felåterställning | Reset funktion. Se ytterligare information, SD50. | A |

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SM51 | Låg batterinivå | Från: Normal<br>Till: Låg batterinivå | Ettställs då batterispänningen sjunker under specificerad nivå.<br>Återställs inte då spänningen stiger över nivån. | S<br>(Fel) |
| SM52 | Låg batterispänning | Från: Normal<br>Till: Låg batterinivå | Ettställs då batterispänningen sjunker under specificerad nivå.<br>Återställs då spänningen stiger över nivån. | |
| SM53 | AC-spänning<br>låg nivå | Från: Normal AC-spänning<br>Till: Låg AC-spänning | Ettställs då matningsspänningen är bruten mindre än 20 ms.<br>Återställ genom slå av och på spänningen igen.<br>Ettställs vid mycket korta spänningsavbrott. | |
| | | | Ettställs då matningsspänningen är bruten mindre än 10 ms.<br>Återställ genom slå av och på spänningen igen.<br>Ettställs vid mycket korta spänningsavbrott.<br>(Endast Q) | |
| | | | Ettställs då matningsspänningen är bruten mindre än 1 ms.<br>Återställ genom slå av och på spänningen igen.<br>Ettställs vid mycket korta spänningsavbrott.<br>(Endast QnA) | |
| SM54 | MINI NET fel | Från: Normal<br>Till: Error | Ettställs om ett MINI (S3)-linkfel upptäcks på någon av de anslutna AJ71PT32(S3)-modulerna.<br>Kvarstår om normaltillstånd uppnås.<br>(Endast QnA) | |
| SM56 | Funktionsfel | Från: Normal<br>Till: Funktionsfel | Ettställd om ett funktionsfel upptäcks. Ligger till även om normaltillstånd uppnås. | |
| SM60 | Säkringsbrott | Från: Normal<br>Till: Modul med säkringsbrott | Ettställd om någon utgångsmodul får ett säkringsbrott Ligger till även om normaltillstånd uppnås. | |
| SM61 | I/O modul<br>Verifikationsfel | Från: Normal<br>Till: Fel | Ettställd om skillnader finns i informationen om anslutna I/O-moduler och de fysiska modulerna. | |
| SM62 | Signaldetektion | Från: Inte detekterat<br>Till: Detekterat | Ettställd om en F-minnescell ettställs. | S<br>(Instruktion) |
| SM80 | CHK-detektion | Från: Inte detekterat<br>Till: Detekterat | Ettställd om fel detekterats av CHK-instruktionen.<br>Kvarstår om normaltillstånd uppnås.<br>(Ej Q00(J), Q01) | |

| Adress | Namn | Betydelse | Beskrivning | | Aktiveras av |
|---|---|---|---|---|---|
| SM90 | Aktivering av watchdog-timern för sekvensprogram (SFC). Går endast att ettställa när SFC-program finns. | Från: WDT ej startad<br>Till: WDT startad | Samverkar med SD90 | Ettställd när mätningen av stegöverföringen påbörjades. Reset WDT när den slås från. (Ej i Q00(J) eller Q01) | A |
| SM91 | | | Samverkar med SD91 | | |
| SM92 | | | Samverkar med SD92 | | |
| SM93 | | | Samverkar med SD93 | | |
| SM94 | | | Samverkar med SD94 | | |
| SM95 | | | Samverkar med SD95 | | |
| SM96 | | | Samverkar med SD96 | | |
| SM97 | | | Samverkar med SD97 | | |
| SM98 | | | Samverkar med SD98 | | |
| SM99 | | | Samverkar med SD99 | | |
| SM100 | Seriell kommunikationsfunktion | Från: Seriell kommunikation används ej<br>Till: Seriell kommunikation används | Indikator på om funktionen för seriell kommunikation är aktiverad eller ej.<br>(Endast i Q00(J) och Q01) | | S<br>(Spänningstillslag eller reset) |
| SM101 | Statusflagga, kommunikationsprotokoll | Från: Programmeringsprotokoll<br>Till: MC-protokoll | Indikator på om kommunikationen via RS232-gränssnittet sker via programmeringsprotokollet eller MC-protokollet.<br>(Endast i Q00(J) och Q01) | | S<br>(RS232-kommunikation) |
| SM110 | Protokollfel | Från: Inget fel<br>Till: Fel | Ettställs när ett felaktigt protokoll använts för att kommunicera seriellt. Förblir ettställd även om protokollet rättas till.<br>(Endast i Q00(J) och Q01) | | S<br>(Fel) |
| SM111 | Kommunikationsstatus | Från: Inget fel<br>Till: Fel | Ettställs när kommunikationssättet skiljer sig från inställningarna i funktionen seriell kommunikation.<br>Förblir ettställd även om kommunikationen återgår till det normala.<br>(Endast i Q00(J) och Q01) | | S<br>(Fel) |
| SM112 | Rensa felinformation | Till: Rensa diagnostiska specialminnesceller och specialdataregister. | När minnescellen ettställs, nollställs specialminnescellerna SM110 och SM111, samt specialdataregisterna i SD110 och SD111.<br>(Endast i Q00(J) och Q01) | | A |

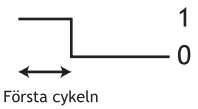
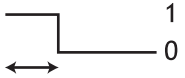
| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SM113 | Overrunfel | Från: Inget fel<br>Till: Fel | Ettställs när ett overrunfel (för mycket data) inträffar medan kommunikation sker.<br>(Endast i Q00(J) och Q01) | S<br>(Fel) |
| SM114 | Paritetsfel | Från: Inget fel<br>Till: Fel | Ettställs när ett paritetsfel inträffar medan kommunikation sker.<br>(Endast i Q00(J) och Q01) | S<br>(Fel) |
| SM115 | Framingfel | Från: Inget fel<br>Till: Fel | Ettställs när ett framingfel inträffar medan kommunikation sker.<br>(Endast i Q00(J) och Q01) | S<br>(Fel) |
| SM202 | LED frånslagskommando | Från > Till: LED från | Vid ändring från Från till Till släcks LED för respektive bit i SD202.<br>(Ej i Q00(J) eller Q01) | A |
| SM203 | STOPP-kontakt | Från: Ej i STOPP-mode<br>Till:STOPP-mode | Minnescellen ettställs när nyckelomkopplaren vrids till stopp. | S<br>(Statusförändring) |
| SM204 | PAUSE-kontakt | Från: Ej i PAUSE-mode<br>Till: PAUSE-mode | Ettställd då PLC-systemet är i PAUSE mode. | |
| SM205 | STEP-RUN-kontakt | Från: Ej i STEP-RUN<br>Till: STEP-RUN | Ettställd vid STEP-RUN-status.<br>(Ej i Q00(J) eller Q01) | |
| SM206 | Tillåt PAUSE-mode | Från: PAUSE ej tillåten<br>Till: PAUSE tillåten | PAUSE-mode är tillåten om denna cell är Till. | A |
| | Tvångsställningsbegäran | Från: Tvångsställning en utförd<br>Till: Tvångsställning utförd. | Ettställs när tvångsställning utförts i programmeringsmjukvaran.<br>(Endast i Q00(J) och Q01) | S<br>(Begäran) |
| SM210 | Begäran om uppdatering av klockdata | Från: Ingen åtgärd<br>Till: Begäran | När minnescellen ettställs överförs klockdata från SD210-SD213 till realtidsklockan vid END. | A |
| SM211 | Klockdatafel | Från: Normal<br>Till: Fel | Minnescellen ettställs vid felaktig klockdata i register (SD210-SD213). | S<br>(Begäran) |

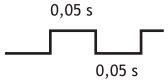
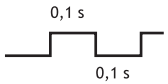
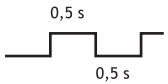
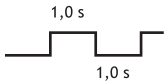
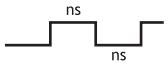
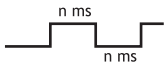
| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SM212 | Klockdata på display | Från: Ingen åtgärd<br>Till: Visning | Visar klockdata som månad, dag, timme, minut och sekund på displayen.<br>(Endast Q3A, Q4A, Q4AR). | A |
| SM213 | Klockdata läsbegäran | Från: Ingen åtgärd<br>Till: Begäran | När minnescellen är till läses klockdata till SD210-SD213 som BCD-värden. | |
| SM240 | Återställningsflagga CPU 1 | Från: Normal<br>Till: Centralenhet 1 har återställts | Ettställs när centralenhetern har återställts eller tagits bort från bottenplattan. Övriga centralenheter i ett multiprocessor-system ges återsällningsstatus.<br>(Endast MELSEC Q med multiprocessorstöd) | S<br>(Statusförändring) |
| SM241 | Återställningsflagga CPU 2 | Från: Normal<br>Till: Centralenhet 2 har återställts | | |
| SM242 | Återställningsflagga CPU 3 | Från: Normal<br>Till: Centralenhet 3 har återställts | | |
| SM243 | Återställningsflagga CPU 4 | Från: Normal<br>Till: Centralenhet 4 har återställts | | |
| SM244 | Felflagga CPU 1 | Från: Normal<br>Till: Centralenhet 1 har stoppats pga fel | Ettställs när ett fel har inträffat som stoppar bearbetningen i centralenheten.<br>Nollställs när centralenheten bearbetar programmet normalt, eller när ett fel inträffas som inte orsakar centralenheten att stoppa bearbetningen.<br>(Endast MELSEC Q med multiprocessorstöd) | |
| SM245 | Felflagga CPU 2 | Från: Normal<br>Till: Centralenhet 2 har stoppats pga fel | | |
| SM246 | Felflagga CPU 3 | Från: Normal<br>Till: Centralenhet 3 har stoppats pga fel | | |
| SM247 | Felflagga CPU 4 | Från: Normal<br>Till: Centralenhet 4 har stoppats pga fel | | |
| SM250 | I/O Läsning | Från: Normal<br>Till: Läsning | När denna minnescell ettställs skrivs den högsta I/O-adressen till SD250.<br>(Ej Q00(J), Q01) | A |
| SM251 | I/O-modul aktivering | Från: Inget utbyte<br>Till: Ubyte | Efter att huvudadress till I/O-modulen som skall bytas ut angetts i SD251, kan utbyte ske när minnescellen är Till. Endast en modul i taget kan bytas ut. För att byta ut en I/O-modul i RUN, sätts minnescellen Till med programmet eller programmeringsverktyget. Växla inte status RUN/STOPP förrän utbytet är klart.<br>(Endast QnA). | S<br>(END) |

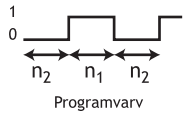
| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SM252 | Utbyte av I/O-modul | Från: Utbyte förbjudet<br>Till: Utbyte tillåtet | Ettställd när I/O-utbyte är tillåtet (Endast QnA). | S |
| SM254 | Kommando för att uppdatera alla stationer | Från: Uppdatera nya stationer<br>Till: Uppdatera alla stationer | Användbar för deluppdatering<br>Ange om uppdatering skall ske av nya stationer eller alla stationer.<br>(Endast Q, ej Q00(J, Q01) | A<br>(END) |
| SM255 | MELSECNET/10 information om modul 1 | Från: Nätverket i drift<br>Till: Nätverket i standby | Ettställd när nätverket är i standby.<br>(Ej Q00(J), Q01) | S<br>(Initiering) |
| SM256 | | Från: Läser<br>Till: Läser inte | Indikerar om läsning ska ske från länkmodulen vid uppdatering från länken till CPU (B,W etc).<br>(Ej Q00(J), Q01) | A |
| SM257 | | Från: Skriver<br>Till: Skriver inte | Indikerar om skrivning ska ske från länkmodulen vid uppdatering från länken till CPU (B,W etc).<br>(Ej Q00(J), Q01) | |
| SM260 | MELSECNET/10 information om modul 2 | Från: Nätverket i drift<br>Till: Nätverket i standby | Ettställd när nätverket är i standby.<br>(Ej Q00(J), Q01) | S<br>(Initiering) |
| SM261 | | Från: Läser<br>Till: Läser inte | Indikerar om läsning ska ske från länkmodulen vid uppdatering från länken till CPU (B,W etc).<br>(Ej Q00(J), Q01) | A |
| SM262 | | Från: Skriver<br>Till: Skriver inte | Indikerar om skrivning ska ske från länkmodulen vid uppdatering från länken till CPU (B,W etc).<br>(Ej Q00(J), Q01) | |
| SM265 | MELSECNET/10 information om modul 3 | Från: Nätverket i drift<br>Till: Nätverket i standby | Ettställd när nätverket är i standby.<br>(Ej Q00(J), Q01) | S<br>(Initiering) |
| SM266 | | Från: Läser<br>Till: Läser inte | Indikerar om läsning ska ske från länkmodulen vid uppdatering från länken till CPU (B,W etc).<br>(Ej Q00(J), Q01) | A |
| SM267 | | Från: Skriver<br>Till: Skriver inte | Indikerar om skrivning ska ske från länkmodulen vid uppdatering från länken till CPU (B,W etc).<br>(Ej Q00(J), Q01) | |

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SM270 | MELSECNET/10 information om modul 4 | Från: Nätverket i drift<br>Till: Nätverket i standby | Ettställd när nätverket är i standby.<br>(Ej Q00(J), Q01) | S<br>(Initiering) |
| SM271 | | Från: Läser<br>Till: Läser inte | Indikerar om läsning ska ske från länkmodulen vid uppdatering från länken till CPU (B,W etc).<br>(Ej Q00(J), Q01) | A |
| SM272 | | Från: Skriver<br>Till: Skriver inte | Indikerar om skrivning ska ske från länkmodulen vid uppdatering från länken till CPU (B,W etc).<br>(Ej Q00(J), Q01) | |
| SM280 | CC-Link-fel | Från: Normal<br>Till: Fel | Ettställs när ett CC-Link-fel uppstår i någon ansluten QJ61BT11 eller A(1S)J61BT11. Förblir ettställd även efter normal kommunikation återfåtts.<br>(Ej Q00J, Q00, Q01) | S<br>(Fel) |
| SM315 | Fördröjd cykeltid | Från: Utan fördröjning<br>Till: Fördröj | Vid ettställd aktiveras tidsfördröjning specificerad i SD315 för kommunikation mot andra enheter.<br>(Endast i Q00(J), Q01) | A |
| SM320 | Kontroll om SFC-program existerar | Från: Existerar ej<br>Till: Existerar | Till om SFC programmet är korrekt registrerat.<br>(Ej Q00J, Q00, Q01) | S<br>(Initiering) |
| SM321 | Start/Stopp SFC program | Från: SFC program Stopp<br>Till: SFC program Start | Initial värde sätts till samma värde som SM320. | S/A<br>(Initiering) |
| SM322 | SFC program start status | Från:Initial start<br>Till:Återstart | Initialvärdet ettställs eller nollställs beroende på parametrarna. | |
| SM323 | Aktivering av kontinuerlig övergång av hela block | Från: Kontinuerlig övergång används ej<br>Till: Kontinuerlig övergång används | Vid nollställning utförs övergången för ett steg/scan för alla block.<br>Vid ettställd sker övergången kontinuerligt för alla blocken på ett scanvarv. | A |
| SM324 | Kontinuerlig övergång blockeringsflagga | Från: När övergång exekveras<br>Till: När ingen övergång exekveras | Normalt ettställd när den kontinuerliga övergången inte används. | S |
| SM325[1)] | Utgångsmode vid blockstopp | Från: Från<br>Till: Bevara status | När block stoppar, väljer det aktiva steget hur utgångarna skall bete sig. Alla utgångar stängs av om minnescellen är Från. Utgångarnas status bevaras då minnescellen är ettställd. | |

