

Maskinorienterad Programmering 2010/2011

MC68HC12, Stencil
"Assemblerprogrammering.pdf"
CPU12 Reference Guide
Arbetsbok för MC12, avsnitt 2

Ur innehållet:

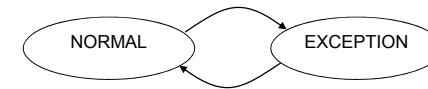
- Processorns tillstånd
- Undantagshantering ("Exceptions")
 - Auto-vektoriserat
 - Vektoriserat
 - Avbrottsprioritering

Exekveringstillstånd

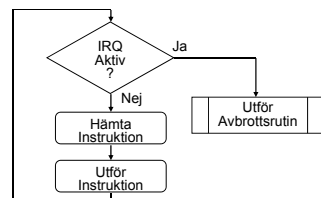
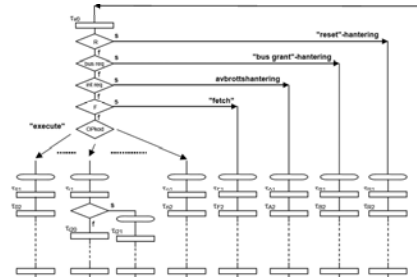
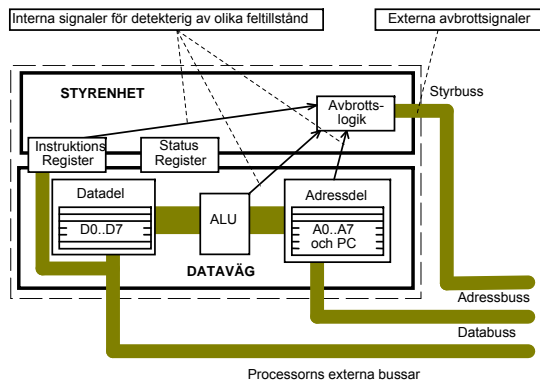
Processorn befinner sig alltid i något av tillstånden:

NORMAL, processorn hämtar och utför instruktioner, dvs. normal exekvering.

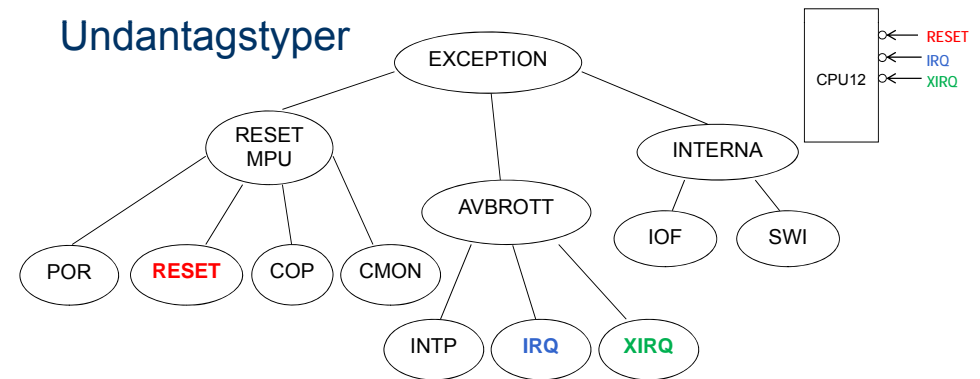
EXCEPTION, något "undantag" har inträffat som gör att processorn inte kan (eller ska) fortsätta normal exekvering.



Test av Undantagstillstånd



Undantagstyper



RESET MPU, händelser som alltid föranleder återstart (RESET) av processorn.

AVBROTT, externa händelser, dvs. utanför processorn, detta kan alltså vara enheter på samma krets som processorn (sammanbyggda periferienheter), det kan också vara en speciell insignal (IRQ eller XIRQ) som aktiveras.

INTERNA, händelser som uppträder under programexekvering, exempelvis att en otillåten instruktion avkodas eller den speciella instruktionen SWI.

