

Maskinorienterad Programmering 2010/2011

"Skrivarporten", Arbetsbok MC12, avsnitt 2

Ur innehållet:

Vi ansluter en skrivare

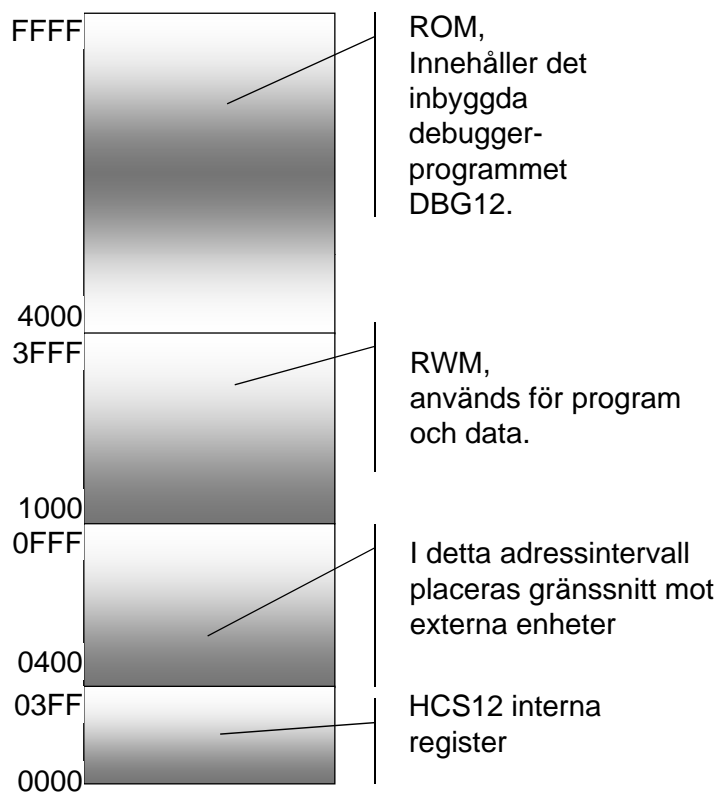
Skrivarport

Arbetsbok, kap. 2

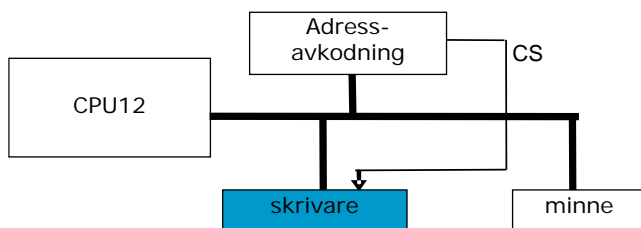
- ❑ Konstruktion av en Printer-Port
- ❑ Synkroniseringsproblem
- ❑ Villkorlig / ovillkorlig överföring
"Busy Wait" och "Polling"
- ❑ Handskakningssignaler

MC12 adressrum

Beskrivande figur över hur minnesmoduler och IO-portar är placerade i minnet



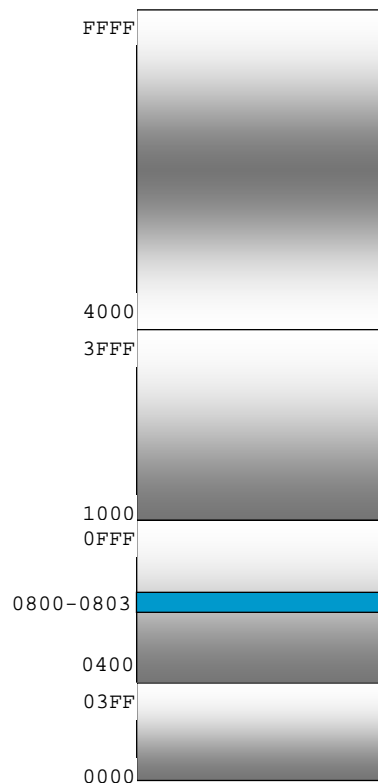
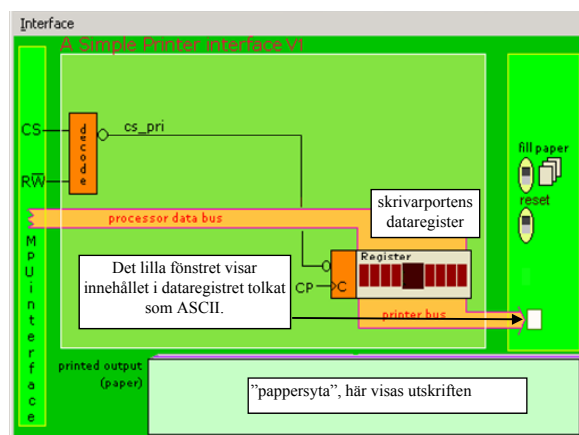
Adressavkodning för skrivar-porten