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SM326 | Nollställning av SFC-variabler | Från: Nollställer variabler<br>Till: Behåller variabelstatus | Bestämmer hur variabelstatus skall vara då en stoppad centralenhet går i RUN efter ändringar i SFC-programmet. | A |
| SM327 | Utgångsstatus vid END-bearbetning | Från: Från<br>Till: Behåller status | Bestämmer hur utgångarna skall uppföra sig då ett steg är till när END-instruktionen bearbetas. | A |
| SM328 | Återställ bearbetningsmode när sista steget aktiveras | Från: Återställning utförs.<br>Till: Återställning utförs ej. | Bestämmer om sekvensstegen skall återställas när sista steget aktiveras.<br>När minnescellen är Från nollställs alla aktiva steg i blocket.<br>När minnescellen är Till fortsätter blockbearbetningen som vanligt. | A |
| SM330 | Arbetssätt för låghastighetsbearbetning | Från: Asynkron bearbetning<br>Till: Synkron bearbetning | Synkron bearbetning: Arbetssätt där låghastighetsprogram alltid påbörjas på ett nytt programvarv även om det finns tid över för bearbetning.<br>Asynkron bearbetning: Arbetssätt där låghastighetsprogram bearbetas under överbliven tid.<br>(Ej Q00(J), Q01) | A<br>(END) |
| SM331 | Normal SFC-program bearbetningsstatus | Från: Ingen bearbetning<br>Till: Bearbetas | Indikerar om det normala SFC-programmet bearbetas eller ej.<br>Används vid förregling av SFC-program.<br>(Endast Q, serienummer 04122 och senare) | S<br>(Statusförändring) |
| SM332 | Bearbetningsövervakning av SFC-program status | Från: Ingen bearbetning<br>Till: Bearbetas | Indikerar om bearbetningsövervakningen av SFC-programmet bearbetas eller ej.<br>Används vid förregling av SFC-program.<br>(Endast Q, serienummer 04122 och senare) | |
| SM390 | Access bearbetingsflagga | Till då en förfrågan till en intelligent funktionsmodul är avslutad | När minnescellen är Till indikerar detta att de intelligenta funktionsmodulerna har blivit avfrågade.<br>Flaggan kan användas som indikering på slutförd operation.<br>(Ej Q00(J), Q01) | S<br>(Statusförändring) |

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SM400 | Normalt ettställd | Till: ——————<br>Från: | Denna minnescell kan användas när något skall utföras kontinuerligt från första programcykeln, som t ex nollställning eller initiering. Behåller sin status oberoende av nyckelomkopplarens läge. | S<br>(Varje END) |
| SM401 | Normalt nollställd | Till:<br>Från: —————— | Behåller sin status oberoende av nyckelomkopplarens läge. | |
| SM402 | Ettställd första cykeln |  | Minnescellen är Till under ett programvarv efter STOPP ⇒ RUN. | |
| SM403 | Nollställd första cykeln | | Minnescellen är Från under ett programvarv efter STOPP ⇒ RUN. | |
| SM404 | Ettställd första cykeln (låghastighetsprogram) |  | Minnescellen är Till under ett programvarv efter STOPP ⇒ RUN.<br>(Ej Q00J, Q00, Q01 | |
| SM405 | Nollställd första cykeln (låghastighetsprogram) | | Minnescellen är Från under ett programvarv efter STOPP ⇒ RUN.<br>(Ej Q00J, Q00, Q01 | |
| SM409 | 0,01 s | | Minnescellen växlar mellan 1 och 0 med periodtiden 0,01s.<br>(Endast Q, ej Q00J, Q00, Q01 | |

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SM410 | 0,1 s |  | Minnesceller som växlar mellan 1 och 0 med bestämda periodtider 0.1 s, 0.2 s, 1.0 s, 2.0 s.<br><br>Observera att dessa minnesceller ändrar status direkt när tiden löpt ut, dvs status kan ändras under programbearbetningen. När matningsspänningen slås till eller när reset utförs, startar pulståget med 0. | S<br>(Statusförändring) |
| SM411 | 0,2 s |  | | |
| SM412 | 1 s |  | | |
| SM413 | 2,0 s |  | | |
| SM414 | 2n s |  | Växlar mellan 1 och 0 med antalet sekunder angivet i SD414. | S<br>(Statusförändring) |
| SM415 | 2n ms |  | Växlar mellan 1 och 0 med antalet millisekunder angivet i SD415.<br>(Endast Q, ej Q00J, Q00, Q01) | |

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SM420 | Pulståg nr 0 |  | Minnesceller som tillsammans med instruktionen DUTY kan växla mellan 0 och 1. När matningsspänningen slås till eller när reset utförs, startar pulståget med status 0. Tiderna för halvperioderna anger antalet programvarv (n1: 1 och n2: 0). | S<br>(Varje END) |
| SM421 | Pulståg nr 1 | | | |
| SM422 | Pulståg nr 2 | | | |
| SM423 | Pulståg nr 3 | | | |
| SM424 | Pulståg nr 4 | | | |
| SM430 | Pulståg nr 5 | | | |
| SM431 | Pulståg nr 6 | | | |
| SM432 | Pulståg nr 7 | | För användning med SM420-SM424 låghastighetsprogram.<br>(För Q00J-, Q00-, Q01-program med fast cykeltid) | |
| SM433 | Pulståg nr 8 | | | |
| SM434 | Pulståg nr 9 | | | |
| SM510 | Låghastighetsprogram exekveringsflagga | Från: Slutförd eller ej exekverad<br>Till: Exekvering på väg | Ettställs när låghastighetprogram exekveras.<br>(Ej Q00J, Q00, Q01) | S<br>(Varje END) |
| SM551 | Modul serviceintervall | Från: Ignorerad<br>Till: Läs | När den ettställs läses intervallet som ställts in (SD550) till SD551 och SD552.<br>(Ej Q00J, Q00, Q01) | A |
| SM580 | Program för I/O-uppdatering | Från: Ej uppdaterad<br>Till: Uppdaterad | När minnescellen ettställs utförs I/O-uppdatering då första programdelen utförts sedan bearbetas nästa programdel.<br>När ett PLC-program och ett sekvensprogram skall bearbetas utförs I/O-uppdatering mellan PLC-programmet och sekvensprogrammet.<br>(Endast Q, serienummer 04122 eller senare) | A |

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SM600 | Minneskort A statusflagga | Från:Kan ej användas<br>Till:Kan användas | Ettställs när minneskort A kan användas.<br>(Ej Q00J, Q00, Q01) | S<br>(Initiering) |
| SM601 | Minneskort skyddsflagga | Från:Inget skydd<br>Till:Skydd | Ettställs när minneskort A är skyddat.<br>(Ej Q00J, Q00, Q01) | |
| SM602 | Flagga för Drive 1 | Från: Drive 1 saknas<br>Till: Drive 1 finns | Ettställs när drive 1 (RAM area i kort 1) finns.<br>(Ej Q00J, Q00, Q01) | |
| SM603 | Flagga för Drive 2 | Från: Drive 2 saknas<br>Till: Drive 2 finns | Ettställd när drive 2 (ROM area i kort 1) finns.<br>(Ej Q00J, Q00, Q01) | |
| SM604 | Användarflagga för minneskort A | Från: Används inte<br>Till: Används | Ettställs när minneskort A används.<br>(Ej Q00J, Q00, Q01) | |
| SM605 | Ta bort/sätt in-flagga för minneskort A | Från:Ta bort/sätt in tilllåtet<br>Till:Ta bort/sätt in ej tillåtet | Ettställs när minneskort A inte kan sättas in eller tas bort.<br>(Ej Q00J, Q00, Q01) | A |
| SM609 | Ta bort/sätt in-flagga för minneskort | Från:Ta bort/sätt in tilllåtet<br>Till:Ta bort/sätt in ej tillåtet | Ettställs av användaren för att tillåta borttagning och isättning av minneskort.<br>Nollställs av systemet efter kortet tagits bort.<br>Minnescellen kan endast användas när SM604 och SM605 är Från.<br>(Endast Q00(J), Q01) | A/S |
| SM620 | Statusflagga för minneskort B | Från: Kan ej användas<br>Till: Kan användas | Q: Alltid till.<br>QnA: Till när minneskort B kan användas. | S<br>(Initiering) |
| SM621 | Skyddsflagga för minneskort B | Från: Inget skydd<br>Till: Skydd | Q: Alltid till.<br>QnA: Till när minneskortet B är skyddat. | |
| SM622 | Flagga för Drive 3 | Från: Drive 3 saknas<br>Till: Drive 3 finns | Q: Alltid till.<br>QnA: Till när drive 3 (RAM area i kort 2) finns. | |
| SM623 | Flagga för Drive 4 | Från: Drive 4 saknas<br>Till: Drive 4 finns | Q: Alltid till.<br>QnA: Till när drive 4 (ROM area) finns. | |
| SM624 | Minneskort B användarflagga | Från: Används Inte<br>Till: Används | Ettställs när minneskort B används. (Endast QnA) | S<br>(Initiering) |

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SM625 | Minneskort B ta bort/sätt in flagga | Från:Ta bort/sätt in tillåtet<br>Till: Ta bort/sätt in ej tillåtet | Ettställs när minneskort B inte kan sättas in eller tas bort.<br>(Endast QnA) | A |
| SM640 | Användning av filregister | Från: Används inte<br>Till: Används | Ettställs när filregister används. | S<br>(Statusförändring) |
| SM650 | Benämningsanvändning | Från: Används inte<br>Till: Används | Ettställs när benämningsfilen används.<br>(Ej Q00J, Q00, Q01) | |
| SM660 | Boot-funktion | Från: Intern minnes-bearbetning<br>Till: Bootfunktionen | Ettställd så länge boot-funktionen pågår. | |
| SM672 | Minneskort A filregister adressområde | Från: I adressområdet<br>Till: Utanför adressområdet | Ettställs när adressen ligger utanför adressområdet i minneskort A.<br>Återställs av användaren.<br>(Ej Q00J, Q00, Q01) | S/A |
| SM673 | Minneskort B filregister adrssområde | Från: I adressområdet<br>Till: Utanför adressområdet | Ettställs när adressen ligger utanför adressområdet i minneskort B.<br>Återställs av användaren.<br>(Endast QnA) | S/A |
| SM700 | Carry-flagga | Från: Carry nollställd<br>Till: Carry ettställd | Carry-flaggan används i applikationsinstruktioner | S<br>(Instruktion) |
| SM701 | Antal utgångar tecken-urval | Från: 16 tecken utgångar<br>Till: Utgångar fram till Null | När SM701 är nollställd matas 16 tecken ASCII-kod till utgång-arna.<br>När SM701 är ettställd matas tecken till utgångar tills Null (00H) påträffas.<br>(Ej Q00J, Q00, Q01) | A |
| SM702 | Sökmetod | Från: Sök nästa<br>Till: 2-part sök | Bestäm metod för sökinstruktion<br>Data måste arrangeras för 2-partsökning. | |
| SM703 | Sökordning | Från: Stigande ordning<br>Till: Fallande ordning | Stigande eller fallande ordning. | S<br>(Instruktion) |
| SM704 | Blockjämförelse | Från: Inget stämmer<br>Till: Allt stämmer | Ettställs när all data stämmer i BKCMP instruktionen. | |

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SM707 | Flyttalsbearbetningssätt | Från: Hastighetsorienterat<br>Till: Noggranhetsorienterat | När SM707 är från, bearbetas flyttalsinstruktioner så snabbt som möjligt. När SM707 är till bearbetas flyttalsinstruktioner med så stor noggrannhet som möjligt.<br>(Endast Q4AR) | A |
| SM710 | CHK instruktionen prioritetsflagga | Från: Villkorsprioritet<br>Till: Mönsterprioritet | Behåller sin ursprungliga status då den nollställs.<br>CHK instruktionen prioritetfunktioner uppdateras när minnescellen är ettställd.<br>(Ej Q00(J), Q01) | S<br>(Instruktion) |
| SM714 | Kommunikationsbegäran BUSY-signal MININET | Från: Kommunikationsbegäran till fjärrstyrd modul aktiverad<br>Till: Kommunikationsbegäran till fjärrstyrd modul avaktiverad | Minnescellen används för att avgöra om kommunikationsbegäran till fjärrstyrda moduler anslutna till en AJ71PT32-S3 kan utföras eller ej.<br>(Endast QnA) | |
| SM715 | Interrupt flagga | Från: Under DI flagga<br>Till: Under EI flagga | Ettställd när EI-instruktionen utörs. | |
| SM720 | Läsning av benämningar klar | Från: Läsning ej klar<br>Till: Läsning klar | Går till under ett cykelvarv när COMRD- eller PRC-instruktionerna är slutförda.<br>(Endast Q, ej Q00J, Q00, Q01) | S<br>(Statusförändring) |
| SM721 | Filåtkomst | Från: Filen ledig<br>Till: Filen upptagen | Går till när en fil påverkas av instruktionerna S.FWRITE, S.FREAD, COMRD, PRC eller LEDC.<br>(Endast Q) | |
| SM722 | BIN/DBIN koppla från felflagga | Från: Felflagga inkopplad<br>Till: Felflagga urkopplad | Ettställs av användaren om felkod ej skall returneras när instruktionerna BIN/DBIN bearbetas.<br>(Endast Q) | A |
| SM730 | Kommunikationsbegäran BUSY-signal CC-Link | Från: Kommunikationsbegäran med CC-Link-modul aktiverad<br>Till: Kommunikationsbegäran med CC-Link-modul avaktiverad | Minnescellen används för att avgöra om en kommunikationsbegäran till stationer anslutna till en CC-Link-modul kan utföras eller ej.<br>(Endast QnA) | S<br>(Instruktion) |

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SM736 | Exekveringsflagga för PKEY-instruktionen | Från: Instruktionen ej utförd<br>Till: Instruktionen utförs | Ettställd när PKEY-instruktionen exekveras. Nollställs när CR matas in eller när inmatad teckensträng blir 32 tecken lång.<br>(Ej Q00J, Q00, Q01) | S<br>(Instruktion) |
| SM737 | Tangentbordsmottagningsflagga för PKEY-instruktionen | Från: Tangentbordsmottagning tillåten<br>Till: Tangentbordsmottagning ej tillåten | Ettställs när inmatningen sker på tangentbordet.<br>Nollställs när inmatade tecken lagrats i CPU.<br>(Ej Q00J, Q00, Q01) | |
| SM738 | Mottagningsflagga för MSG-instruktionen | Från: Instruktionen inte utförd<br>Till: Instruktionen utförs | Ettställs när MSG-instruktionen utförs.<br>(Ej Q00J, Q00, Q01) | |
| SM774 | PID<br>"complete algorithm" | Från: Anpassa<br>Till: Anpassa inte | Tvinga anpassning av börvärdet till ärvärdet eller inte i manuellt mode. | A |
| SM775 | Val av länkuppdatering under COM instruktionen | Från: Utför länkuppdatering<br>Till: Ingen länkuppdatering | Val av länkuppdatering under COM-instruktionen. | |
| SM776 | Tillåt/tillåt ej lokala variabler vid CALL-anrop | Från: Lokala variabler kan användas<br>Till: Lokala variabler kan ej användas | Ettställd minnescell innebär att lokala variabler kan användas i subrutiner som anropats med CALL-instruktionen.<br>(Ej Q00J, Q00, Q01 och endast med GX Developer) | A<br>(Statusförändring) |
| SM777 | Tillåt/tillåt ej lokala variabler vid interruptprogram | Från: Lokala variabler kan användas<br>Till: Lokala variabler kan ej användas | Ettställd minnescell innebär att lokala variabler kan användas i interruptprogram.<br>(Ej Q00J, Q00, Q01 och endast med GX Developer) | A<br>(Statusförändring) |
| SM780 | Exekverbara CC-Link-instruktioner | Från: CC-Link-instruktioner kan bearbetas<br>Till: CC-Link-instruktioner kan ej bearbetas | Minnescellen ettställs när antalet samtidigt bearbetade CC-Link-instruktioner når 32.<br>Nollställs när antalet går under 32. | A<br>(Statusförändring) |
| SM794 | PID<br>"incomplete algorithm" | Från: Anpassa<br>Till: Anpassa inte | Tvinga anpassning av börvärdet till ärvärdet eller inte i manuellt mode. | A |
| SM800 | Förberedelse för Trace | Från: Ej förberedd<br>Till: Klar | Ettställs när förberedelserna för Trace är klara.<br>(Endast Q, ej Q00J, Q00, Q01) | S<br>(Statusförändring) |
| SM800 | Sampling trace start | | Ettställs när Sampling trace är redo att starta.<br>(Endast QnA(S)) | |