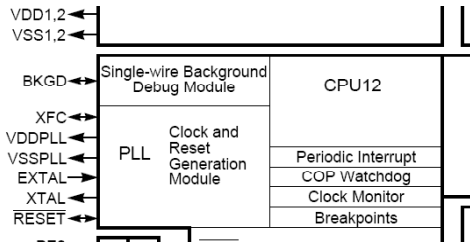
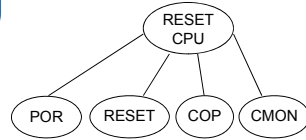
Fatala fel, kräver RESET av CPU

POR, *Power On Reset*, vid spänningstillslag

RESET, insignal till processorn aktiveras.

COP, *Computer Operating Properly*, så kallad *watchdog-funktion*.

CMON, *Clock Monitor Reset*, övervakar E-klockan, om frekvensen sjunker under 10 kHz genereras RESET.



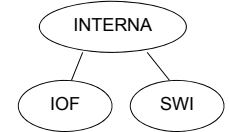
Adress (hex)	Funktion
FFFE	RESET, Startvektor
FFFC	Clock Monitor Fail
FFFA	COP Watchdog Timeout
FFF8	Illegal Op Code
FFF6	SWI
FFF4	XIRQ
FFF2	IRQ
FF00-FFF0	Enhetsspecifika vektorer, skiljer sig något beroende på olika varianter

Interna undantag

Om processorn avkodar en otillåten operationskod kallas detta *Illegal Opcode Fetch* (IOF).

Processorn avbryter då, **sparar registerinnehåll på stacken**, läser autovektor för IOF och utför undantagshantering.

Instruktionen *SoftWare Interrupt* (SWI) fungerar på samma sätt, men har en annan autovektor och en bestämd operationskod.



Adress (hex)	Funktion
FFFE	RESET, Startvektor
FFFC	Clock Monitor Fail
FFFA	COP Watchdog Timeout
FFF8	Illegal Op Code
FFF6	SWI
FFF4	XIRQ
FFF2	IRQ
FF00-FFF0	Enhetsspecifika vektorer, skiljer sig något beroende på olika varianter

EXEMPEL, Hantera "Software Interrupt", SWI

```

main    ORG    $1000
        LDAB  #$11
        LDAA  #$22
        LDX   #$3333
        LDY   #$4444
        NOP
        SWI
        NOP
        BRA  main
    
```

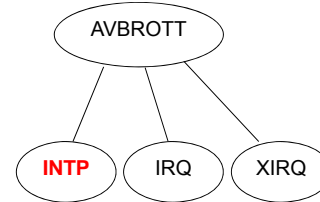
```

SWI_hantering:
        CLRA
        NOP
        RTI

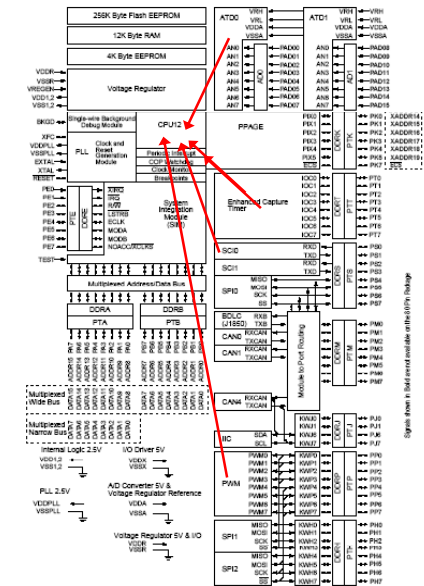
        ORG    $FFF6
        FDB   SWI_hantering
    
```

Adress (hex)	Funktion
FFFE	RESET, Startvektor
FFFC	Clock Monitor Fail
FFFA	COP Watchdog Timeout
FFF8	Illegal Op Code
FFF6	SWI
FFF4	XIRQ
FFF2	IRQ
FF00-FFF0	Enhetsspecifika vektorer, skiljer sig något beroende på olika varianter

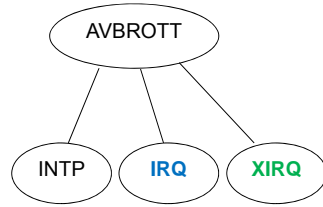
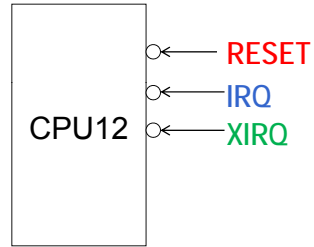
Internt genererade avbrott