Med portdefinition
PRINTER EQU \$0800

och instruktionerna
LDAA #30
STAA PRINTER

överförs hexadecimala värdet 30 till skrivaren



Förutsättningar

Vår skrivare är från början en "dum" skrivare:

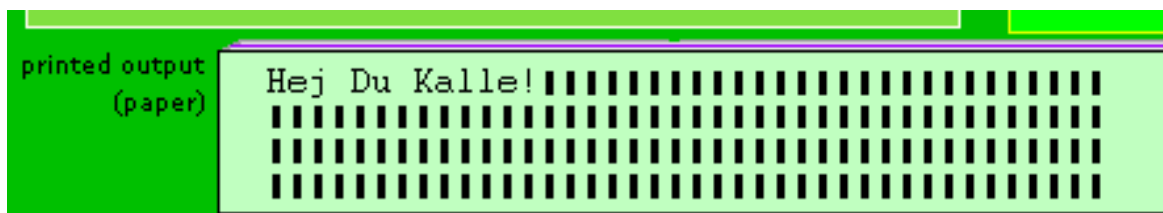
- ❑ Den kan endast arbeta med **ett tecken i taget**.
(hämtar ett tecken - skriv ut - hämta nästa)
- ❑ Det finns inledningsvis **inga handskaknings-signaler**
- ❑ Max utskriftshastighet: **4 tecken per sekund**.

Första programexemplet

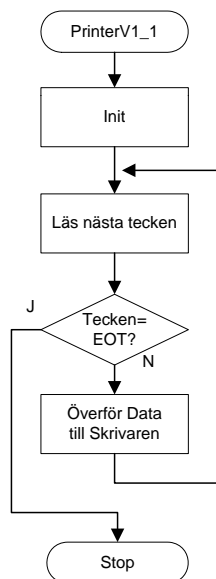
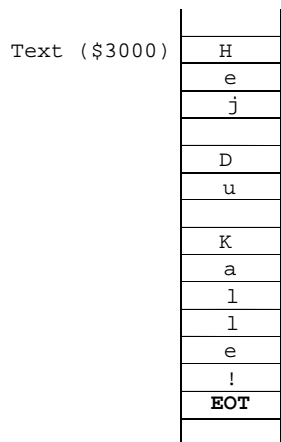
Text (\$3000)

```
* Printer V1_0
   ORG      $1000
   LDX      #Text          Pekare till textsträng -> X
Loop LDAA   1,X+          Tecken -> A, peka på nästa
   STAA    PRINTER       Skriv ut till port
   BRA     Loop          Fortsätt med nästa tecken
; så här kan du använda assemblerdirektiv för att
; skapa textsträngen på adress 3000:
   ORG     $3000
Text  FCS    "Hej Du Kalle!"
```

H
e
j
D
u
K
a
l
l
e
!