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SM801 | Trace start | Från: Väntar<br>Till: Start | Trace startar när minnescellen ettställs. Trace pausar när minnescellen nollställs. Alla relaterade SM nollställs.<br>(Endast Q, ej Q00J, Q00, Q01) | A |
| SM801 | Sampling trace start | | Sampling trace startas när minnescellen ettställs. Sampling trace pausar när minnescellen nollställs. Alla relaterade SM nollställs.<br>(Endast QnA(S)) | |
| SM802 | Trace bearbetas | Från: Väntar<br>Till: Start | Ettställs när Trace bearbetas.<br>(Endast Q, ej Q00J, Q00, Q01) | S<br>(Statusförändring) |
| SM802 | Sampling trace bearbetas | | Ettställs när Sampling trace bearbetas.<br>(Endast QnA(S)) | |
| SM803 | Trace trigger | Från > Till: Start | Trace trigger startas när denna minnescell ettställs (positiv flank), Samma funktion som instruktionen TRACE).<br>(Endast Q, ej Q00J, Q00, Q01) | A |
| SM803 | Sampling trace trigger | | Sampling trace trigger startas när denna minnescell ettställs (positiv flank), Samma funktion som instruktionen STRA).<br>(Endast QnA(S)) | |
| SM804 | Trace har utförts | Från: Trigger har ej aktiverats<br>Till: Trigger har aktiverats | Ettställs när Trace trigger har aktiverats.<br>(Endast Q, ej Q00J, Q00, Q01) | S<br>(Statusförändring) |
| SM804 | Sampling trace har utförts | | Ettställs när Sampling trace trigger har aktiverats.<br>(Endast QnA(S)) | |
| SM805 | Trace avslutat | Från: Ej avslutat<br>Till: Avslutat | Ettställs när Trace har avslutats.<br>(Endast Q, ej Q00J, Q00, Q01) | |
| SM805 | Sampling trace avslutat | | Ettställs när Sampling trace har avslutats.<br>(Endast QnA(S)) | |
| SM806 | Förberedelse för Status latch | Från: Ej redo<br>Till: Redo | Ettställs när Status latch är redo.<br>(Endast QnA(S)) | S |
| SM807 | Status latch-kommando | Från > Till: Exekvera latch | Exekverar Status latch-kommandot när minnescellen ettställs.<br>(Endast QnA(S)) | A |

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SM808 | Status latch slutförd | Från: Ej slutförd<br>Till: Slutförd | Ettställs när latch är slutförd.<br>(Endast QnA(S)) | S |
| SM809 | Nollställ<br>Status latch | Från > Till: Nollställ | Nollställ så nästa Status latch kan aktiveras.<br>(Endast QnA(S)) | A |
| SM810 | Förberedelse för Program trace | Från: Ej redo<br>Till: Redo | Ettställs när Program trace är redo.<br>(Endast QnA(S)) | S |
| SM811 | Start av Program trace | Från: Väntar<br>Till: Start | Program trace startar när minnescellen ettställs.<br>(Endast QnA(S)) | A |
| SM812 | Program trace utförs | Från: Väntar<br>Till: Start | Ettställs när Program trace utförs.<br>(Endast QnA(S)) | S |
| SM813 | Program trace trigger | Från > Till: Start | Program trace trigger startas när denna minnescell ettställs (positiv flank), Samma funktion som instruktionen PTRA).<br>(Endast QnA(S)) | A |
| SM814 | Program trace har utförts | Från: Trigger har ej aktiverats<br>Till: Trigger har aktiverats | Ettställs när Program trace trigger har aktiverats.<br>(Endast QnA(S)) | S<br>(Statusförändring) |
| SM815 | Program trace slutförd | Från: Ej slutförd<br>Till: Slutförd | Ettställs när Program trace har slutförts.<br>(Endast QnA(S)) | |
| SM820 | Förberedelse för Step trace | Från: Ej redo<br>Till: Redo | Ettställs när Step trace är redo.<br>(Ej Q00J, Q00, Q01) | |
| SM821 | Start av Step trace | Från: Väntar<br>Till: Start | Ettställs när Step trace har startat.<br>(Endast QnA(S)) | |
| SM822 | Step trace utförs | Från: Väntar<br>Till: Start | Ettställs när Step trace exekveras.<br>(Endast QnA(S)) | |

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SM823 | Step trace trigger har utförts | Från: Ingen trigger har aktiverats<br>Till: En trigger har aktiverats | Ettställs när ett block i aktiverat Step trace har aktiverats.<br>(Endast QnA(S)) | S<br>(Statusförändring) |
| SM824 | Step trace efter trigger | Från: Alla trigger har ej aktiverats<br>Till: Alla trigger har aktiverats | Ettställs när alla block i det exekverade Step trace är triggade.<br>(Endast QnA(S)) | |
| SM825 | Step trace slutfört | Från: Ej slutförd<br>TIll: Slutförd | Ettställs när Step trace slutförts.<br>(Endast QnA(S)) | |
| SM826 | Trace-fel | Från: Normalt<br>Till: Fel | Ettställs om ett fel uppstår när Trace bearbetas.<br>(Endast Q, ej Q00J, Q00, Q01) | |
| SM826 | Sampling trace-fel | | Ettställs om ett fel uppstår när Sampling trace bearbetas.<br>(Endast QnA(S)) | |
| SM827 | Status latch-fel | | Ettställs om ett fel uppstår när Status latch bearbetas.<br>(Endast QnA(S)) | |
| SM828 | Program trace-fel | | Ettställs om ett fel uppstår när Program trace bearbetas.<br>(Endast QnA(S)) | |
| SM900 | Spänningsavbrottsfil | Från: Ingen<br>Till Spänningsavbrott | Ettställs om filen är tillgänglig vid spänningsavbrott<br>(Endast QnA(S)) | |
| SM910 | Registreringsflagga för RKEY-instruktionen | Från: Inget registrerat från tangentbord<br>Till: Signal registrerad från tangentbord | Ettställs när signal registreras från tangentbordet, nollställs när inget registreras.<br>(Ej Q00J, Q00, Q01) | S<br>(Instruktion) |

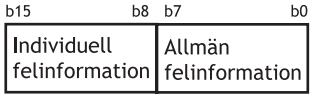
# 11 Specialdataregister

Interna specialregister är bestämda till PLC-typen och kan inte användas som andra register i PLC-programmet. Däremot kan systemet ettställa och nollställa minnescellerna för att kontrollera PLC-systemet.

| | | |
|---|---|---|
| **Adress** | Specialregistrens adress | |
| **Namn** | Specialregistrens namn | |
| **Betydelse** | Specialregistrens betydelse | |
| **Beskrivning** | Detaljerad beskrivning av specialregistren | |
| **Aktiveras av** | **Beskriver vem som aktiverar specialregistren** | |
| | Systemet | S |
| | Aktiveras av användaren (via PLC-programmet eller programmeringsverktyget) | A |
| | Aktiveras av systemet eller användaren | S/A |
| | **Ytterligare beskrivning om specialregistren sätts av systemet** | |
| | Aktiveras efter begäran från användaren | Begäran |
| | Aktiveras under instruktionsbearbetning | Instruktion |
| | Aktiveras efter att ett fel inträffat | Fel |
| | Aktiveras efter END-bearbetningen | END |
| | Aktiveras under initieringen | Initiering |
| | Aktiveras efter en statusförändring | Förändring |

# 11.1 Tabell specialdataregister

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SD0 | Diagnostikfel | Diagnostisk felkod | Felkod för diagnostikfel lagras som BIN-data. | S<br>(Fel) |
| SD1 | Tidpunkt för diagnosfel | Tidpunkt då diagnostidfel inträffade | År och månad<br>b15 b8 b7 b0<br>År (00-99) \| Månad (01-12)<br>Exempel: Februari 2001<br>H0102 | S<br>(Fel) |
| SD2 | | | Dag och timme<br>b15 b8 b7 b0<br>Dag (01-31) \| Timme (00-23)<br>Exempel: 10.00 den 25:e<br>H2510 | |
| SD3 | | | Minut och sekund<br>b15 b8 b7 b0<br>Minut (00-59) \| Sekund (00-59)<br>Exempel: 35 minuter och 30 sek<br>H3530 | |

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SD4 | Information om felkategori | Informationskod för felkategori | Kategorikod allmän: SD5 - SD15<br>individuell: SD16 - SD26<br>  <br><br>Allmän kategorikod:<br>0:Inget fel<br>1:Modulnummer/PLC-nummer/Bottenplattenummer<br>2:Filnamn/Drivenamn<br>3:Tid (värde)<br>4:Programfel<br><br>Individuell kategorikod :<br>0:Inget fel<br>1:(Öppen)<br>2:Filnamn/Drivenamn<br>3:Tid (värde)<br>4:Programfel<br>5:Parameternummer<br>6:Felminnesadress<br>7:Kontrollinstruktion (CHK) med detta nummer fungerar ej | S<br>(Fel) |

<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD5-SD15</td><td>Allmän felinformation</td><td>Allmän felinformation</td><td>Allmän information motsvarande felkoderna (SD0)<br>(1) Stations-/Moduladress<br><br><table><tr><th>Adress</th><th>Betydelse</th></tr><tr><td>SD5</td><td>Stations/Moduladress</td></tr><tr><td>SD6</td><td>I/O-adress</td></tr><tr><td>SD7-SD15</td><td>Lediga</td></tr></table><br>(2) Filnamn/Drivenamn<br><br><table><tr><th>Adress</th><th>Betydelse</th></tr><tr><td>SD5</td><td>Drive</td></tr><tr><td>SD6</td><td rowspan="4">Filnamn*<br>(ASCII 8 tecken)</td></tr><tr><td>SD7</td></tr><tr><td>SD8</td></tr><tr><td>SD9</td></tr><tr><td>SD10</td><td rowspan="2">Extention<br>(ASCII 3 tecken)</td></tr><tr><td>SD11</td></tr><tr><td>SD12</td><td rowspan="4">Lediga</td></tr><tr><td>SD13</td></tr><tr><td>SD14</td></tr><tr><td>SD15</td></tr></table><br>*Exempel: Filnamn = ABCDEFGH.IJK<br><br><table><tr><td>b15</td><td>b0</td></tr><tr><td>B</td><td>A</td></tr><tr><td>D</td><td>C</td></tr><tr><td>F</td><td>E</td></tr><tr><td>H</td><td>G</td></tr><tr><td>I</td><td>.</td></tr><tr><td>K</td><td>J</td></tr></table></td><td>S<br>(Fel)</td></tr>
</table>

<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD5-SD15</td><td>Allmän felinformation</td><td>Allmän felinformation</td><td>(Forts.)<br>(3) Tid (värde)<br><br><table><tr><th>Adress</th><th>Betydelse</th></tr><tr><td>SD5</td><td>Tid: 1 µs steg (0-999 µs)</td></tr><tr><td>SD6</td><td>Tid: 1 ms steg (0-65 535 ms)</td></tr><tr><td>SD7-SD15</td><td>Lediga</td></tr></table><br>(4) Programfel<br><br><table><tr><th>Adress</th><th>Betydelse</th></tr><tr><td>SD5</td><td rowspan="4">Filnamn</td></tr><tr><td>SD6</td></tr><tr><td>SD7</td></tr><tr><td>SD8</td></tr><tr><td>SD9</td><td rowspan="2">Extention</td></tr><tr><td>SD10</td></tr><tr><td>SD11</td><td>Bitmönster *)</td></tr><tr><td>SD12</td><td>Blocknummer</td></tr><tr><td>SD13</td><td>Steg-/övergångsnummer</td></tr><tr><td>SD14</td><td>Sekvensstegnummer (L)</td></tr><tr><td>SD15</td><td>Sekvensstegnummer (H)</td></tr></table><br>*) Innehåll i bitmönster<br><br><table><tr><td>15</td><td>14 till 4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td colspan="2">Används ej</td><td>*3</td><td>*2</td><td>*1</td></tr></table><br>Felinformation om:<br>*1 SFC-block finns (1)/finns inte (0)<br>*2 SFC-steg finns (1)/finns inte (0)<br>*3 SFC-övergång finns (1)/finns inte (0)</td><td>S<br>(Fel)</td></tr>
</table>

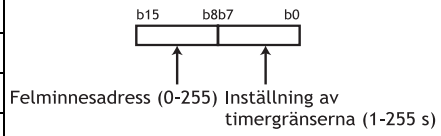
<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD16-SD26</td><td>Individuell felinformation</td><td>Individuell felinformation</td><td>Individuell information motsvarande felkoderna (SD0)<br>(1) Stations-/Moduladress<br><br><table><tr><th>Adress</th><th>Betydelse</th></tr><tr><td>SD16</td><td>Stations/Moduladress</td></tr><tr><td>SD17</td><td>I/O-adress</td></tr><tr><td>SD18-SD26</td><td>Lediga</td></tr></table><br>(2) Filnamn/Drivenamn<br><br><table><tr><th>Adress</th><th>Betydelse</th></tr><tr><td>SD16</td><td>Drive</td></tr><tr><td>SD17</td><td rowspan="4">Filnamn*<br>(ASCII 8 tecken)</td></tr><tr><td>SD18</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td></tr><tr><td>SD21</td><td rowspan="2">Extention<br>(ASCII 3 tecken)</td></tr><tr><td>SD22</td></tr><tr><td>SD23</td><td rowspan="4">Lediga</td></tr><tr><td>SD24</td></tr><tr><td>SD25</td></tr><tr><td>SD26</td></tr></table><br>*Exempel: Filnamn = ABCDEFGH.IJK<br><br>b15 b0<br><table><tr><td>B</td><td>A</td></tr><tr><td>D</td><td>C</td></tr><tr><td>F</td><td>E</td></tr><tr><td>H</td><td>G</td></tr><tr><td>I</td><td>.</td></tr><tr><td>K</td><td>J</td></tr></table></td><td>S<br>(Fel)</td></tr>
</table>

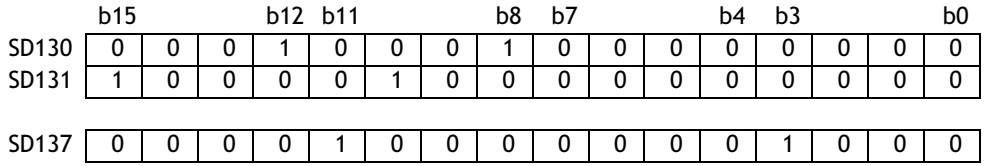
<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD16-SD26</td><td>Individuell felinformation</td><td>Individuell felinformation</td><td>(Forts.)<br>(3) Tid (värde)<br><br><table><tr><th>Adress</th><th>Betydelse</th></tr><tr><td>SD16</td><td>Tid: 1 µs steg (0-999 µs)</td></tr><tr><td>SD17</td><td>Tid: 1 ms steg (0-65 535 ms)</td></tr><tr><td>SD18-SD26</td><td>Lediga</td></tr></table><br>(4) Programfel<br><br><table><tr><th>Adress</th><th>Betydelse</th></tr><tr><td>SD16</td><td rowspan="4">Filnamn</td></tr><tr><td>SD17</td></tr><tr><td>SD18</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td><td rowspan="2">Extention</td></tr><tr><td>SD21</td></tr><tr><td>SD22</td><td>Bitmönster *)</td></tr><tr><td>SD23</td><td>Blocknummer</td></tr><tr><td>SD24</td><td>Steg-/övergångsnummer</td></tr><tr><td>SD25</td><td>Sekvensstegnummer (L)</td></tr><tr><td>SD26</td><td>Sekvensstegnummer (H)</td></tr></table><br>*) Innehåll i bitmönster<br><br><table><tr><td>15 14 till 4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>Används ej</td><td>*3</td><td>*2</td><td>*1</td></tr></table><br>Felinformation om:<br>*1 SFC-block finns (1)/finns inte (0)<br>*2 SFC-steg finns (1)/finns inte (0)<br>*3 SFC-övergång finns (1)/finns inte (0)</td><td>S<br>(Fel)</td></tr>
</table>

<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD16-SD26</td><td>Individuell felinformation</td><td>Individuell felinformation</td><td>(Forts.)<br>(5) Parameternummer<br><br><table><tr><th>Adress</th><th>Betydelse</th></tr><tr><td>SD16</td><td>Parameternummer</td></tr><tr><td>SD17-SD26</td><td>Lediga</td></tr></table><br>(6) Felminnesadress/adress till felaktig kontrollrestriktion<br><br><table><tr><th>Adress</th><th>Betydelse</th></tr><tr><td>SD16</td><td>Nummer</td></tr><tr><td>SD17-SD26</td><td>Lediga</td></tr></table></td><td>S<br>(Fel)</td></tr>
<tr><td>SD50</td><td>Felåterställning</td><td>Felkod som skall återställas</td><td>Felkoden som skall återställas skrivs här, se även SM50.</td><td>A</td></tr>
<tr><td>SD51</td><td>Låg batterispänning</td><td>Bitmönster som visar var låg batterispänning har uppstått</td><td>Alla korresponderande bitar ettställs när spänningen är låg. De nollställs inte när spänningsnivån återställs.<br><br><table><tr><th></th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr><tr><td>0</td><td>*1</td><td>*2</td><td>*3</td><td>*4</td><td>*5</td></tr></table><br>*1: Fel på minneskort B<br>*2: Larm på minneskort B<br>*3: Fel på minneskort A<br>*4: Larm på minneskort A<br>*5: Fel på CPU</td><td>S<br>(Fel)</td></tr>
<tr><td>SD52</td><td>Låg batterispänning</td><td>Bitmönster som visar var låg batterispänning har uppstått</td><td>Samma funktion som SD51 men bitarna nollställs när spänningen återgår till den normala.</td><td></td></tr>
<tr><td>SD53</td><td>Låg matningsspänning</td><td>Antal gånger låg matningsspänning uppstått.</td><td>Summan ökar med 1 varje gång växelspänningen understiger 80 % av tillåtet värde.</td><td></td></tr>
</table>