Adress (hex)	Funktion
FFF0	Real Time Interrupt
FFFE	Enhanced Capture Timer channel
FFEC	Enhanced Capture Timer channel 1
FFEA	Enhanced Capture Timer channel 2
....
FF8E	Port P Interrupt
FF8C	PWM Emergency Shutdown
FF8A-FF80	Reserverade



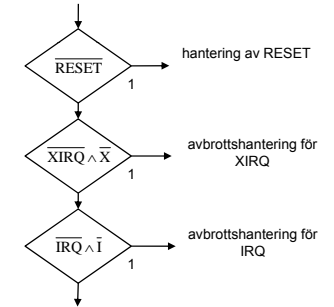
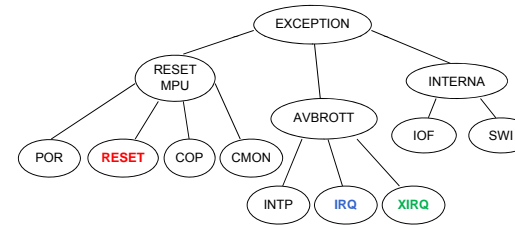
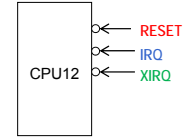
Externt genererade avbrott



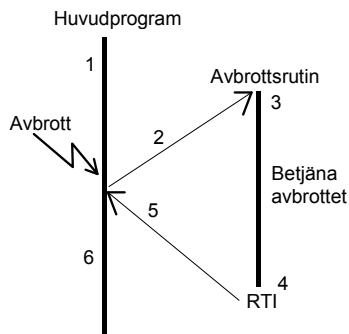
Adress (hex)	Funktion
FFFE	RESET, Startvektor
FFFC	Clock Monitor Fail
FFFA	COP Watchdog Timeout
FFF8	Illegal Op Code
FFF6	SWI
FFF4	XIRQ
FFF2	IRQ
FF00-FFF0	Enhetsspecifika vektorer, skiljer sig något beroende på olika varianter

Undantags prioritet

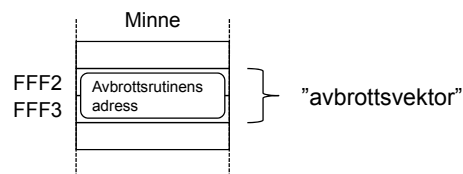
1. RESET MPU och "INTERNA" , (alltid)
2. XIRQ, (om X i CCR är 0)
3. IRQ, (om I i CCR är noll)



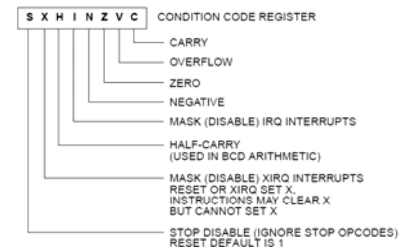
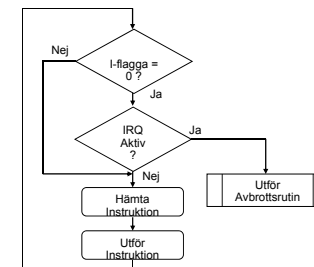
Avbrottshantering



- 1) Huvudprogram exekveras när ett avbrott aktiveras
- 2) Hopp till avbrottsrutin
- 3) Avbrottsrutin startar
- 4) Avbrottsrutin avslutas med en speciell instruktion, *return from interrupt (RTI)*
- 5) Återhopp till huvudprogram
- 6) Huvudprogrammet fortsätter.