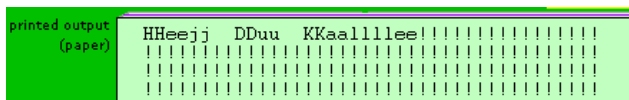


Inför specialtecken för strängslut



```

* Printer V1_1
PRINTER EQU    $0800
EOT EQU    4
ORG    $1000
LDX    #Text
Loop: LDAA    1,X+
      CMPA    #EOT
      BEQ    Stop
      STAA    PRINTER
      BRA    Loop
Stop:  NOP
      BRA    Stop
      ORG    $3000
Text:  FCS    "Hej Du Kalle!"
      FCB    EOT
    
```

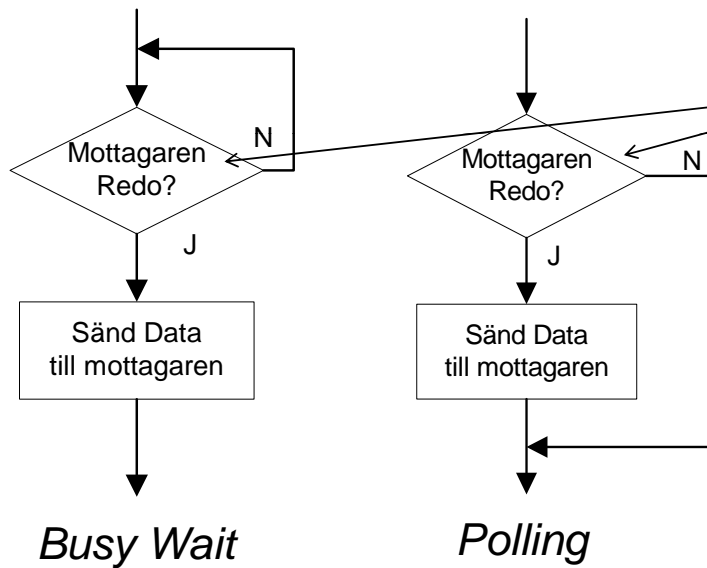


Synkronisera arbetstakerna ...

	Skrivare	Simulator STEP	Simulator RUN	Simulator RUN FAST	Hårdvara
Instruktioner/ sekund		?	10	1000	1 000 000
Tecken/ sekund	4	?	2	200	200 000

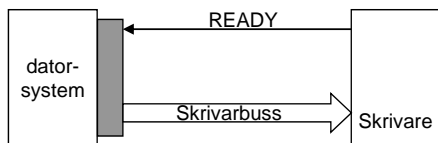
Lösningen blir villkorlig överföring vilket kräver ett asynkront gränssnitt...

Villkorlig överföring



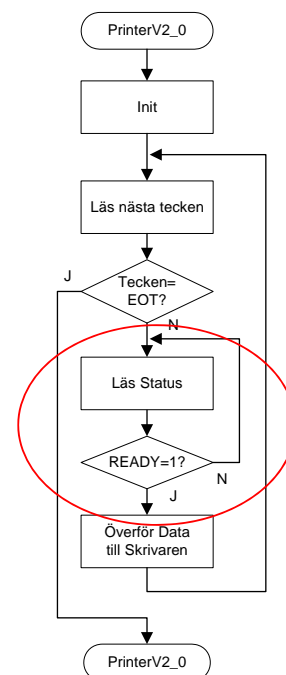
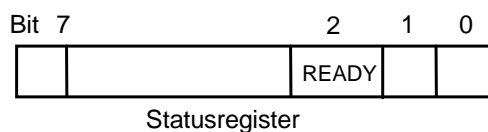
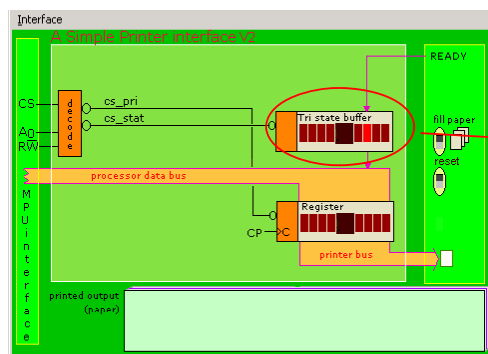
Statustest, kräver asynkront gränssnitt...