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SD54 | MINI link fel | | (1) Den relevanta stationsbiten ettställs när någon av de installerade MINI(-S3) X(n+0)/X(n+20), X(n+6)/X(n+26), X(n+7)/X(n+27)/X(n+8)/X(n+28) blir ettställda.<br>(2) Ettställs när kommunikationen bryts mellan den installerade MINI(-S3) och centralenheten.<br><br>b15 .. b8 b7 .. b0<br>Modul 8 .. Modul 1 Modul 8 .. Modul 1<br>Information finns (2) Information finns (1)<br><br>(Endast QnA) | S<br>(Fel) |
| SD60 | Säkringsbrott | Moduladress | Startadress på den modul som har säkringsbrott. | S<br>(Fel) |
| SD61 | Verifieringsfel I/O-modul | Moduladress | Startadress på den modul som har verifieringsfel. | S<br>(Fel) |
| SD62 | Felminnesadress | Felminnesadress | Första felminnesadressen lagras här | S<br>(Instruktion) |
| SD63 | Antal fel-minnen | Adresser på felminnen | Antal aktiverade felminnen | |

<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD64-SD79</td><td>Tabell över felminnen</td><td>Antal fel, adresser</td><td>
När SD64-SD79 aktiveras med OUT F eller<br>
SET F, visas aktiverad F-cell i SD64-SD79 i den ordning som de aktiveras. F-celler raderas från SD64-SD79 av RST F och innehållet i registerna skiftas till förgående. När 16 fel har registrerats kommer efterföljande fel inte att lagras i SD64-SD79.
<table>
<tr><th></th><th>SET F50</th><th>SET F25</th><th>SET F19</th><th>RST F25</th><th>SET F15</th><th>SET F70</th><th>SET F65</th><th>SET F38</th><th>LEDR</th></tr>
<tr><td></td><td>→</td><td>→</td><td>→</td><td>→</td><td>→</td><td>→</td><td>→</td><td>→</td><td>→</td><td>→</td></tr>
<tr><td>SD62</td><td>0</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>19</td></tr>
<tr><td>SD63</td><td>0</td><td>1</td><td>2</td><td>3</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>5</td></tr>
<tr><td>SD64</td><td>0</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>19</td></tr>
<tr><td>SD65</td><td>0</td><td>0</td><td>25</td><td>25</td><td>19</td><td>19</td><td>19</td><td>19</td><td>19</td><td>15</td></tr>
<tr><td>SD66</td><td>0</td><td>0</td><td>0</td><td>19</td><td>0</td><td>15</td><td>15</td><td>15</td><td>15</td><td>70</td></tr>
<tr><td>SD67</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>70</td><td>70</td><td>70</td><td>65</td></tr>
<tr><td>SD68</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>65</td><td>65</td><td>38</td></tr>
<tr><td>SD69</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>38</td><td>0</td></tr>
<tr><td>SD70</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>SD71</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>SD72</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>SD73</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>SD74</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>SD75</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>SD76</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>SD77</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>SD78</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>SD79</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</td><td>S<br>(Instruktion)</td></tr>
</table>

| Adress | Namn | Betydelse | Beskrivning | | Aktiveras av |
|---|---|---|---|---|---|
| SD80 | CHK-adress | CHK-adress | Felkoden som detekterats av CHK-instruktionen lagras som BCD-kod i SD80<br>(Ej Q00J, Q00, Q01) | | S<br>(Instruktion) |
| SD90 | Watchdog timer för övergångs villkor (aktiverad endast då SFC-program finns) | F-adress för timer-inställningsvärde | Samverkar med SM90 | F-adress som ettställs vid stegövergångs watchdogtimer och vid fel<br>  <br><br>Timern startas genom att SM90-SM99 aktiveras i SFC-händelsen. Om efterföljande övergångsvillkor ej uppfylls innan tiden gått ut kommer det valda felminnet att aktiveras.<br>(Ej Q00J, Q00, Q01) | A |
| SD91 | | | Samverkar med SM91 | | |
| SD92 | | | Samverkar med SM92 | | |
| SD93 | | | Samverkar med SM93 | | |
| SD94 | | | Samverkar med SM94 | | |
| SD95 | | | Samverkar med SM95 | | |
| SD96 | | | Samverkar med SM96 | | |
| SD97 | | | Samverkar med SM97 | | |
| SD98 | | | Samverkar med SM98 | | |
| SD99 | | | Samverkar med SM99 | | |
| SD100 | Överföringshastighet | Lagrar överförings-hastigheten som angets i kommuni-kationsinställning-arna | K96: 9,6 kbit/s<br>K384: 38,4 kbit/s<br>K1152: 115,2 kbit/s<br>(Endast Q00J, Q00, Q01) | K192: 19,2 kbit/s<br>K576: 57,6 kbit/s | S<br>(spänningstill-slag) |
| SD101 | Kommunikations-inställningar | Lagrar kommunika-tionsinställningar | Bit4 = Från: Utan checksumma<br>Bit4 = Till: Med checksumma<br><br>Bit5 = Från: Onlineändringar ej tillåtna<br>Bit5 = Till: Onlineändringar tillåtna<br>Övriga bitar har ingen funktion<br><br>(Endast Q00J, Q00, Q01) | | |
| SD102 | Väntetid | Lagrar väntetiden som angets i kom-munikationsinställ-ningarna | 0: Ingen väntetid<br>1 - F (Hex): Väntetid (enhet 10 ms)<br>Grundinställning: 0<br>(Endast Q00J, Q00, Q01) | | |

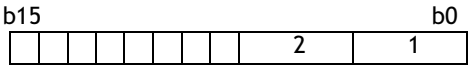
<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD105</td><td>Inställning av överföringshastighet (RS232) på kanal 1 (QJ71C24)</td><td>Lagrar inställd hastighet då GX Developer används för konfigurering</td><td>K3: 300 bit/s K192: 19200 bit/s<br>K6: 600 bit/s K384: 38400 bit/s<br>K24: 2400 bit/s K576: 57600 bit/s<br>K48: 4800 bit/s K1152: 115200 bit/s<br>K96: 9600 bit/s<br>(Endast Q, ej Q00J, Q00, Q01)</td><td>S</td></tr>
<tr><td>SD110</td><td>Sändresultat</td><td>Lagrar sändresultatet vid seriell kommunikation</td><td>Lagrar eventuell felkod vid seriell kommunikation.<br>(Endast Q00J, Q00, Q01)</td><td rowspan="2">S<br>(Fel)</td></tr>
<tr><td>SD111</td><td>Mottagningsresultat</td><td>Lagrar mottagningsresultatet vid seriell kommunikation</td><td>Lagrar eventuell felkod vid seriell kommunikation.<br>(Endast Q00J, Q00, Q01)</td></tr>
<tr><td>SD120</td><td>Adress på modul med fel på spänningsmatning</td><td>Adressnummer</td><td>Lagrar lägsta adressnummer på modul vars externa spänningsmatning är Från.<br>(Endast Q, ej Q00J, Q00, Q01)</td><td>S<br>(Fel)</td></tr>
<tr><td>SD130<br>SD131<br>SD132<br>SD133<br>SD134<br>SD135<br>SD136<br>SD137</td><td>Moduler med trasig säkring</td><td>Bitmönstret indikerar vilken modul som har en trasig säkring<br>Från: Säkring hel<br>Till: Säkring trasig</td><td>Antalet utgångsmoduler vars säkringar har gått sönder lagras i ett bitmönster (16-bitar)<br>Trasiga säkringar på fjärrstyrda I/O-moduler registreras också.<br>Bitarna återställs ej automatiskt när modulen byts ut. Flaggan måste nollställas genom en Reset<br>  <br>(Endast Q00J, Q00, Q01)</td><td>S<br>(Fel)</td></tr>
</table>

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SD150<br>SD151<br>SD152<br>SD153<br>SD154<br>SD155<br>SD156<br>SD157 | Verifieringsfel med I/O-modul | Bitmönstret indikerar vilken modul som har ett verifieringsfel<br>Från: Inget fel<br>Till: Fel | Vid spänningstillslag kontrolleras om modulinformationen stämmer överens med den som angetts i parametrarna. Om skillnader hittas indikeras detta genom att en bit ettställs i bitmönstret.<br><br>b15 b12 b11 b8 b7 b4 b3 b0<br>SD150 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0<br>SD151 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0<br><br>SD157 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0<br><br>(Endast Q00J, Q00, Q01) | S<br>(Fel) |

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SD200 | Switchstatus | Tillstånd på CPU-switch | Status på CPU switch lagras i följande form<br><br>b15 b8 b7 b4 b3 b0<br>Ledig \| 2) \| 1)<br>1): CPU-status (0): RUN<br>(1): STOP<br>2): Switch på minneskort Alltid nollställd<br>(Endast Q00J, Q00, Q01) | S<br>(END) |
| | | | Status på CPU switch lagras i följande form<br><br>b15 b8 b7 b4 b3 b0<br>3) \| Ledig \| 2) \| 1)<br>1): CPU-status (0): RUN<br>(1): STOP<br>(2): LCLR<br>2): Switch på minneskort Alltid nollställd<br>3): DIP-switch b8 till bC motsvarar SW1-SW5<br>Från: Nollställd (0)<br>Till: Ettställd (1)<br>bD-bF används ej<br>(Endast Q, ej Q00J, Q00, Q01) | |

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SD200 | Switchstatus | Tillstånd på CPU-switch | Status på CPU switch lagras i följande form<br><br>b15 b8 b7 b4 b3 b0<br>3) \| Ledig \| 2) \| 1)<br>1): CPU-status (0): RUN<br>(1): STOP<br>(2): LCLR<br>2): Switch på minneskort b4 motsvarar kort A<br>b5 motsvarar kort B<br>Från: Nollställd (0)<br>Till: Ettställd (1)<br>3): DIP-switch b8 till b15 motsvarar SW1-SW8<br>Från: Nollställd (0)<br>Till: Ettställd (1)<br><br>(Endast QnA) | S<br>(END) |

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SD201 | LED status | Tillstånd på CPU LED | Information om i vilket tillstånd lysdioderna på centralenheten har lagrats i följande bitmönster:<br>0: Från, 1: Till, 2: Blinkar<br>b15 b0<br>8 7 6 5 4 3 2 1<br>(1): RUN (5): Boot<br>(2): Fel (6): Minneskort A (QnA(S))<br>(3): Användare (7): Minneskort B (QnA)<br>(4): Batterilarm (8): 0: Från<br>1: Grön<br>2: Orange<br>(Endast QnA) | S<br>(Förändring) |
| | | | Information om i vilket tillstånd lysdioderna på centralenheten har lagrats i följande bitmönster:<br>b15 b0<br>8 7 6 5 4 3 2 1<br>(1): RUN (5): Boot<br>(2): Fel (6): Ledig<br>(3): Användare (7): Ledig<br>(4): Batterilarm (8): 0: Från<br>1: Grön<br>2: Orange<br>(Endast Q) | |
| SD202 | Kommando till LED | LED bitmönster som skall släckas | Lagrade bitmönster för lysdioder som skall släckas.<br>Släcks vid ettställd<br>Tänds vid nollställd<br>(Endast användar- och Boot-aktiverade)<br>(Endast QnA) | A |

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SD203 | Funktions-tillstånd hos CPU | Funktions-tillstånd hos CPU |  <br>(1) Funktionstillstånd hos CPU 0: RUN<br>1: STEP - RUN<br>2: STOP<br>3: PAUSE<br>(2) STOP/PAUSE-orsak 0: Nyckelomkopplare<br>1: Fjärrstyrd via ingång<br>2: Yttre kommunikation<br>3: Intern program-instruktion<br>4: Fel<br>Anmärkning: Endast första felet lagras. | S<br>(END) |
| SD206 | Device test execution type | 0: Test ännu ej exekverat<br>1: X device test<br>2: Y device test<br>3: X/Y device test | Värdet ändras beroende på hur Device Test mode exekveras i GX Developer. | S<br>(Begäran) |

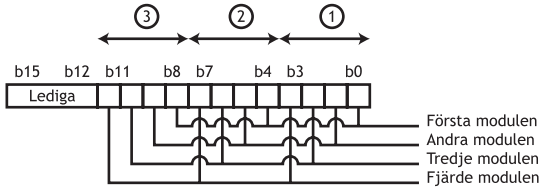
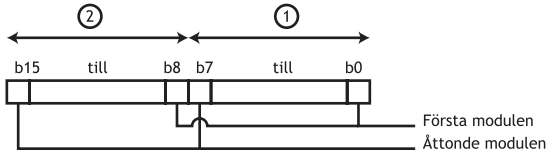
<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD207</td><td rowspan="3">LED display prioritetsval</td><td>Prioritet 1- 4</td><td rowspan="3">När fel uppstår blinkar displayen, beroende på prioritet. Inställningsområden för prioritet:<br><br><table><tr><td></td><td>b15-b12</td><td>b11-b8</td><td>b7-b4</td><td>b3-b0</td></tr><tr><td>SD207</td><td>Prioritet 4</td><td>Prioritet 3</td><td>Prioritet 2</td><td>Prioritet 1</td></tr><tr><td>SD208</td><td>Prioritet 8</td><td>Prioritet 7</td><td>Prioritet 6</td><td>Prioritet 5</td></tr><tr><td>SD209</td><td>_</td><td>_</td><td>Prioritet 10</td><td>Prioritet 9</td></tr></table><br>Grundinställning: SD207: H4321<br>SD208: H8765<br>SD209: H00A9<br>Parameterinställningar utan villkor. Inget visas på displayen om 0 är inställt. Även om 0 är inställt kommer lysdioderna indikera fel som stoppar programbearbetningen<br>(Ej Q00J, Q00, Q01)</td><td rowspan="3">A</td></tr>
<tr><td>SD208</td><td>Prioritet 5 - 8</td></tr>
<tr><td>SD209</td><td>Prioritet 9 - 10</td></tr>
<tr><td>SD210</td><td>Klockdata</td><td>Klockdata<br>(år, månad)</td><td>År och månad lagras i SD210 i BCD-kod enligt:<br><br><table><tr><td>b15</td><td>b0</td></tr><tr><td>År</td><td>Månad</td></tr></table><br>Exempel: Juli 1993 = H9307</td><td rowspan="3">S/A<br>(Begäran)</td></tr>
<tr><td>SD211</td><td>Klockdata</td><td>Klockdata<br>(dag, timme)</td><td>Dag och timme lagras i SD211 i BCD-kod enligt:<br><br><table><tr><td>b15</td><td>b0</td></tr><tr><td>Dag</td><td>Timme</td></tr></table><br>Exempel: Dag 31 kl 10 = H3110</td></tr>
<tr><td>SD212</td><td>Klockdata</td><td>Klockdata<br>(minut, sekund)</td><td>Minut och sekund lagras i SD212 i BCD-kod enligt:<br><br><table><tr><td>b15</td><td>b0</td></tr><tr><td>Minut</td><td>Sekund</td></tr></table><br>Exempel: 35 min 48 s = H3548</td></tr>
</table>