Maskering av avbrott



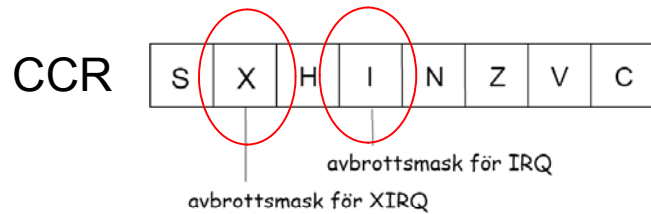
Maskera avbrott:
SEI
Alternativt
ORCC # %00010000

Demaskera avbrott:
CLI
Alternativt
ANDCC # %11101111

Demaskera X-avbrott:
ANDCC # %10111111
OBS: Kan INTE maskeras ("Non Maskable Interrupt")

CPU12 – Beteende vid RESET/Avbrott

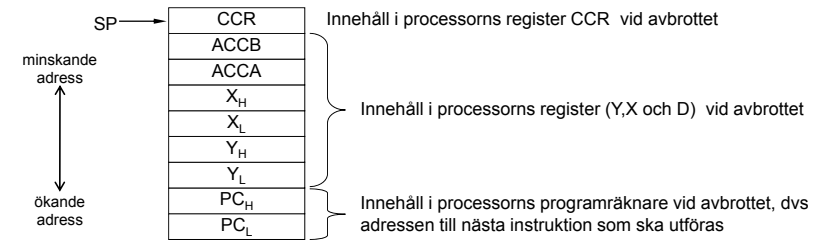
RESET	XIRQ	IRQ
CCR = 1101000	REGISTERS->[SP]	REGISTERS->[SP]
PC=[FFFE,FFFF]	CCR[I]=1	CCR[I]=1
	PC=[FFF4,FFF5]	PC=[FFF2,FFF3]



Spara registerinnehåll...

”Atomär operation”
 PUSH PC
 PUSH Y
 PUSH X
 PUSH D
 PUSH CCR

Stackens utseende i avbrottsrutin



Avslut av avbrottsrutin ”ReTurn from Interrupt”, RTI

RTI Return from Interrupt

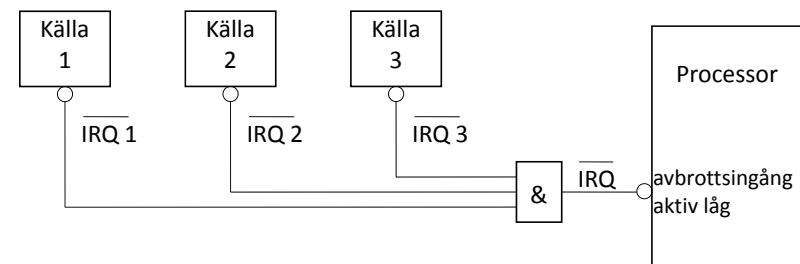
Operation: $(M(SP) \Rightarrow CCR; (SP) + \$0001 \Rightarrow SP$
 $(M(SP) : M(SP + 1)) \Rightarrow B : A; (SP) + \$0002 \Rightarrow SP$
 $(M(SP) : M(SP + 1)) \Rightarrow XH : XL; (SP) + \$0004 \Rightarrow SP$
 $(M(SP) : M(SP + 1)) \Rightarrow PCH : PL; (SP) - \$0002 \Rightarrow SP$
 $(M(SP) : M(SP + 1)) \Rightarrow YH : YL; (SP) + \$0004 \Rightarrow SP$

Description: Restores system context after interrupt service processing is completed. The condition codes, accumulators B and A, index register X, the PC, and index register Y are restored to a state pulled from the stack. The X mask bit may be cleared as a result of an RTI instruction, but cannot be set if it was cleared prior to execution of the RTI instruction.

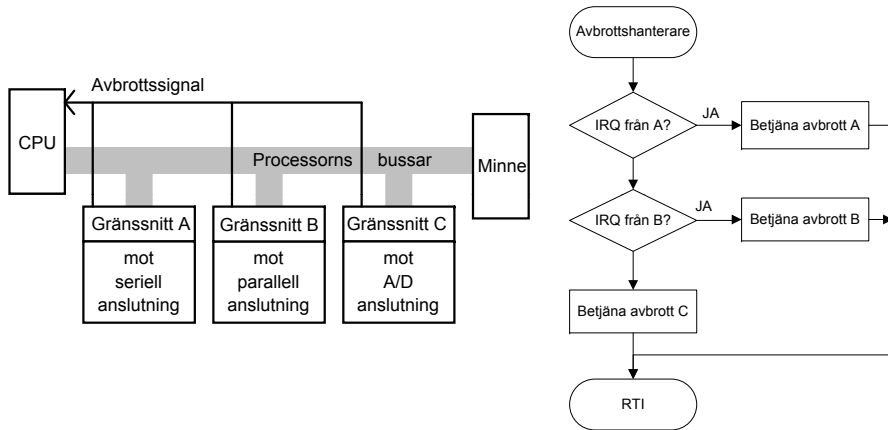
RTI ”Atomär operation”

