

**LV6 Fo14**

Arb s 164

**Aktivera Kursens mål:**

- ▶ Konstruera en dator mha grindar och programmera denna

**Aktivera Förra veckans mål:**

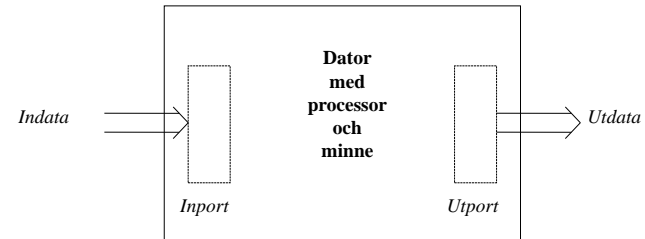
- ▶ Konstruera styrenheten
- ▶ Skriva (strukturerade) assemblerprogram för FLISP
  - ▶ Villkorliga hopp
  - ▶ Kontrollstrukturer "if then else" etc.
  - ▶ Subrutiner

**Veckans mål:**

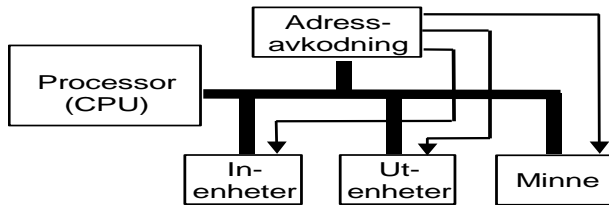
- ▶ Ansluta In- och Utportar till Flisp
- ▶ In-och utmatning
- ▶ Avbrott
- ▶ Adressavkodning
- ▶ Ansluta minnen och I/O-moduler

**Läs LITTA  
Lär dig mer!**

**In och Utportar på FLISP  
Hårdvara**



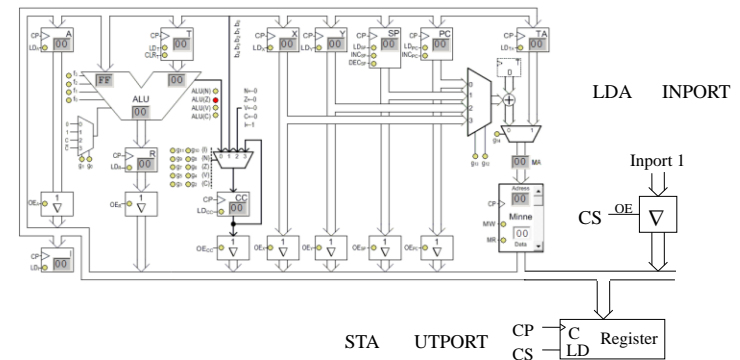
**In och Utportar på FLEX**



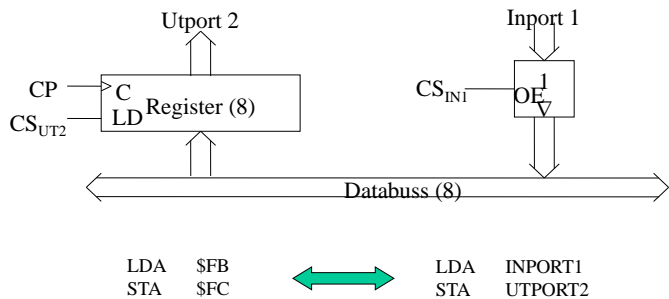
LDA	Adr	LDA	INPORT
STA	Annan Adr	STA	UTPORT

**In och Utportar på FLISP**

Arb s 164



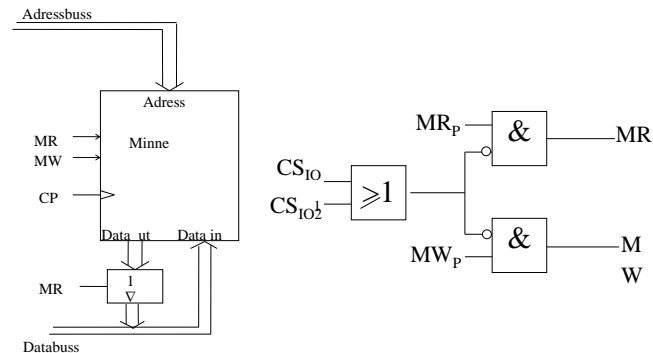
## In och Utportar på FLEX



Grundläggande datorteknik OHLV6

5

## In och Utportar på FLISP



Grundläggande datorteknik OHLV6

6

## LV6 Fo14

### Dagens mål. Du ska :

- ▶ Kunna ansluta In- och Utportar till Flisp
- ▶ **Förstå problem i samband med in-och utmatning**
  - ▶ Beskriva Villkorlig/Ovillkorlig överföring
  - ▶ Förstå enkla IO-program som utnyttjar Busy Wait / Polling - principer
- ▶ Förstå avbrott
  - ▶ Skriva mycket enkla avbrottsrutiner

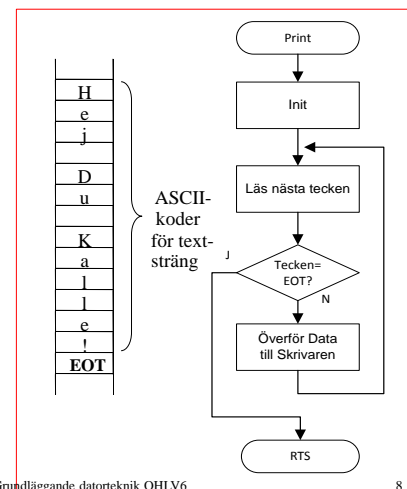
Läs lurtt!  
Lär dig mere!