<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD213</td><td>Klockdata</td><td>Klockdata<br>(veckodag)</td><td>b15-b8 b7-b0<br><table><tr><td>1)</td><td>Se tabell</td></tr></table>1) QnA(S): Sätts till noll<br>MELSEC Q: Två högsta siffror representerar århundrade. Exempel: H2005, betyder fredag år 20xx.<br><table><tr><th colspan="2">Veckodag</th></tr><tr><td>0</td><td>Söndag</td></tr><tr><td>1</td><td>Måndag</td></tr><tr><td>2</td><td>Tisdag</td></tr><tr><td>3</td><td>Onsdag</td></tr><tr><td>4</td><td>Torsdag</td></tr><tr><td>5</td><td>Fredag</td></tr><tr><td>6</td><td>Lördag</td></tr></table></td><td>S/A<br>(Begäran)</td></tr>
<tr><td>SD220 - SD227</td><td>LED<br>displaydata</td><td>LED displaydata</td><td>LED - display/ASCII data lagras<br><table><tr><td></td><td>b15b8</td><td>b7b0</td></tr><tr><td>SD220</td><td>Tkn 15 från höger</td><td>Tkn 16 från höger</td></tr><tr><td>SD221</td><td>Tkn 13 från höger</td><td>Tkn 14 från höger</td></tr><tr><td>SD222</td><td>Tkn 11 från höger</td><td>Tkn 12 från höger</td></tr><tr><td>SD223</td><td>Tkn 9 från höger</td><td>Tkn 10 från höger</td></tr><tr><td>SD224</td><td>Tkn 7från höger</td><td>Tkn 8 från höger</td></tr><tr><td>SD225</td><td>Tkn 5 från höger</td><td>Tkn 6 från höger</td></tr><tr><td>SD226</td><td>Tkn 3 från höger</td><td>Tkn 4 från höger</td></tr><tr><td>SD227</td><td>Tkn 1 från höger</td><td>Tkn 2 från höger</td></tr></table></td><td>S<br>(Status-<br>ändring)</td></tr>
</table>

<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD240</td><td>Inställning av expansionsplattor</td><td>0: Automatiskt läge<br>1: Detaljerat läge</td><td>Till om man i detaljerat läge manuellt angett storleken på expansionsplattorna.<br>(Endast Q)</td><td>S</td></tr>
<tr><td>SD241</td><td>Antal expansionsplattor</td><td>Visar antalet anslutna expansionsplattor.</td><td>0: Endast grundplatta<br>1-7: Antal expansionsplattor<br>(Endast Q)</td><td>S<br>(Initiering)</td></tr>
<tr><td>SD242</td><td>Typ av bottenplattor</td><td>Visar vilken typ av expansionsplatta som är ansluten.</td><td>0: MELSEC AnS/QnAS-expansions-platta ansluten<br>1: MELSEC Q-expansionsplatta ansluten<br><br><table><tr><td>b15-b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr></table><br>Bit 0 betecknar grundplattan<br>Bit 1-7 betecknar expansionsplatta 1-7<br>(Endast Q, ej Q00J, Q00, Q01)<br><br>0: MELSEC AnS/QnAS-expansions-platta ansluten<br>1: MELSEC Q-expansionsplatta ansluten<br><br><table><tr><td>b15-b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr></table><br>Bit 0 betecknar grundplattan<br>Bit 1-4 betecknar expansionsplatta 1-4<br>(Endast Q00J, Q00, Q01)</td><td>S<br>(Initiering)</td></tr>
<tr><td>SD243<br>SD244</td><td>Antal platser på bottenplattorna</td><td>Antal platser på bottenplattorna</td><td><table><tr><td>b15-b12</td><td>b11-b8</td><td>b7-b4</td><td>b3-b0</td></tr><tr><td>Expansionsplatta 3</td><td>Expansionsplatta 2</td><td>Expansionsplatta 1</td><td>Grundplatta</td></tr><tr><td>Expansionsplatta 7</td><td>Expansionsplatta 6</td><td>Expansionsplatta 5</td><td>Expansionsplatta 4</td></tr></table><br>Varje area lagrar antalet platser på respektive platta<br><br>(Endast Q)</td><td>S<br>(Initiering)</td></tr>
</table>

<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD245<br>SD246</td><td>Antal bottenplattor<br>(Mounting status)</td><td>No. of base slots</td><td>
<table>
<tr><th></th><th>b15 b12</th><th>b11 b8</th><th>b7 b4</th><th>b3 b0</th></tr>
<tr><td>SD245</td><td>Expansion 3</td><td>Expansion 2</td><td>Expansion 1</td><td>Main</td></tr>
<tr><td>SD246</td><td colspan="3">Alltid 0</td><td>Expansion 4</td></tr>
</table>
Varje område i specialdataregistret lagrar antalet modulfyllda platser på respektive botten-platta enligt tabellen ovan.<br>(Endast Q00(J) och Q01 med serienummer 04122 eller senare)</td><td>S<br>(Initial)</td></tr>
<tr><td>SD250</td><td>Antal anslutna I/O</td><td>Antal anslutna I/O</td><td>När SM250 går Till lagras högsta adressen +1 i SD250 (lagras i BIN-värden).<br>Ex. Sista I/O-modulen har adress 75H som högsta adress. I SD250 lagras värdet 80H.</td><td>S<br>(END)</td></tr>
<tr><td>SD251</td><td>Adress<br>för modul- utbyte</td><td>Grundadress<br>modulutbyte</td><td>Lagrar de två övre tecknen i den första I/O moduladressen som har tagits bort eller ersatts under drift.<br>(Endast QnA)</td><td>A</td></tr>
<tr><td>SD253</td><td>RS422<br>kommunikationshas-tighet</td><td>RS422<br>kommunikations-hastighet</td><td>0: 9,6 kbit/s<br>1: 19,2 kbit/s<br>2: 38,4 kbit/s<br>(Endast QnA)</td><td>S<br>(Vid förändring)</td></tr>
</table>

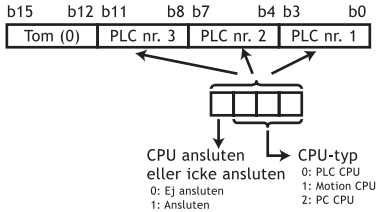
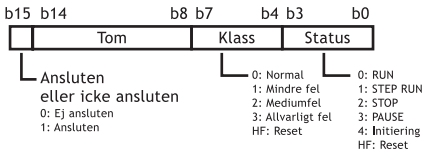
| Adress | Namn | Betydelse | Beskrivning | | Aktiveras av |
|---|---|---|---|---|---|
| SD254 | NET/10 information | Antal installerade moduler | Antal installerade NET/10 moduler. | | S (Initiering) |
| SD255 | | Information från den första modulen | I/O-adress | Den första adressen på NET/10 modulen. | |
| SD256 | | | Nätverksadress | Nätverksnumret till första NET/10 modulen. | |
| SD257 | | | Gruppadress | Gruppnumret till första NET/10 modulen. | |
| SD258 | | | Stationsadress | Stationsadressen på första NET/10 modulen. | |
| SD259 | | | Standby-information | I det fall när man använder standbystationer lagras standbystationens nr (1-4).<br>(Ej Q00J, Q00, Q01) | |
| SD260-SD264 | | Information för den andra modulen | Konfigurationen är samma som för den första modulen.<br>(Ej Q00J, Q00, Q01) | | |
| SD265-SD269 | | Information för den tredje modulen | Konfigurationen är samma som för den första modulen.<br>(Ej Q00J, Q00, Q01) | | |
| SD270-SD274 | NET/10 information | Information för den fjärde modulen | Konfigurationen är samma som för den första modulen.<br>(Ej Q00J, Q00, Q01) | | S (Initiering) |

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SD280 | CC-Link-fel | Status feldetektering | 1: När Xn0 på den anslutna CC-Link-modulen går TILL, ettställs modulens motsvarande bit i registret<br>2: När Xn1 eller XnF på den anslutna CC-Link-modulen går FRÅN, ettställs modulens motsvarande bit i registret.<br>3: Ettställs när kommunikation mellan den anslutna CC-Link-modulen och centralenheten ej kan genomföras.<br>  <br>Ovanstående modulnummer är i startadressordning. (Moduler som ansluts men ej parametrerats räknas inte in.) | S<br>(Fel) |
| | | | 1: När Xn0 på den anslutna CC-Link-modulen går TILL, ettställs modulens motsvarande bit i registret<br>2: När Xn1 eller XnF på den anslutna CC-Link-modulen går FRÅN, ettställs modulens motsvarande bit i registret.<br>  | |

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SD290 | Operandallokering (samma som parameter innehåll) | Antal X-adresser | Lagrar adresserna med X | S (Initiering) |
| SD291 | | Antal Y-adresser | Lagrar adresserna med Y | |
| SD292 | | Antal M-adresser | Lagrar adresserna med M | |
| SD293 | | Antal L-adresser | Lagrar adresserna med L | |
| SD294 | | Antal B-adresser | Lagrar adresserna med B | |
| SD295 | | Antal F-adresser | Lagrar adresserna med F | |
| SD296 | | Antal SB-adresser | Lagrar adresserna med SB | |
| SD297 | | Antal V-adresser | Lagrar adresserna med V | |
| SD298 | | Antal S-adresser | Lagrar adresserna med S | |
| SD299 | | Antal T-adresser | Lagrar adresserna med T | |
| SD300 | Operandallokering (samma som parameter innehåll) | Antal ST-adresser | Lagrar adresserna med ST | S (Initiering) |
| SD301 | | Antal C-adresser | Lagrar adresserna med C | |
| SD302 | | Antal D-adresser | Lagrar adresserna med D | |
| SD303 | | Antal W-adresser | Lagrar adresserna med W | |
| SD304 | | Antal SW-adresser | Lagrar adresserna med SW | |
| SD315 (Endast Q) | Reserverad tid för kommunikation | Reserverad tid för kommunikation | Här reserveras tid för kommunikation med GX Developer och andra programmeringsverktyg. Ju större värde som anges desto kortare blir svarstiden vid kommunikation med yttre enheter (GX Developer, seriell kommunikation).<br>Inställningsområde: 1 - 100 ms.<br>Om angivet värde är utanför områdesgränserna antas grundinställningen (0 ms).<br>Cykeltiden förlängs med det angivna värdet.<br>(Endast Q) | END |

| Adress | Namn | Betydelse | Beskrivning | | Aktiveras av |
|---|---|---|---|---|---|
| SD340 | Ethernetinformation MELSEC Q | Antal anslutna moduler | Anger antalet anslutna Ethernetmoduler. | | S (Initiering) |
| SD341 | | Information för första modulen | I/O-adress | Huvudadress för första anslutna Ethernet-modulen. | |
| SD342 | | | Nätverksnummer | Nätverksadress för första anslutna Ethernet-modulen. | |
| SD343 | | | Gruppnummer | Gruppnummer för första anslutna Ethernet-modulen. | |
| SD344 | | | Stationsnummer | Stationsnummer för första anslutna Ethernet-modulen. | |
| SD345-SD347 | | | Reserverade | Reserverade. (Ej Q00J, Q00, Q01) | |
| SD348-SD354 | | Information för andra modulen | Konfigurationen är samma som för den första modulen. (Ej Q00J, Q00, Q01) | | |
| SD355-SD361 | | Information för tredje modulen | Konfigurationen är samma som för den första modulen. (Ej Q00J, Q00, Q01) | | |
| SD362-SD368 | | Information för fjärde modulen | Konfigurationen är samma som för den första modulen. (Ej Q00J, Q00, Q01) | | |

<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th colspan="2">Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD340</td><td rowspan="10">Ethernetinformation MELSEC QnA(S)</td><td>Antal anslutna moduler</td><td colspan="2">Anger antalet anslutna Ethernet-moduler.</td><td rowspan="10">S<br>(Initiering)</td></tr>
<tr><td>SD341</td><td rowspan="6">Information för första modulen</td><td>I/O-adress</td><td>Huvudadress för första anslutna Ethernet-modulen.</td></tr>
<tr><td>SD342</td><td>Nätverksnummer</td><td>Nätverksadress för första anslutna Ethernet-modulen.</td></tr>
<tr><td>SD343</td><td>Gruppnummer</td><td>Gruppnummer för första anslutna Ethernet-modulen.</td></tr>
<tr><td>SD344</td><td>Stationsnummer</td><td>Stationsnummer för första anslutna Ethernet-modulen.</td></tr>
<tr><td>SD345-<br>SD346</td><td>IP-adress</td><td>IP-adress för första anslutna Ethernet-modulen.</td></tr>
<tr><td>SD347</td><td>Felkod</td><td>Felkoden för första anslutna Ethernet-modulen.</td></tr>
<tr><td>SD348-<br>SD354</td><td>Information för andra modulen</td><td colspan="2">Konfigurationen är samma som för den första modulen.</td></tr>
<tr><td>SD355-<br>SD361</td><td>Information för tredjemodulen</td><td colspan="2">Konfigurationen är samma som för den första modulen.</td></tr>
<tr><td>SD362-<br>SD368</td><td>Information för fjärde modulen</td><td colspan="2">Konfigurationen är samma som för den första modulen.</td></tr>
<tr><td>SD380</td><td rowspan="4">Ethernetinformation, status på mottagning (Endast QnA(S))</td><td>Information för första modulen</td><td colspan="2"><table><tr><td>b15-b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>Används ej</td><td>*1</td><td>*1</td><td>*1</td><td>*1</td><td>*1</td><td>*1</td><td>*1</td><td>*1</td></tr></table>*1) Bit 0-7 indikerar vilka portar som är öppna,<br>0: Porten är stängd.<br>1: Porten är öppen.</td><td rowspan="4">S<br>(Initiering)</td></tr>
<tr><td>SD381</td><td>Information för andra modulen</td><td colspan="2">Funktionen är samma som för den första modulen.</td></tr>
<tr><td>SD382</td><td>Information för tredje modulen</td><td colspan="2">Funktionen är samma som för den första modulen.</td></tr>
<tr><td>SD383</td><td>Information för fjärde modulen</td><td colspan="2">Funktionen är samma som för den första modulen.</td></tr>
</table>

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SD392 | Mjukvaruversion | Intern mjukvaru-version | Lagrar det interna mjukvaruversionsnumret i ASCII-kod. Versionsnumret lagras i den höga byten. (Den låga byten innehåller ingen information.) Om versionen är "A" lagras värdet "H41xx". **Observera! Det interna mjukvaruversionsnumret kan skilja sig från symbolen på framsidan av centralenheten.** (Endast QnA(S)) | S (Initiering) |
| SD393 | Systeminformation multiprocessor | Antal multiprocessorer | Antalet anslutna centralenheter i ett multiprocessorsystem lagras (1 till 3, inga anges också) | |
| SD394 | | Information om anslutna centralenheter | Registret innehåller information om hur många och av vilken typ de anslutna centralenheterna är.<br>  | |
| SD395 | | Adress på processor | I ett multipelt PLC-system lagras adressen för det fysiska PLC-systemet.<br>1: PLC nr 1<br>2: PLC nr 2<br>3: PLC nr 3<br>4: PLC nr 4<br>(Endast Q version B och senare, samt Q00, Q01 med serienummer 04122 eller senare) | |
| SD396 | | Status CPU nr 1 | Statusinformation om varje centralenhet lagras i respektive register.<br>  | S (END) |
| SD397 | | Status CPU nr 2 | | |
| SD398 | | Status CPU nr 3 | | |
| SD412 | Sekundräknare | Antal pulser gjorda i sekund-räknaren | Specialdataregistret ökas med 1 varje sekund. Värdet växlar från 0 till 32 767 till -32 768 till 0. | S (Statusändring) |
| SD414 | Räknare 2n sek. | Tidbas för 2n s | Värdet anger tidbasen för 2n sekundräknaren. Grundinställningen är 30 s. | A |

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SD415 | Räknare 2n ms | Tidbas för 2n ms | Värdet anger tidbasen för 2n ms räknaren. Grundinställningen är 30 ms.<br>(Endast Q, ej Q00J, Q00, Q01) | A |
| SD420 | Cykelvarvsräknare | Antal cykelvarv | Registret ökas med 1 varje programcykelvarv. Värdet växlar från 0 till 32 767 till - 32 768 till 0.<br>Programdelar som körs som initiering räknas ej med. | S<br>(END) |
| SD430 | Räknare för låghastighetscykelvarv | Antal cykelvarv för låghastighetsprogram | Registret ökas med 1 varje cykelvarv för låghastighetsprogram. Värdet växlar från 0 till 32 767 till - 32 768 till 0.<br>(Ej Q00J, Q00, Q01) | S<br>(END) |
| SD500 | Programadress | Program som exekveras | Adress för program som körs, lagrad som BIN-värde.<br>(Ej Q00J, Q00, Q01) | S<br>(Statusändring) |
| SD510 | Låghastighetsprogramadress | Låghastighetsprogram som exekveras | Adress för låghastighetsprogram som körs, lagrad som BIN-värde. Aktiverad när SM510 är ettställd.<br>(Ej Q00J, Q00, Q01) | S<br>(END) |
| SD520 | Aktuell scantid | Aktuell scantid i enheter om 1 ms | Lagrar aktuell scantid i enheter om 1 ms.<br>Område 0 - 65 535. | S<br>(END) |
| SD521 | | Aktuell scantid i enheter om 1 µs | Lagrar aktuell scantid i enheter om 1 µs.<br>Området 0 - 900.<br>Exempel:<br>En scantid på 23,6 ms kommer att lagras som:<br>D520=23<br>D521=600 | |
| SD522 | Scantid för program som körs Initialt | Scantid i enheter om 1 ms | Lagrar scantiden i enheter om 1 ms.<br>Område 0 - 65 535.<br>(Ej Q00J, Q00, Q01) | S<br>(END) |
| SD523 | | Scantid i enheter om 1 µs | Lagrar scantiden i enheter om 1 µs.<br>Område 0 - 900.<br>(Ej Q00J, Q00, Q01) | |
| SD524 | Kortaste scantid | Kortaste scantid i enheter om 1 ms | Lagrar kortaste scantid i enheter om 1 ms.<br>Område 0 - 65 535. | S<br>(END) |
| SD525 | | Minimum scantid i enheter om 1 µs | Lagrar kortaste scantid i enheter om 1 µs.<br>Område 0 - 900. | |