PULL CCR
 PULL D
 PULL X
 PULL Y
 PULL PC

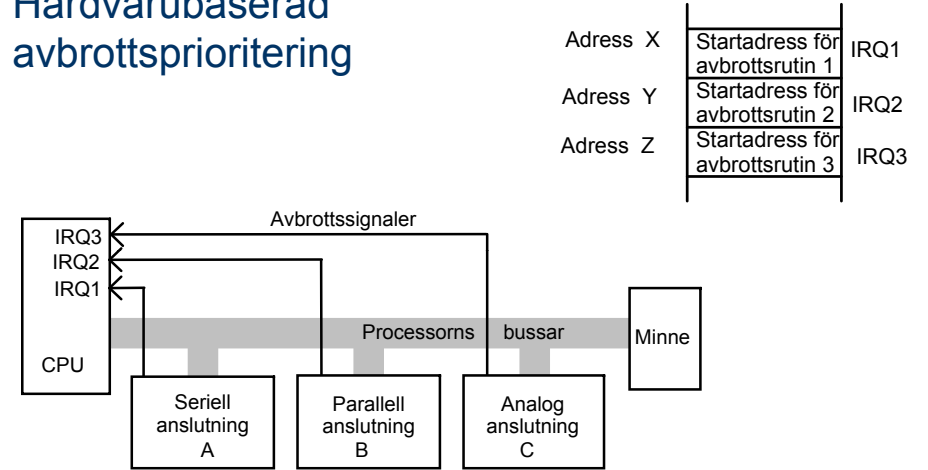
Multipla avbrottskällor



Programbaserad avbrottsprioritering



Hårdvarubaserad avbrottsprioritering



Intern avbrottsprioritering

För avbrott från interna kretsar bestäms prioriteten av avbrottsvektorns adress.

Ju högre adress, desto högre prioritet.

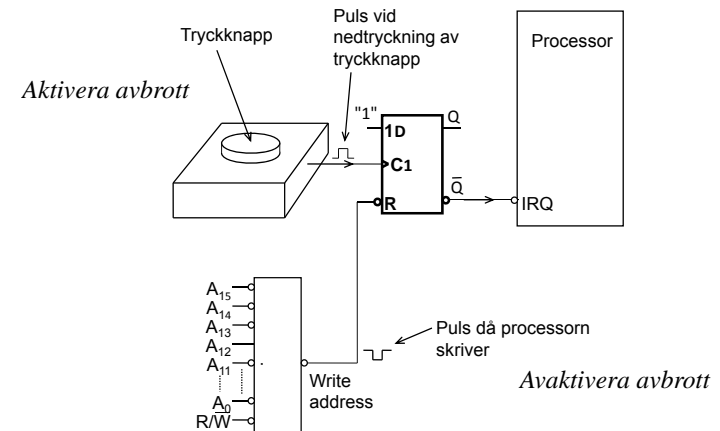
Det finns vissa möjligheter att ändra detta programmässigt.