Gränssnitt, version 2



READY-signalen definieras enligt:

- READY = 1 (Hög nivå) indikerar att skrivaren är klar att ta emot ett nytt tecken.
- READY = 0 (Låg nivå) indikerar att skrivaren är upptagen med att skriva ut ett tecken.



”Programmerarens bild”

READY = 1 (Hög nivå): skrivaren är REDO
 READY = 0 (Låg nivå): skrivaren är UPPTAGEN



```

* Printer V2_1
PRINTER EQU    $0800
PSTATUS EQU    $0801
EOT EQU        4
                ORG    $1000
                LDX    #Text
Loop:           LDAA   1,X+
                CMPA   #EOT
                BEQ    Exit
LoopForReady:
                LDAB   PSTATUS
                ANDB   #4
                BEQ    LoopForReady
                STAA   PRINTER
LoopForNotReady:
                LDAB   PSTATUS
                ANDB   #4
                BNEQ   LoopForReady

                BRA    Loop
Exit:          NOP
                BRA    Exit

                ORG    $3000
Text:         FCS    "Hej Du Kalle!"
                FCB    EOT
    
```

Resultat

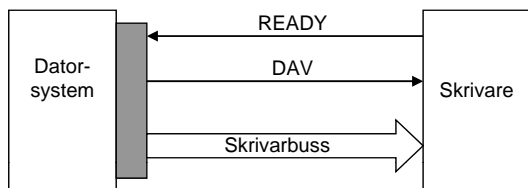
Klarar nu situationen att centralenheten arbetar snabbare än skrivaren.

Fortfarande problem då centralenheten är långsammare än skrivaren.

Fortfarande problem med att få skrivaren att stoppa då sista tecknet skrivits ut.

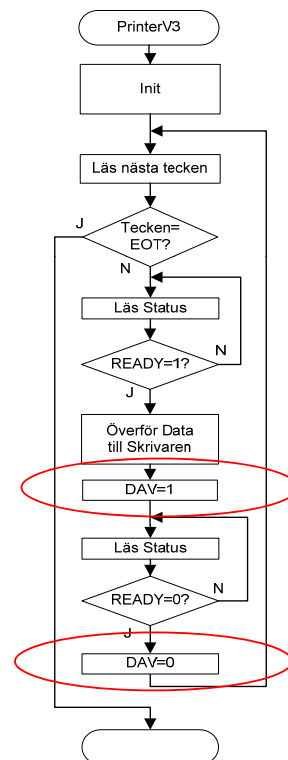
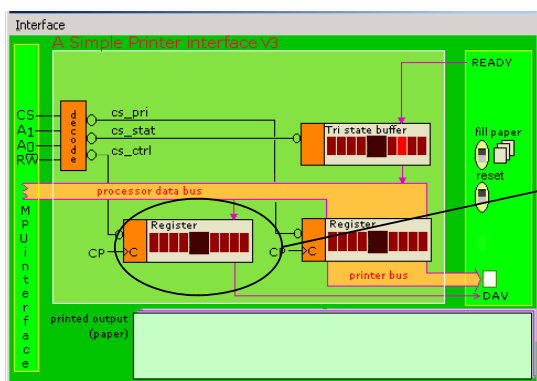
Vi behöver ytterligare handskakningssignal "Tecken finns"...

Gränssnitt, version 3



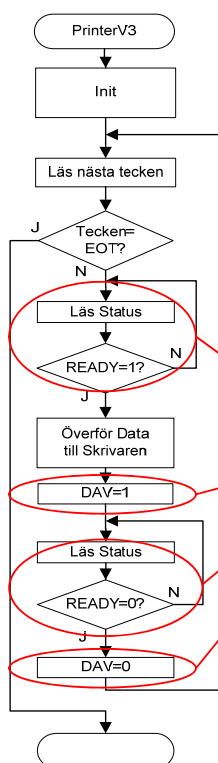
DAV = 1 (Hög nivå) indikerar för skrivaren att giltigt tecken finns att hämta på skrivarbussen.

DAV = 0 (Låg nivå) indikerar för skrivaren att skrivarbussen har ett ogiltigt värde.