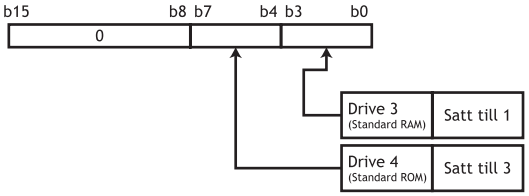
| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SD526 | Längsta scantid | Längsta scantid i enheter om 1 ms | Lagrar längsta scantid i enheter om 1 ms.<br>Område 0 - 65 535. | S<br>(END) |
| SD527 | | Längsta scantid i enheter om 1 µs | Lagrar längsta scantid i enheter om 1 µs.<br>Område 0 - 900. | |
| SD528 | Låghastighets exekvering, aktuell scantid | Aktuell scantid i enheter om 1 ms | Lagrar aktuell scantid för alla låghastighetsprogram i enheter om 1 ms.<br>Område 0 - 65 535.<br>(Ej Q00J, Q00, Q01) | S<br>(END) |
| SD529 | | Aktuell scantid i enheter om 1 µs | Lagrar aktuell scantid för alla låghastighetsprogram i enheter om 1 µs.<br>Område 0 - 900.<br>(Ej Q00J, Q00, Q01) | |
| SD532 | För låghastighets exekvering kortaste scantid | Kortaste scantid i enheter om 1 ms | Lagrar kortaste scantid för alla låghastighetsprogram i enheter om 1 ms (utom första scanvarvet).<br>Område 0 - 65 535.<br>(Ej Q00J, Q00, Q01) | S<br>(END) |
| SD533 | | Kortaste scantid i enheter om 1 µs | Lagrar kortaste scantid för alla låghastighetsprogram i enheter om 1 µs.<br>Område 0 - 900.<br>(Ej Q00J, Q00, Q01) | |
| SD534 | För låghastighets exekvering längtsta scantid | Längtsta scantid i enheter om 1 ms | Lagrar längsta scantid för alla låghastighetsprogram i enheter om 1 ms (utom första scanvarvet).<br>Område 0 - 65 535.<br>(Ej Q00J, Q00, Q01) | S<br>(END) |
| SD535 | | Längtsta scantid i enheter om 1 µs | Lagrar längsta scantid för alla låghastighetsprogram i enheter om 1 µs (utom första scanvarvet).<br>Område 0 - 900.<br>(Ej Q00J, Q00, Q01) | |
| SD540 | END bearbetningstid | END bearbetningstid i enheter om 1 ms | Lagrar tiden för ett komplett programvarv i enheter om 1 ms.<br>Område 0 - 65 535. | S<br>(END) |
| SD541 | | END bearbetningstid i enheter om 1 µs | Lagrar tiden för ett komplett programvarv i enheter om 1 µs.<br>Område i 0 - 900. | |

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SD542 | Konstant scantid, väntetid | Konstant scantid, väntetid i enheter om 1 ms | Lagrar väntetiden när en konstant scantid har ställts in i enheter om 1 ms.<br>Område 0 - 65 535. | S<br>(END) |
| SD543 | | Konstant scantid, väntetid i enheter om 1 µs | Lagrar väntetiden när en konstant scantid har ställts in i enheter om 1 µs.<br>Område 0 - 900. | |
| SD544 | Total exekveringstid för låghastighetsprogram under ett cykelvarv | Total exekveringstid för låghastighetsprogram i enheter om 1 ms under ett cykelvarv | Lagrar total exekveringstid för låghastighetsprogram i enheter om 1 ms.<br>Område 0 - 65 535.<br>Nollställs följande programvarv.<br>(Ej Q00J, Q00, Q01) | S<br>(END) |
| SD545 | | Total exekveringstid för låghastighetsprogram i enheter om 1 µs under ett cykelvarv | Lagrar total exekveringstid för låghastighetsprogram i enheter om 1 µs.<br>Område 0 - 900.<br>Nollställs följande programvarv.<br>(Ej Q00J, Q00, Q01) | |
| SD546 | Exekeringstid för låghastighetsprogram | Exekveringstiden för ett varv i låghastighetsprogram i enheter om 1 ms | Lagrar exekveringstiden för ett varv i låghastighetsprogram i enheter om 1 ms.<br>Område 0 - 65 535.<br>(Ej Q00J, Q00, Q01) | S<br>(END) |
| SD547 | | Exekveringstiden för ett varv i låghastighetsprogram i enheter om 1 µs | Lagrar exekveringstiden för ett varv i låghastighetsprogram i enheter om 1 µs.<br>Område 0 - 900.<br>(Ej Q00J, Q00, Q01) | |

<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD548</td><td rowspan="2">Scanprogram exekveringstid</td><td>Exekveringstiden för ett varv i scan-program i enheter om 1 ms</td><td>Lagrar exekveringstiden för ett varv i scanprogram i enheter om 1 ms.<br>Område 0 - 65 535.</td><td rowspan="2">S<br>(END)</td></tr>
<tr><td>SD549</td><td>Exekveringstiden för ett varv i scan-program i enheter om 1 µs</td><td>Lagrar exekveringstiden för ett varv i scanprogram i enheter om 1 µs.<br>Område 0 - 900.</td></tr>
<tr><td>SD550</td><td>Serviceintervall mät-modul</td><td>Moduladress</td><td>Ställer in adress till modul som skall mäta serviceintervallet.<br>(Ej Q00J, Q00, Q01)</td><td>A</td></tr>
<tr><td>SD551</td><td rowspan="2">Serviceintervall tid</td><td>Serviceintervall i 1 ms enheter</td><td>När SM551 ettställs lagras serviceintervallet i 1 ms enheter för modulen i SD550.<br>Område 0 - 65 535.<br>(Ej Q00J, Q00, Q01)</td><td rowspan="2">S<br>(Begäran)</td></tr>
<tr><td>SD552</td><td>Serviceintervall i 1 µs enheter</td><td>När SM551 ettställs lagras serviceintervallet i 1 µs enheter för modulen i SD550.<br>Område 0 - 900.<br>(Ej Q00J, Q00, Q01)</td></tr>
<tr><td rowspan="2">SD600</td><td>Minneskort A<br>MELSEC Q</td><td>Minneskort A</td><td>Anger vilken typ av minneskort som är installerat.<br><table><tr><th>Korttyp</th><th>Ettställd bit</th></tr><tr><td>SRAM</td><td>b0</td></tr><tr><td>ATA</td><td>b5</td></tr><tr><td>FLASH</td><td>b4 och b5</td></tr></table>(Ej Q00J, Q00, Q01)</td><td rowspan="2">S<br>(Initiering och då kortet tas ur)</td></tr>
<tr><td>Minneskort A<br>MELSEC QnA(S)</td><td>Minneskort A</td><td>Anger vilken typ av minneskort som är installerat på plats A.<br><table><tr><th>Korttyp</th><th>Ettställd bit</th></tr><tr><td>SRAM</td><td>b0</td></tr><tr><td>FLASH/ EEPROM</td><td>b0 och b5</td></tr></table></td></tr>
</table>

<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD602</td><td>Drive 1 RAM</td><td>Drive 1 kapacitet</td><td>Kapaciteten för Drive 1 lagras i 1 Kbyte enheter<br>(Ej Q00J, Q00, Q01)</td><td>S<br>(Initial)</td></tr>
<tr><td>SD603</td><td>Drive 2 ROM</td><td>Drive 2 kapacitet</td><td>Kapaciteten för Drive 2 lagras i 1 Kbyte enheter<br>(Ej Q00J, Q00, Q01)</td><td>S<br>(Initial)</td></tr>
<tr><td>SD604</td><td>Minneskort A användning</td><td>Minneskort A användning MELSEC Q</td><td>Informationen på minneskort A lagras som bitmönster i registret (Används om biten är Till)<br>Betydelsen visas nedan:<br><table><tr><th>Bit</th><th>Betydelse</th></tr><tr><td>b0</td><td>Boot-funktion (QBT)</td></tr><tr><td>b1</td><td>Parametrar (QPA)</td></tr><tr><td>b2</td><td>Benämningar (QCD)</td></tr><tr><td>b3</td><td>Initieringsvärde (QDI)</td></tr><tr><td>b4</td><td>Filregister R (QDR)</td></tr><tr><td>b5</td><td>Trace (QTS)</td></tr><tr><td>b6</td><td></td></tr><tr><td>b7</td><td></td></tr><tr><td>b8</td><td></td></tr><tr><td>b9</td><td>CPU felhistorik (QFD)</td></tr><tr><td>b10</td><td></td></tr><tr><td>b11</td><td>Lokal operand (QDL)</td></tr><tr><td>b12</td><td></td></tr><tr><td>b13</td><td></td></tr><tr><td>b14</td><td></td></tr><tr><td>b15</td><td></td></tr></table>(Ej Q00J, Q00, Q01)</td><td>S<br>(Statusändring)</td></tr>
</table>

<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD604</td><td>Minneskort A användning</td><td>Minneskort A användning MELSEC QnA(S)</td><td>Informationen på minneskort A lagras som bitmönster i registret (Används om biten är Till)<br>Betydelsen visas nedan:<br><br><table><tr><th>Bit</th><th>Betydelse</th></tr><tr><td>b0</td><td>Boot-funktion (QBT)</td></tr><tr><td>b1</td><td>Parametrar (QPA)</td></tr><tr><td>b2</td><td>Benämningar (QCD)</td></tr><tr><td>b3</td><td>Initieringsvärde (QDI)</td></tr><tr><td>b4</td><td>Filregister R (QDR)</td></tr><tr><td>b5</td><td>Trace (QTS)</td></tr><tr><td>b6</td><td>Status latch (QTL)</td></tr><tr><td>b7</td><td>Program trace (QTP)</td></tr><tr><td>b8</td><td>Simuleringsdata (QDS)</td></tr><tr><td>b9</td><td>CPU felhistorik (QFD)</td></tr><tr><td>b10</td><td>SFC trace (QTR)</td></tr><tr><td>b11</td><td>Lokal operand (QDL)</td></tr><tr><td>b12</td><td></td></tr><tr><td>b13</td><td></td></tr><tr><td>b14</td><td></td></tr><tr><td>b15</td><td></td></tr></table></td><td>S<br>(Statusändring)</td></tr>
</table>

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SD620 | Intern drive 3 och 4 MELSEC Q | Drive 3 och 4 | Anger vilken typ av minneskort som är installerat i Drive 3 och 4.<br>  | S<br>(Initiering och då kortet tas ur) |
| | Minneskort B MELSEC QnA | Minneskort B | Anger vilken typ av minneskort som är installerat på plats B.<br>**Korttyp** / **Ettställd bit**<br>SRAM: b0<br>SRAM/ EEPROM: b0 och b5 | S<br>(Initiering och då kortet tas ur) |
| SD622 | Drive 3 (RAM) kapacitet | Kapacitet för Drive 3 (Q) | Kapacitet i drive 3, i enheter om 1 Kbyte. | S<br>(IInitiering) |
| | | Kapacitet för Drive 3 (QnA) | Kapacitet i drive 3, i enheter om 1 Kbyte. | |
| SD623 | Drive 4 (ROM) kapacitet | Kapacitet för Drive 4 (Q) | Kapacitet i drive 4, i enheter om 1 Kbyte. | |
| | | Kapacitet för Drive 4 (QnA) | Kapacitet i drive 4, i enheter om 1 Kbyte. | |

<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD624</td><td>Användning av minneskort B</td><td>Användning av Drive 3</td><td>Användandet av Drive 3 bestämms av bit 4<br>Bit4 = Från: Drive 3 används ej<br>Bit4 = Till: Drive 3 används<br>(Endast Q00J, Q00, Q01)</td><td>S<br>(Statusändring)</td></tr>
<tr><td>SD624</td><td>Drive 3 och 4</td><td>Användning av Drive 3 och 4 MELSEC Q</td><td>Informationen på drive 3 och 4 lagras som bitmönster i registret (Används om biten är Till)<br>Betydelsen visas nedan:<br><table><tr><th>Bit</th><th>Betydelse</th></tr><tr><td>b0</td><td>Boot-funktion (QBT)</td></tr><tr><td>b1</td><td>Parametrar (QPA)</td></tr><tr><td>b2</td><td>Benämningar (QCD)</td></tr><tr><td>b3</td><td>Initieringsvärde (QDI)</td></tr><tr><td>b4</td><td>Filregister R (QDR)</td></tr><tr><td>b5</td><td>Trace (QTS)</td></tr><tr><td>b6</td><td></td></tr><tr><td>b7</td><td></td></tr><tr><td>b8</td><td></td></tr><tr><td>b9</td><td>CPU felhistorik (QFD)</td></tr><tr><td>b10</td><td></td></tr><tr><td>b11</td><td>Lokal operand (QDL)</td></tr><tr><td>b12</td><td></td></tr><tr><td>b13</td><td></td></tr><tr><td>b14</td><td></td></tr><tr><td>b15</td><td></td></tr></table>(Ej Q00J, Q00, Q01)</td><td></td></tr>
</table>

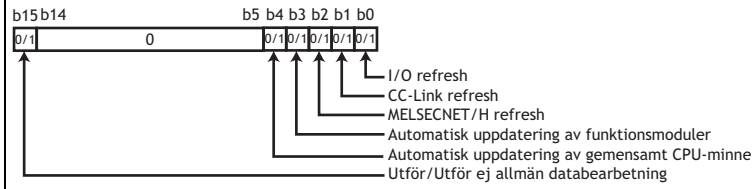
<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD624</td><td>Användning av minneskort B</td><td>Minneskort B användning<br>QnA</td><td>Informationen på minneskort B lagras som bitmönster i registret (Används om biten är Till)<br>Betydelsen visas nedan:<br><table><tr><th>Bit</th><th>Betydelse</th></tr><tr><td>b0</td><td>Boot-funktion (QBT)</td></tr><tr><td>b1</td><td>Parametrar (QPA)</td></tr><tr><td>b2</td><td>Benämningar (QCD)</td></tr><tr><td>b3</td><td>Initieringsvärde (QDI)</td></tr><tr><td>b4</td><td>Filregister R (QDR)</td></tr><tr><td>b5</td><td>Trace (QTS)</td></tr><tr><td>b6</td><td>Status latch (QTL)</td></tr><tr><td>b7</td><td>Program trace (QTP)</td></tr><tr><td>b8</td><td>Simuleringsdata (QDS)</td></tr><tr><td>b9</td><td>CPU felhistorik (QFD)</td></tr><tr><td>b10</td><td>SFC trace (QTR)</td></tr><tr><td>b11</td><td>Lokal operand (QDL)</td></tr><tr><td>b12</td><td></td></tr><tr><td>b13</td><td></td></tr><tr><td>b14</td><td></td></tr><tr><td>b15</td><td></td></tr></table></td><td></td></tr>
<tr><td>SD640</td><td>Filregister drive</td><td>Drivenummer</td><td>Lagrar drivenumret som används för filregistren</td><td>S<br>(Initiering)</td></tr>
</table>