Högre prioritet

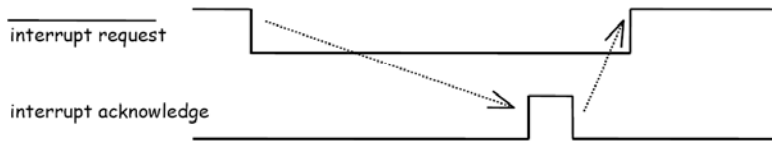
Lägre prioritet

Adress	Funktion
FF00	Real Time Interrupt
FFEE	Enhanced Capture Timer channel
FFEC	Enhanced Capture Timer channel 1
FFEA	Enhanced Capture Timer channel 2
FFE8	Enhanced Capture Timer channel 3
FFE6	Enhanced Capture Timer channel 4
FFE4	Enhanced Capture Timer channel 5
FFE2	Enhanced Capture Timer channel 6
FFE0	Enhanced Capture Timer channel 7
FFDE	Enhanced Capture Timer overflow
FFDC	Pulse accumulator A overflow
FFDA	Pulse accumulator input edge
FFD8	SPI0
FFD6	SCI0
FFD4	SCI1
FFD2	ATD0
FFD0	ATD1
FFCE	Port J
FFCC	Port H
FFCA	Modulus Down Counter underflow
FFC8	Pulse Accumulator B Overflow
FFC6	PLC lock
FFC4	CRG Self Clock Mode
FFC2	Används ej (BDLC)
FFC0	IIC Bus
FFBE	SPI1
FFBC	Reserverad
FFBA	EEPROM 1-bit
FFB8	FLASH 1-bit
FFB6	CAN0 wake-up
FFB4	CAN0 errors
FFB2	CAN0 receive
FFB0	CAN0 transmit
FF9E	CAN4 wake-up
FF94	CAN4 errors
FF92	CAN4 receive
FF90	CAN4 transmit
FF8E	Port P Interrupt
FF8C	PWM Emergency Shutdown
FF8A	Reserverade
FF80	Reserverade

Avbrottsvipa

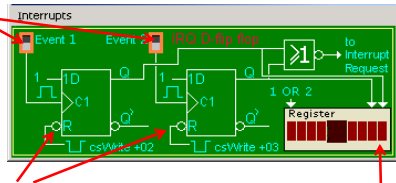


Kvittering av avbrott "Interrupt Acknowledge"



EXEMPEL: (jfr: laborationskort ML19)

Aktivera avbrott



Nollställ ("kvittera") avbrott

Status hos avbrottskällor

EXEMPEL, Arbetsbok uppgift 45 ("IRQ4.S12")

```

; Definitioner, initieringssekvens
; och avbrottsvektor
PARPORT1 EQU    $0880
PARPORT2 EQU    $0881

IrqStat EQU    $0D00
IrqRes1 EQU    $0D02
IrqRes2 EQU    $0D03

        ORG    $1000
* Nollställ våra variabler
        CLR    Var1
        CLR    Var2
        CLR    IrqRes1
        CLR    IrqRes2

* Sätt om avbrottsmasken hos processorn
        CLI

* Initiera avbrottsvektor IRQ
        ORG    $FFF2
        FDB    IrqR
    
```

```

; Huvudprogram
Loop
        LDAB    Var1
        ADDB    #1
        STAB    Var1
        STAB    PARPORT1

        LDAB    Var2
        STAB    PARPORT2
        BRA    Loop

* Variabler
Var1    RMB    1
Var2    RMB    1
    
```

EXEMPEL, forts.

```

* Avbrottsrutin
IrqR:
        LDAA    IrqStat
        BITA    #2    ; Event 2 ?
        BEQ    IrqR1 ; Om inte prova nästa
        CLR    IrqRes2
        INC    Var2    ; Räkna upp

IrqR1:
* Kontrollera även Event 1...
        BITA    #1
        BEQ    IrqR2
        CLR    IrqRes1
        CLR    Var2    ; Nollställ

IrqR2:
        RTI
    
```

```

IrqStat EQU    $0D00
IrqRes1 EQU    $0D02
IrqRes2 EQU    $0D03
    
```

Applikation för avbrott (IRQ)

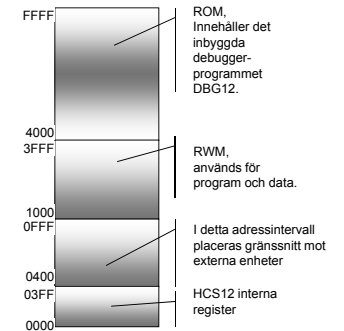
```

; Initieringssekvens
        ORG    XXXX
Application_Start:
        LDS    #TopOfStack
        ...
        ; nollställ I-flagga
        ANDCC # $FE
        JSR    _main
        ...
    
```

```

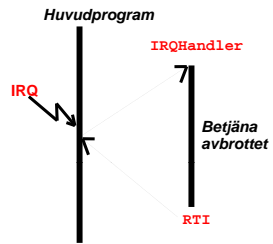
        ORG    $FFF2
        FDB    irq_service_routine
        ORG    $FFFE
        FDB    Application_Start

; Avbrottsshanterare
irq_service_routine:
        RTI
    
```