Händelser i Datorsystemet
Inväntar READY=1
När READY=1 skrivs nästa tecken till skrivarens dataregister. Sätter DAV=1
Inväntar READY=0
När READY=0 nollställs DAV som indikation på att det inte finns giltigt tecken på skrivarbussen

Händelser i skrivaren
Skrivaren är upptagen med att skriva ut ett tecken. READY=0.
Skrivaren är redo för nästa tecken och sätter READY=1
Inväntar DAV=1
Ser att DAV=1. Läser nytt tecken från skrivarbussen. Signalerar upptagen, READY=0.
Skrivaren är upptagen med att skriva ut ett tecken. READY=0.



* Printer V3

```

PRINTER EQU    $0800
PSTATUS EQU    $0801
PCONTROL EQU   $0802
EOT EQU       4
    
```

```

ORG    $1000
LDX    #Text
Loop:  LDAA  1,X+
      CMPA  #EOT
      BEQ   Stop
    
```

```

Ready:  BRCLR  PSTATUS,#4,Ready
      STAA  PRINTER
      BSET  PCONTROL,#2
NotReady: BRSET  PSTATUS,#4,NotReady
      BCLR  PCONTROL,#2
      BRA  Loop
    
```

```

Stop:  NOP
      BRA  Stop
    
```

```

ORG    $3000
Text:  FCS    "Hej Du Kalle!"
      FCB    EOT
    
```

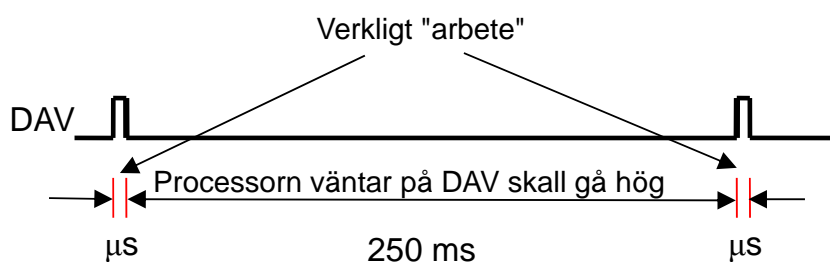
Resultat

Klarar nu situationen att centralenheten arbetar snabbare än skrivaren.

Klarar nu situationen då centralenheten är långsammare än skrivaren.

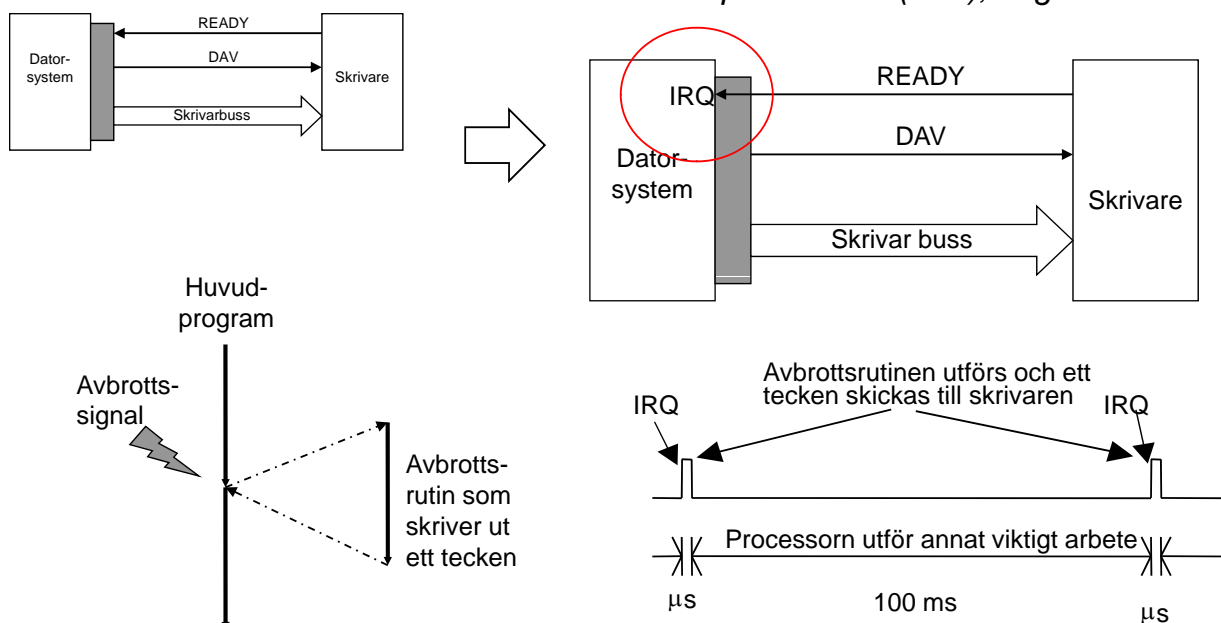
Klarar nu situationen med att få skrivaren att stoppa då sista tecknet skrivits ut