<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD641-SD646</td><td>Filnamn, Filregister</td><td>Filnamn Filregister</td><td>Lagrar filregisterfilens namn (med filändelse) som valts i parametrarna eller med QDRSET instruktionen som ASCII kod<br><br>b15 b8 b7 b0<br><table><tr><td>SD641</td><td>Tecken 2 i filnamn</td><td>Tecken 1 i filnamn</td></tr><tr><td>SD642</td><td>Tecken 4 i filnamn</td><td>Tecken 3 i filnamn</td></tr><tr><td>SD643</td><td>Tecken 6 i filnamn</td><td>Tecken 5 i filnamn</td></tr><tr><td>SD644</td><td>Tecken 8 i filnamn</td><td>Tecken 7 i filnamn</td></tr><tr><td>SD645</td><td>Tecken 1 i filändelsen</td><td>2 $E_H$ (.)</td></tr><tr><td>SD646</td><td>Tecken 3 i filändelsen</td><td>Tecken 2 i filändelsen</td></tr></table></td><td>S<br>(Initiering)</td></tr>
<tr><td>SD647</td><td>Filregisterkapacitet</td><td>Filregisterkapacitet</td><td>Lagrar datakapaciteten i det aktuella filregistret i 1 K word enheter</td><td>S<br>(Statusändring)</td></tr>
<tr><td>SD648</td><td>Filregister, blocknummer</td><td>Filregister, blocknummer</td><td>Lagrar det aktuella blocknumret</td><td>S<br>(Statusändring)</td></tr>
<tr><td>SD650</td><td>Benämnings- drive</td><td>Benämnings- drive</td><td>Lagrar drivenummret för benämningarna som valts i parametrarna eller av QCDSET-instruktionen som ASCII kod.</td><td>S<br>(Statusändring)</td></tr>
<tr><td>SD651-SD656</td><td>Filnamn, benämningsfil</td><td>Filnamn, benämningsfil</td><td>Lagrar filnamnet (med filändelse) som valts i parametrarna eller med QCDSET-instruktionen som ASCII-kod<br><br>b15 b8 b7b0<br><table><tr><td>SD651</td><td>Tecken 2 i filnamn</td><td>Tecken 1 i filnamn</td></tr><tr><td>SD652</td><td>Tecken 4 i filnamn</td><td>Tecken 3 i filnamn</td></tr><tr><td>SD653</td><td>Tecken 6 i filnamn</td><td>Tecken 5 i filnamn</td></tr><tr><td>SD654</td><td>Tecken 8 i filnamn</td><td>Tecken 7 i filnamn</td></tr><tr><td>SD655</td><td>Tecken 1 i filändelsen</td><td>2 $E_H$ (.)</td></tr><tr><td>SD656</td><td>Tecken 3 i filändelsen</td><td>Tecken 2 i filändelsen</td></tr></table><br>(Ej Q00J, Q00, Q01)</td><td>S<br>(Statusändring)</td></tr>
</table>

<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD660</td><td rowspan="2">Bootfunktionsfil</td><td>Bootfunktions-drive-nummer</td><td>Lagrar drive-nummer där boot-filen (*.QBT) lagras<br>(Ej Q00J, Q00, Q01)</td><td>S<br>(Initiering)</td></tr>
<tr><td>SD661 - SD666</td><td>Namn på boot-filen</td><td>Lagrar filnamnet på boot-filen (*.QBT)<br><table><tr><th></th><th>b15 b8</th><th>b7 b0</th></tr><tr><td>SD661</td><td>Tecken 2 i filnamn</td><td>Tecken 1 i filnamn</td></tr><tr><td>SD662</td><td>Tecken 4 i filnamn</td><td>Tecken 3 i filnamn</td></tr><tr><td>SD663</td><td>Tecken 6 i filnamn</td><td>Tecken 5 i filnamn</td></tr><tr><td>SD664</td><td>Tecken 8 i filnamn</td><td>Tecken 7 i filnamn</td></tr><tr><td>SD665</td><td>Tecken 1 i filändelsen</td><td>2 $E_H$ (.)</td></tr><tr><td>SD666</td><td>Tecken 3 i filändelsen</td><td>Tecken 2 i filändelsen</td></tr></table>(Ej Q00J, Q00, Q01)</td><td>S<br>(Initiering)</td></tr>
<tr><td>SD705</td><td rowspan="2">Maskningsmönster</td><td rowspan="2">Maskningsmönster</td><td rowspan="2">Om SM705 ettställs under blockexekvering kan man med innehåller i SD705 (eller SD705 och SD706 vid dubbelordsoperationer) maska ut vilka block som skall exekveras.<br>(Ej Q00J, Q00, Q01)</td><td rowspan="2">A</td></tr>
<tr><td>SD706</td></tr>
<tr><td>SD714</td><td>Registreringsarea för antal lediga kommu-nikationsbegäran</td><td>0 till 32</td><td>Lagrar antalet lediga block i arean för kommunikationsbegäran för fjärrstyrda moduler anslutna via MELSECNET/MINI-S3.<br>(Endast QnA(S))</td><td>S<br>(Exekvering)</td></tr>
</table>

<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD715<br>SD716<br>SD717</td><td>IMASK-instruktion maskningsmönster</td><td>Maskningsmönster</td><td>Maskningsmönster för IMASK-instruktionen lagras på följande sätt:
<table>
<tr><th></th><th>SD715</th><th>SD716</th><th>SD717</th></tr>
<tr><td>b15</td><td>I15</td><td>I31</td><td>I47</td></tr>
<tr><td>b14</td><td>I14</td><td>I30</td><td>I46</td></tr>
<tr><td>b13</td><td>I13</td><td>I29</td><td>I45</td></tr>
<tr><td>b12</td><td>I12</td><td>I28</td><td>I44</td></tr>
<tr><td>b11</td><td>I11</td><td>I27</td><td>I43</td></tr>
<tr><td>b10</td><td>I10</td><td>I26</td><td>I42</td></tr>
<tr><td>b9</td><td>I9</td><td>I25</td><td>I41</td></tr>
<tr><td>b8</td><td>I8</td><td>I24</td><td>I40</td></tr>
<tr><td>b7</td><td>I7</td><td>I23</td><td>I39</td></tr>
<tr><td>b6</td><td>I6</td><td>I22</td><td>I38</td></tr>
<tr><td>b5</td><td>I5</td><td>I21</td><td>I37</td></tr>
<tr><td>b4</td><td>I4</td><td>I20</td><td>I36</td></tr>
<tr><td>b3</td><td>I3</td><td>I19</td><td>I35</td></tr>
<tr><td>b2</td><td>I2</td><td>I18</td><td>I34</td></tr>
<tr><td>b1</td><td>I1</td><td>I17</td><td>I33</td></tr>
<tr><td>b0</td><td>I0</td><td>I16</td><td>I32</td></tr>
</table></td><td>S<br>(Exekvering)</td></tr>
<tr><td>SD718<br>SD719</td><td>Ackumulator</td><td>Ackumulator</td><td>Används som ersättning för ackumulatorn i A-serien.</td><td>S/A</td></tr>
<tr><td>SD720</td><td>Tilldelning av programnummer för PLOAD-instruktionen</td><td>Tilldelning av programnummer för PLOAD-instruktionen</td><td>Lagrar programnumret för programmet som skall laddas med PLOAD-instruktionen.<br>Inställningsområde: 1 till 124.<br>(Endast Q)</td><td>A</td></tr>
<tr><td>SD730</td><td>Antal lediga areor för CC-Link-kommunikationsbegäran</td><td>0 - 32</td><td>Lagrar antalet lediga areor för kommunikationsbegäran med slavmoduler anslutna till A(1S)QBT61.<br>(Endast QnA)</td><td>S<br>(Exekvering)</td></tr>
<tr><td>SD736</td><td>PKEY-ingång</td><td>PKEY-ingång</td><td>SD736 lagrar temporärt tangentbordsdata från PKEY-instruktionen.<br>(Endast QnA(S))</td><td>S<br>(Exekvering)</td></tr>
</table>

<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD738-SD769</td><td>Meddelandelagring</td><td>Meddelandelagring</td><td>Lagrar meddelande av MSG-instruktionen enligt följande tabell:<br><br>
<table>
<tr><th></th><th>b15 b8</th><th>b7 b0</th><th></th><th>b15 b8</th><th>b7 b0</th></tr>
<tr><td>SD738</td><td>Tecken 2</td><td>Tecken 1</td><td>SD754</td><td>Tecken 34</td><td>Tecken 33</td></tr>
<tr><td>SD739</td><td>Tecken 4</td><td>Tecken 3</td><td>SD755</td><td>Tecken 36</td><td>Tecken 35</td></tr>
<tr><td>SD740</td><td>Tecken 6</td><td>Tecken 5</td><td>SD756</td><td>Tecken 38</td><td>Tecken 37</td></tr>
<tr><td>SD741</td><td>Tecken 8</td><td>Tecken 7</td><td>SD757</td><td>Tecken 40</td><td>Tecken 39</td></tr>
<tr><td>SD742</td><td>Tecken 10</td><td>Tecken 9</td><td>SD758</td><td>Tecken 42</td><td>Tecken 41</td></tr>
<tr><td>SD743</td><td>Tecken 12</td><td>Tecken 11</td><td>SD759</td><td>Tecken 44</td><td>Tecken 43</td></tr>
<tr><td>SD744</td><td>Tecken 14</td><td>Tecken 13</td><td>SD760</td><td>Tecken 46</td><td>Tecken 45</td></tr>
<tr><td>SD745</td><td>Tecken 16</td><td>Tecken 15</td><td>SD761</td><td>Tecken 48</td><td>Tecken 47</td></tr>
<tr><td>SD746</td><td>Tecken 18</td><td>Tecken 17</td><td>SD762</td><td>Tecken 50</td><td>Tecken 49</td></tr>
<tr><td>SD747</td><td>Tecken 20</td><td>Tecken 19</td><td>SD763</td><td>Tecken 52</td><td>Tecken 51</td></tr>
<tr><td>SD748</td><td>Tecken 22</td><td>Tecken 21</td><td>SD764</td><td>Tecken 54</td><td>Tecken 53</td></tr>
<tr><td>SD749</td><td>Tecken 24</td><td>Tecken 23</td><td>SD765</td><td>Tecken 56</td><td>Tecken5 5</td></tr>
<tr><td>SD750</td><td>Tecken 26</td><td>Tecken 25</td><td>SD766</td><td>Tecken 58</td><td>Tecken 57</td></tr>
<tr><td>SD751</td><td>Tecken 28</td><td>Tecken 27</td><td>SD767</td><td>Tecken 60</td><td>Tecken 59</td></tr>
<tr><td>SD752</td><td>Tecken 30</td><td>Tecken 29</td><td>SD768</td><td>Tecken 62</td><td>Tecken 61</td></tr>
<tr><td>SD753</td><td>Tecken 32</td><td>Tecken 31</td><td>SD769</td><td>Tecken 64</td><td>Tecken 63</td></tr>
</table>
(Ej Q00J, Q00, Q01)</td><td>S<br>(Exekvering)</td></tr>
<tr><td>SD774-SD775</td><td>PID-gränsinställning (complete algorithm)</td><td>PID-gränsinställning</td><td>Ange gränser för varje PID-loop enligt nedan<br><br>
<table>
<tr><th></th><th>b15</th><th></th><th>b1</th><th>b0</th></tr>
<tr><td>SD774</td><td>Loop16</td><td>till</td><td>Loop2</td><td>Loop1</td></tr>
<tr><td>SD775</td><td>Loop32</td><td>till</td><td>Loop18</td><td>Loop17</td></tr>
</table>
0: Gränsvärde angivet<br>1: Gränsvärde ej angivet<br>(Endast Q)</td><td>A</td></tr>
</table>

| Adress | Namn | Betydelse | Beskrivning | Aktiveras av |
|---|---|---|---|---|
| SD778 | Inställning av COM-instruktionen | b0-b14:<br>0: Uppdatera ej<br>1: Uppdatera<br>b15:<br>0: Allmän databearbetning utförd<br>1: Allmän databearbetning ej utförd | Registret bestämmer om data skall uppdateras eller ej när COM-instruktionen bearbetas.<br>Inställningarna i SD778 gäller endast när SM775 är ettställd.<br>  <br><br>(Endast Q, serienummer 04012 eller senare) | A |
| SD780 | Återstående antal samtidiga bearbetningar av dedicerade CC-Link-instruktioner | 0 till 32 | Registret innehåller återstående antal samtidiga bearbetningar av dedicerade CC-Link-instruktioner.<br>(Endast QnA) | A |

<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD781-SD793</td><td>IMASK-instruktion maskningsmönster</td><td>Maskningsmönster</td><td>Maskningsmönster för IMASK-instruktionen lagras på följande sätt:<br><table><tr><th></th><th>SD781</th><th>...</th><th>SD793</th></tr><tr><td>b15</td><td>I63</td><td>...</td><td>I255</td></tr><tr><td>b14</td><td>I62</td><td>...</td><td>I254</td></tr><tr><td>b13</td><td>I61</td><td>...</td><td>I253</td></tr><tr><td>b12</td><td>I60</td><td>...</td><td>I252</td></tr><tr><td>b11</td><td>I59</td><td>...</td><td>I251</td></tr><tr><td>b10</td><td>I58</td><td>...</td><td>I250</td></tr><tr><td>b9</td><td>I57</td><td>...</td><td>I249</td></tr><tr><td>b8</td><td>I56</td><td>...</td><td>I248</td></tr><tr><td>b7</td><td>I55</td><td>...</td><td>I247</td></tr><tr><td>b6</td><td>I54</td><td>...</td><td>I246</td></tr><tr><td>b5</td><td>I53</td><td>...</td><td>I245</td></tr><tr><td>b4</td><td>I52</td><td>...</td><td>I244</td></tr><tr><td>b3</td><td>I51</td><td>...</td><td>I243</td></tr><tr><td>b2</td><td>I50</td><td>...</td><td>I242</td></tr><tr><td>b1</td><td>I49</td><td>...</td><td>I241</td></tr><tr><td>b0</td><td>I48</td><td>...</td><td>I240</td></tr></table></td><td></td></tr>
<tr><td>SD794<br>SD795</td><td>PID-gränsinställning (incomplete algorithm)</td><td>PID-gränsinställning</td><td>Ange gränser för varje PID-loop enligt nedan<br><table><tr><th></th><th>b15</th><th></th><th>b1</th><th>b0</th></tr><tr><td>SD774</td><td>Loop16</td><td>till</td><td>Loop2</td><td>Loop1</td></tr><tr><td>SD775</td><td>Loop32</td><td>till</td><td>Loop18</td><td>Loop17</td></tr></table>0: Gränsvärde angivet<br>1: Gränsvärde ej angivet<br>(Endast Q, serienummer 05032 eller senare, samt Q00 och Q01, serienummer 04122 eller senare)</td><td>A</td></tr>
</table>

<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD806-SD811</td><td>Filnamn för Status latch</td><td>Filnamn för Status latch</td><td>Lagrar filnamnet (med filändelse) från en tidpunkt när statuslatch var utförd som ASCII kod<br>
<table>
<tr><th></th><th>b15 b8</th><th>b7 b0</th></tr>
<tr><td>SD806</td><td>Tecken 2 i filnamn</td><td>Tecken 1 i filnamn</td></tr>
<tr><td>SD807</td><td>Tecken 4 i filnamn</td><td>Tecken 3 i filnamn</td></tr>
<tr><td>SD808</td><td>Tecken 6 i filnamn</td><td>Tecken 5 i filnamn</td></tr>
<tr><td>SD809</td><td>Tecken 8 i filnamn</td><td>Tecken 7 i filnamn</td></tr>
<tr><td>SD810</td><td>Tecken 1 i filändelsen</td><td>2 $E_H$ (.)</td></tr>
<tr><td>SD811</td><td>Tecken 3 i filändelsen</td><td>Tecken 2 i filändelsen</td></tr>
</table>
(Endast QnA(S))</td><td>S<br>(Exekvering)</td></tr>
<tr><td>SD812-SD816</td><td>Statuslatch-steg</td><td>Statuslatchsteg</td><td>
<table>
<tr><td>SD812</td><td>Mönster*</td></tr>
<tr><td>SD813</td><td>Blocknummer</td></tr>
<tr><td>SD814</td><td>Stegnummer</td></tr>
<tr><td>SD815</td><td>Sekvensstegnummer (L)</td></tr>
<tr><td>SD816</td><td>Sekvensstegnummer (H)</td></tr>
</table>
*
<table>
<tr><td>b15-b3</td><td>b2</td><td>b1</td><td>b0</td></tr>
<tr><td>Alltid 0</td><td>*1</td><td>*2</td><td>*3</td></tr>
</table>
*1) 0: SFC-övergångsbenämning finns ej<br>
1: SFC-övergångsbenämning finns<br>
*2) 0: SFC-blockbenämning finns ej<br>
1: SFC-blockbenämning finns<br>
*3) 0: SFC-blockbenämning finns ej<br>
1: SFC-blockbenämning finns<br>
(Endast QnA(S))</td><td>S<br>(Exekvering)</td></tr>
<tr><td>SD900</td><td>Drive-nummer spänningsavbrott Endast QnA(S)</td><td>Drive-nummer spänningsavbrott</td><td>Lagrar drive-adressen till filen som var öppen vid spänningsavbrott<br>(Endast QnA(S))</td><td>S<br>(Status-ändring)</td></tr>
</table>