I laborationssystemet (MC12) kan vi INTE placera avbrottsvektorena på deras rätta platser (konflikt med DBG12) I stället placeras dom i RWM

MC12 (DBG12) och avbrott



Vektor ROM	Funktion
FFFE	RESET, Startvektor
FFFC	Clock Monitor Fail, JMP [3FFE]
FFFA	COP Watchdog Timeout, JMP [3FFC]
FFF8	Illegal Op Code, JMP [3FFA]
FFF6	SWI, JMP [3FF6]
FFF4	XIRQ, JMP [3FF4]
FFF2	IRQ, JMP [3FF2]
FF8C	Enhetsspecifika vektorer
FFF0	

Simulator

```

ORG $FFF2
FDB irq_service_routine
; Avbrottshanterare
irq_service_routine:
RTI
    
```

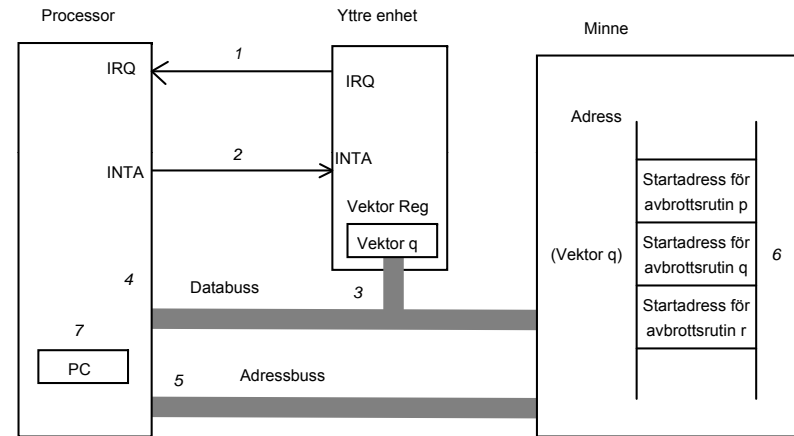
MC12

```

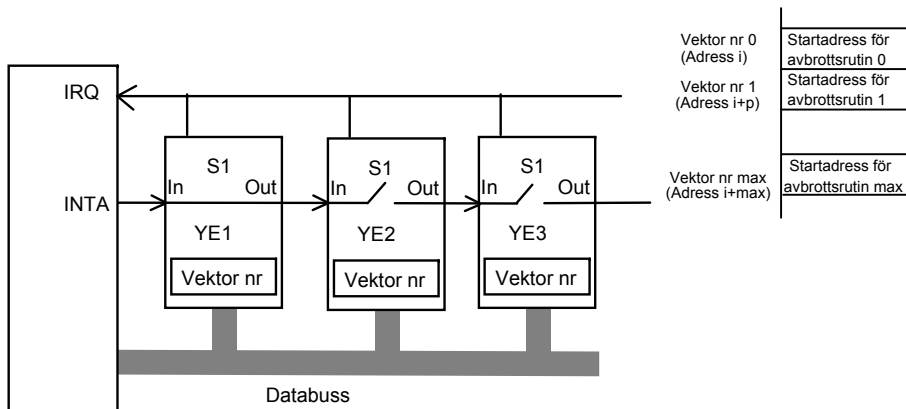
ORG $3FF2
FDB irq_service_routine
; Avbrottshanterare
irq_service_routine:
RTI
    
```

Vektoravbrott

Somliga periferikretsar konstrueras för att tillhandahålla avbrottsvektor (kan ej anslutas till HCS12)



Prioritet vid vektoravbrott (Daisy chain)



EXEMPEL, "ColdFire" (MC68x00)