Lösningen är dock hopplöst ineffektiv med tanke på hur vi utnyttjar systemet...

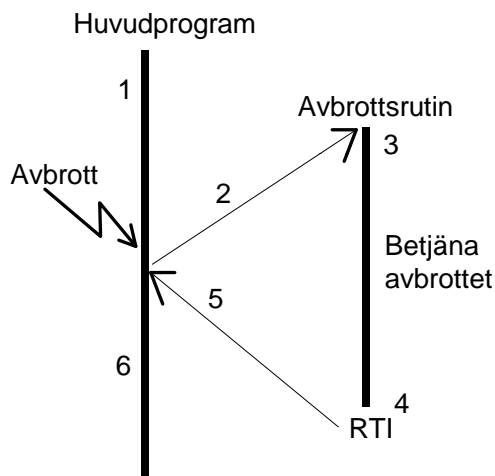


Introduktion till "Undantagshantering"

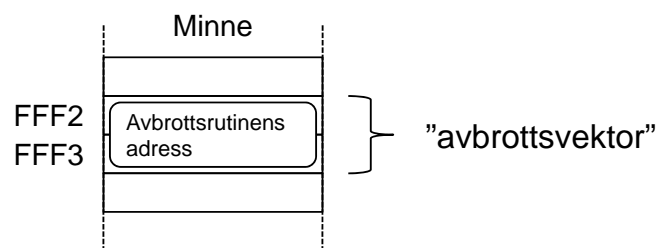
Interrupt ReQuest (IRQ), begäran om avbrott...



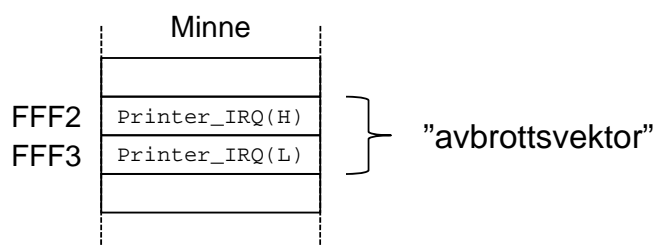
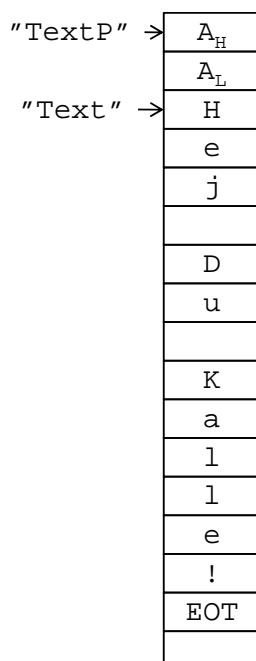
Avbrott



- 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.



EXEMPEL, Skrivarporten



* Avbrottsrutin

Printer_IRQ:

```
LDX  TextP  ;Läs pekare till nästa
LDAA 1,x+   ;Skriv nästa tecken
STAA  Printer
STX   TextP  ;Spara nya pekare
RTI
```