<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD901-SD906</td><td>Filnamnet under spänningsavbrott</td><td>Filnamnet under spänningsavbrott</td><td>Om en fil var öppen vid spänningsavbrott lagras filnamn (inkl. filändelse) i ASCII-kod enligt nedan.<br><table><tr><th></th><th>b15 b8</th><th>b7 b0</th></tr><tr><td>SD901</td><td>Tecken 2 i filnamn</td><td>Tecken 1 i filnamn</td></tr><tr><td>SD902</td><td>Tecken 4 i filnamn</td><td>Tecken 3 i filnamn</td></tr><tr><td>SD903</td><td>Tecken 6 i filnamn</td><td>Tecken 5 i filnamn</td></tr><tr><td>SD904</td><td>Tecken 8 i filnamn</td><td>Tecken 7 i filnamn</td></tr><tr><td>SD905</td><td>Tecken 1 i filändelsen</td><td>2 $E_H$ (.)</td></tr><tr><td>SD906</td><td>Tecken 3 i filändelsen</td><td>Tecken 2 i filändelsen</td></tr></table>(Endast QnA(S))</td><td>S<br>(Status-ändring)</td></tr>
<tr><td>SD910- SD925</td><td>RKEY-ingång</td><td>RKEY-ingång</td><td>Lagrad i sekvens PU teckenkod<br><table><tr><th></th><th>b15b8</th><th>b7b0</th></tr><tr><td>SD910</td><td>Tecken 2</td><td>Tecken 1</td></tr><tr><td>SD911</td><td>Tecken 4</td><td>Tecken 3</td></tr><tr><td>SD912</td><td>Tecken 6</td><td>Tecken 5</td></tr><tr><td>SD913</td><td>Tecken 8</td><td>Tecken 7</td></tr><tr><td>SD914</td><td>Tecken 10</td><td>Tecken 9</td></tr><tr><td>SD915</td><td>Tecken 12</td><td>Tecken 11</td></tr><tr><td>SD916</td><td>Tecken 14</td><td>Tecken 13</td></tr><tr><td>SD917</td><td>Tecken 16</td><td>Tecken 15</td></tr><tr><td>SD918</td><td>Tecken 18</td><td>Tecken 17</td></tr><tr><td>SD919</td><td>Tecken 20</td><td>Tecken 19</td></tr><tr><td>SD920</td><td>Tecken 22</td><td>Tecken 21</td></tr><tr><td>SD921</td><td>Tecken 24</td><td>Tecken 23</td></tr><tr><td>SD922</td><td>Tecken 26</td><td>Tecken 25</td></tr><tr><td>SD923</td><td>Tecken 28</td><td>Tecken 27</td></tr><tr><td>SD924</td><td>Tecken 30</td><td>Tecken 29</td></tr><tr><td>SD925</td><td>Tecken 32</td><td>Tecken 31</td></tr></table>(Endast QnA(S))</td><td>S<br>(Exekvering)</td></tr>
</table>

<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD1300-<br>SD1331</td><td>Säkringsbrott i modul</td><td>Bitmönster som visar moduler med säkringsavbrott</td><td>Av bitmönstret kan utläsas vilka moduler som har trasiga säkringar:<br><table><tr><th></th><th>SD1300</th><th>...</th><th>SD1331</th></tr><tr><td>b15</td><td>0</td><td>...</td><td>0</td></tr><tr><td>b14</td><td>0</td><td>...</td><td>1</td></tr><tr><td>b13</td><td>0</td><td>...</td><td>0</td></tr><tr><td>b12</td><td>0</td><td>...</td><td>0</td></tr><tr><td>b11</td><td>0</td><td>...</td><td>0</td></tr><tr><td>b10</td><td>0</td><td>...</td><td>0</td></tr><tr><td>b9</td><td>0</td><td>...</td><td>0</td></tr><tr><td>b8</td><td>0</td><td>...</td><td>0</td></tr><tr><td>b7</td><td>0</td><td>...</td><td>0</td></tr><tr><td>b6</td><td>0</td><td>...</td><td>0</td></tr><tr><td>b5</td><td>0</td><td>...</td><td>0</td></tr><tr><td>b4</td><td>0</td><td>...</td><td>0</td></tr><tr><td>b3</td><td>0</td><td>...</td><td>0</td></tr><tr><td>b2</td><td>0</td><td>...</td><td>0</td></tr><tr><td>b1</td><td>0</td><td>...</td><td>0</td></tr></table>0: Utan avbrott.<br>1: Avbrott.<br>(Endast Q, ej Q00J, Q00, Q01)</td><td>S<br>(Fel)</td></tr>
</table>

<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD1350 -<br>SD1381</td><td>Extern spännings-matning ej ansluten</td><td>Bitmönster som visar vilka moduler vars externa spänningsmatning ej är ansluten.</td><td>Av bitmönstret kan utläsas vilka moduler vars spänningsmatning ej är ansluten. (Om modulens adress är ändrad i parametrarna visas den adress som angetts i parametrarna.)<br><br>
<table>
<tr><th></th><th>SD1350</th><th>...</th><th>SD1381</th></tr>
<tr><td>b15</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b14</td><td>0</td><td>...</td><td>1</td></tr>
<tr><td>b13</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b12</td><td>1</td><td>...</td><td>0</td></tr>
<tr><td>b11</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b10</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b9</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b8</td><td>1</td><td>...</td><td>0</td></tr>
<tr><td>b7</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b6</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b5</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b4</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b3</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b2</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b1</td><td>0</td><td>...</td><td>0</td></tr>
</table>
0: Extern spänningsmatning ej ansluten<br>1: Extern spänningsmatning ansluten</td><td>S<br>(Fel)</td></tr>
</table>

<table>
<tr><th>Adress</th><th>Namn</th><th>Betydelse</th><th>Beskrivning</th><th>Aktiveras av</th></tr>
<tr><td>SD1400-SD1431</td><td>I/O-modul verifikationsfel</td><td>Bitmönster som visar moduler med verifikationsfel</td><td>När spänningen ansluts, kommer de I/O moduler vars information skiljer sig från de registrerade parametrarna indikera I/O-verifikationsfel i bitmönstret.<br><br>
<table>
<tr><th></th><th>SD1400</th><th>...</th><th>SD1431</th></tr>
<tr><td>b15</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b14</td><td>0</td><td>...</td><td>1</td></tr>
<tr><td>b13</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b12</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b11</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b10</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b9</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b8</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b7</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b6</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b5</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b4</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b3</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b2</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b1</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>b0</td><td>0</td><td>...</td><td>0</td></tr>
</table>
0: Inget fel.<br>1: Verifikationsfel.<br>(Ej Q00J, Q00, Q01)</td><td>S<br>(Fel)</td></tr>
</table>

# Sakregister


### Symboler

- *8-38*
$+ *8-58*
* *8-42*
+ *8-38*
/ *8-42*

### A

ACOS *8-172*
Addition
  *$+ 8-58*
  *+ 8-38*
  *B+ 8-46*
  *D+ 8-40*
  *DB+ 8-48*
  *E+ 8-54*
ANB *8-6*
AND *8-3*
AND$_ *8-36*
AND_ *8-31*
ANDD_ *8-33*
ANDE_ *8-35*
ANDF *8-5*
ANDP *8-5*
ANI *8-3*
Arbetssätt
  *Direkt I/O-avläsning 2-3*
  *I/O-kopiering 2-2*
ASC *8-160*
ASIN *8-170*
ATAN *8-174*

### B

B- *8-46*
B* *8-50*
B+ *8-46*
B/ *8-50*
BAND *8-186*
Batteribackup *4-12*
Batterimatade minnesceller *4-6*
BCD *8-63*
Behörighetskod *4-8*
BIN *8-66*
BMOV *8-86*
BREAK *8-141*
BRST *8-123*
BSET *8-123*
BSFL *8-119*
BSFR *8-119*

### C

CALL *8-143*
CJ *8-93*
CML *8-83*
COM *8-148*
COS *8-166*
Cykeltid *2-1, 2-3, 2-4*
  *Konstant 2-4, 2-5*
  *Tidsövervakning 2-6*

### D

D- *8-40*
D* *8-44*
D*P *8-44*
D+ *8-40*
D/ *8-44*
D/P *8-44*
DAND *8-101*
Dataförflyttning
  *$MOV 8-81*
  *BMOV 8-86*
  *EMOV 8-80*
  *FMOV 8-88*
  *MOV 8-78*
Dataregister *3-10*
DB- *8-48*
DB* *8-52*
DB+ *8-48*
DB/ *8-52*
DBAND *8-186*
DBCD *8-63*
DBIN *8-66*
DBL *8-73*
DCML *8-83*
DDEC *8-61*
DEC *8-59*
DECO *8-132*
DEG *8-177*
Dekad *5-5*
DFLT *8-69*
DFRO *8-155*
DI *3-24, 8-96*
DINC *8-61*

DINT *8-71*
DIS *8-135*
Division
*/ 8-42*
*B/ 8-50*
*D/ 8-44*
*DB/ 8-52*
*E/ 8-56*
DLIMIT *8-184*
DMOV *3-10, 8-78*
DNEG *8-75*
DOR *8-103*
DRCL *8-115*
DRCR *8-113*
DROL *8-115*
DROR *8-113*
DSER *8-127*
DSFL *8-121*
DSFR *8-121*
DSUM *8-129*
DTEST *8-125*
DTO *8-157*
DUTY *8-193*
DXCH *8-90*
DXNR *8-107*
DXOR *8-105*

**E**

E- *8-54*
E* *8-56*
E+ *8-54*
E/ *8-56*
EI *3-24, 8-96*
EN *2-1*
ENCO *8-133*
END *8-26*
ENEG *8-77*
EXP *8-179*
Extern PAUSE-signal *4-13*
Extern RUN-signal *4-13*

**F**

Felindikering *4-13*
Felkoder *4-6, 9-1*
Felminne *4-7*
Felminnesceller *3-3*
FEND *2-1, 8-26*
FF *8-21*
FIFO-register
*Läs 8-151, 8-153*
*Skriv 8-150*
FIFR *8-151*
FIFW *8-150*
Filregister *2-8, 2-11, 4-12*
Flip-Flop
*FF 8-21*
FLT *8-69*
Flyttal *7-14*
FMOV *8-88*
FOR *8-139*
FPOP *8-153*
FROM *8-155*
Funktionsmodul
*Läs, FROM, DFRO 8-155*
*Skriv, TO, DTO 8-157*

**G**

GOEND *8-93*

**H**

HEX *8-162*
Hexadecimala *5-3*
Huvudprogram *4-12*

**I**

I/O-adressering *4-1, 4-13*
I/O-kopiering *2-2*
I/O-RAM *2-2*
INC *8-59*
Indexering *5-6*
Indexregister *3-14*
Ingångar *3-1*
Instruktionsbeskrivning *5-1*
INT *8-71*
Interrupt *2-5, 4-7*
*Pekare 3-19*
*Räknare 3-9*
INV *8-11*
Invertering *8-83*
IRET *8-99*

**J**

JMP *8-93*
Jämförelse *8-31*
*Binär 8-31, 8-33*
*Flyttal 8-35*
*Teckensträng 8-36*

**K**

Klocka *4-9*
Kommunikationsminnesceller B *3-3*
Konstanter *3-25, 5-3*

**L**
LATCH CLEAR *3-2, 3-10, 3-12*
LD *8-3*
LD$_ *8-36*
LD_ *8-31*
LDD_ *8-33*
LDE_ *8-35*
LDF *8-5*
LDI *8-3*
LDP *8-5*
LIMIT *8-184*
LOG *8-181*

**M**
MASK *8-96*
Master Control *3-15*
MC *8-24*
MCR *8-24*
MELSEC NET *3-3, 3-12, 4-12*
Minnesceller *3-2, 3-3*
Minneskort
*Formatera 2-11*
MOV *3-10, 8-78*
MPP *8-8*
MPS *8-8*
MRD *8-8*
Multiplikation
** 8-42*
*B* 8-50*
*D* 8-44*
*DB* 8-52*
*E* 8-56*

**N**
NEG *8-75*
NEXT *8-139*
Noggrannhet *3-4*
NOP *8-29*
Numeriska värde *5-3*

**O**
Omvandling
*16 bitar 8-74*
*32 bitar 8-73*
*ASCII 8-160*
*BCD 8-63*
*Binär 8-66*
*Flyttal 8-69*
*Grader 8-177*
*Heltal 8-71*
*Hexadecimal 8-162*
*Radianer 8-176*
Operander *3-1*
*Dataregister 3-10*
*Felminnesceller 3-3*
*Filregister 3-13*
*Indexregister 3-14*
*Ingångar 3-1*
*Interrupträknare 3-9*
*Kommunikationsregister 3-12*
*Konstanter 3-25*
*Markörer 3-15*
*Minnesceller 3-2*
*Pekare 3-17*
*Räknare 3-8*
*Tidskretsar 3-4*
*Utgångar 3-1*
Operation *3-10*
OR *8-3*
OR$_ *8-36*
OR_ *8-31*
ORB *8-6*
ORD_ *8-33*
ORE_ *8-35*
ORF *8-5*
ORI *8-3*
ORP *8-5*
OUT *8-12*
OUT C *8-15*
OUT T *8-13*
OUTH T *8-13*

**P**
Parameterinställning *4-12*
Parametrar *4-12*
PC-kort *2-7*
*EEPROM 2-8*
*Formatera 2-8*
Pekare *3-17*
PLF *8-19*
PLS *8-19*
Programbearbetning *2-1*
Programcykel *2-3*
Programminne *2-7*
*Formatera 2-10*

**R**
RAD *8-176*
RCL *8-111*
RCR *8-109*
Realtidsklocka *4-9*
RESET-omkopplare *3-2, 3-10*
RET *8-145*
RFS *8-100*
RND *8-183*
ROL *8-111*
ROR *8-109*
Rotation
*DROL, DRCL 8-115*
*DROR, DRCR 8-113*
*ROL, RCL 8-111*
*ROR, RCR 8-109*
RSET *8-190*
RST *8-17*
RUN/PAUSE *4-13*
Räknare *3-8*
*Hastighet 3-9*
*Interrupträknare 3-9*

**S**
SCJ *8-93*
SER *8-127*
SET *8-17*
SFL *8-117*
SFR *8-117*
SFT *8-22*
SIN *8-164*
SQR *8-178*
SRND *8-183*
Status vid STOP - RUN *4-13*
STOP *8-28*
Subtraktion
*- 8-38*
*B- 8-46*
*D- 8-40*
*DB- 8-48*
*E- 8-54*
SUM *8-129*
SWAP *8-92*
Säkerhetsinformation *1-1*

**T**
TAN *8-168*
Teckenbyte
*DNEG 8-75*
*ENEG 8-77*
*NEG 8-75*
TEST *8-125*
Tidskretsar *3-4, 4-13*
*Ackumulerande 3-4, 3-7*
*High speed 3-5*
*Low speed 3-5*
*Utökade 3-5*
TO *8-157*
Tom modulplats *4-1, 4-2*
Trigonometri
*ACOS 8-172*
*ASIN 8-170*
*ATAN 8-174*
*COS 8-166*
*DEG 8-177*
*RAD 8-176*
*SIN 8-164*
*TAN 8-168*
T-vippa
*FF 8-21*

**U**
UNI *8-137*
Uppdatering
*In- och utgångar, RFS 8-100*
Utbyte av modul under drift *4-4*
Utgångar *3-1*
Utgångsstatus *4-7*

**W**
WAND *8-101*
Watch *2-6*
Watchdog-timer *2-6*
WDT *8-192*
WOR *8-103*
WORD *8-74*
WXNR *8-107*
WXOR *8-105*

**X**
XCH *8-90*

**Ö**
Överordnat villkor
*MC, MCR 8-24*