Vektor nr	Adress (offset) (hex)	Funktion
0	000	Initial stackpekare
1	004	Initial programräknare
2	008	Access Error (ex: referens till adress där minne/periferikrets ej finns)
3	00C	Address Error (ex: referens till udda adress med word operand)
4	010	Illegal instruktion (icke-definierad operationskod)
5	014	Division med 0
6,7	018, 01C	Reserverade
8	020	Privilege Violation, försök att utföra supervisor-instruktion i user mode
9	024	Trace, en-instruktions exekvering
10	028	Line 1010, reserverad operationskod
11	02C	Line 1111, reserverad operationskod
12	030	Non-PC breakpoint debug interrupt
13	034	PC breakpoint debug interrupt
14	038	Format error
15	03C	Avbrott från enhet som ej tillhandahållit avbrottsnummer
16-23	040-05F	Reserverade vektorer
24	060	Icke-identifierat avbrott
25	064	Autovektor avbrottsnivå 1
26	068	Autovektor avbrottsnivå 2
27	06C	Autovektor avbrottsnivå 3
28	070	Autovektor avbrottsnivå 4
29	074	Autovektor avbrottsnivå 5
30	078	Autovektor avbrottsnivå 6
31	07C	Autovektor avbrottsnivå 7
32-47	080-0BF	Trap vektor för instruktionen TRAP #<vektor nummer>
48-63	0C0-0FF	Undantag vid flyttalshantering, Reserverade vektorer
64-255	100-3FF	Användardefinierade vektorer

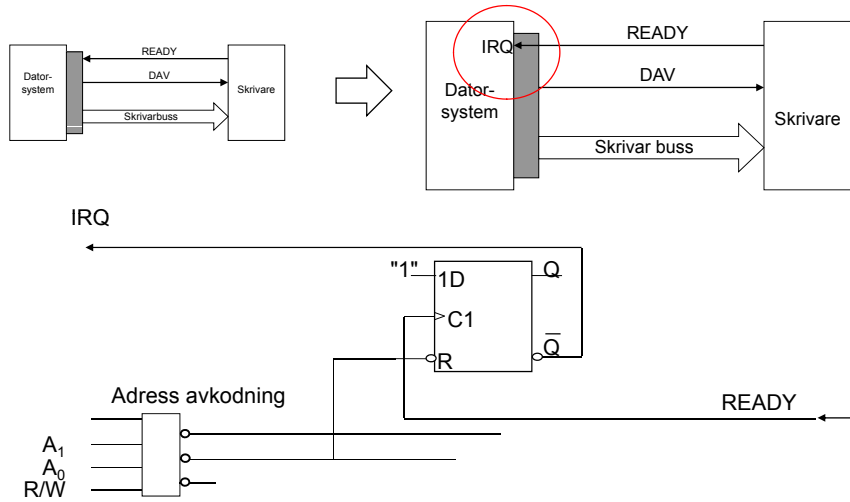
← RESET vektor

”Undantag” genererade av program

Autovektor avbrott

← Vektoravbrott

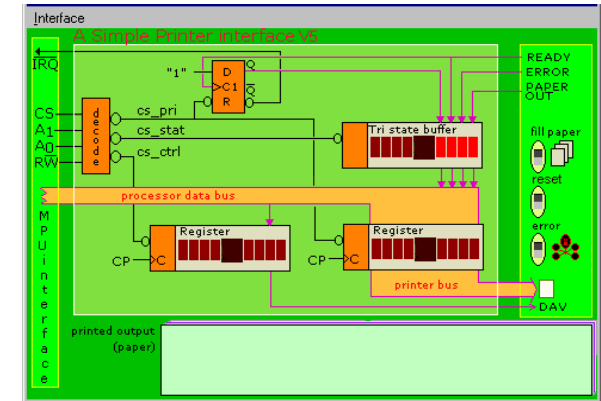
Skrivarport med avbrott, Version 5



Skrivarport, Version 5

```

PRINTER EQU    $0800
PSTATUS EQU    $0801
PCONTROL EQU   $0802
    
```



Denna lösning genererar ALLTID avbrott då skrivarens teckenbuffert är tom....

Skrivarport, Version 6

I vår sista lösning kan vi stänga av avbrotten från skrivaren.
”Disable Interrupt”