Grundläggande datorteknik OHLV6

7

## Ett huvudprogram

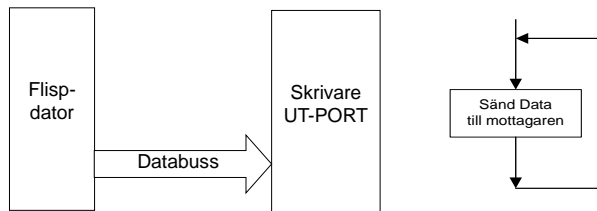
```
Huvudprogram
Main ---
    jsr Rut 1
    jsr Rut 2
    ---
    jsr Print
    ---
    jsr Rut i
    ---
    ---
bra Main
```



Grundläggande datorteknik OHLV6

8

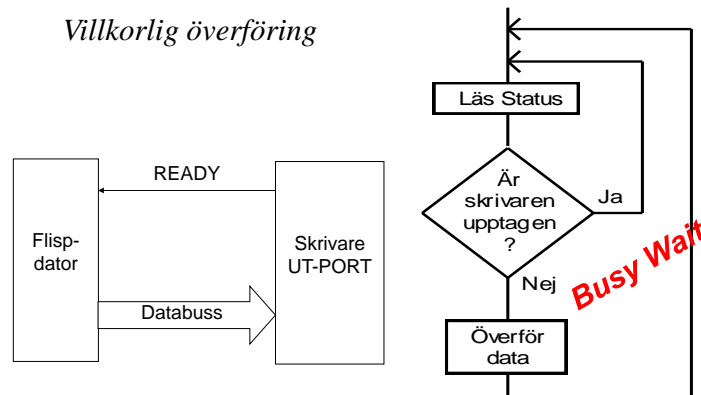
### Ovillkorlig överföring



Grundläggande datorteknik OHLV6

9

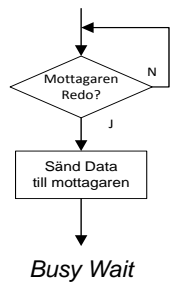
### Villkorlig överföring



Grundläggande datorteknik OHLV6

10

### Villkorlig överföring

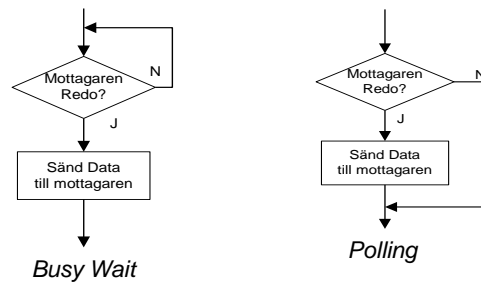


```
Huvudprogram
Main ---
  jsr Rut 1
  jsr Rut 2
  ---
  jsr Print
  ---
  jsr Rut i
  ---
  bra Main
```

Grundläggande datorteknik OHLV6

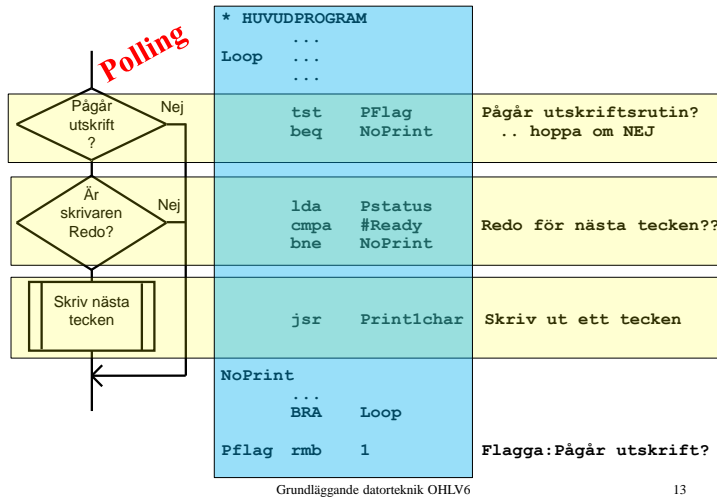
11

### Villkorlig överföring



Grundläggande datorteknik OHLV6

12



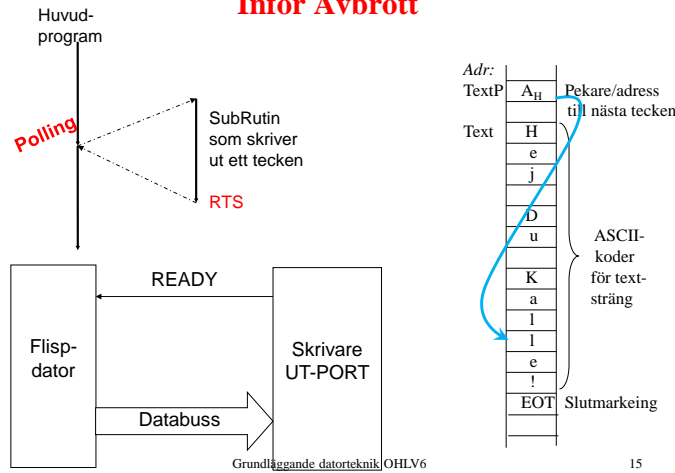
**Dagens mål. Du ska :**

- ▶ Kunna ansluta In- och Utportar till Flisp
- ▶ Förstå problem i samband med in-och utmatning
  - ▶ Beskriva Villkorlig/Ovillkorlig överföring
  - ▶ Förstå enkla IO-program som utnyttjar Busy Wait / Polling - principer
- ▶ **Förstå avbrott**
  - ▶ Skriva mycket enkla avbrottsrutiner

**Läs klart!  
Lär dig mer!**

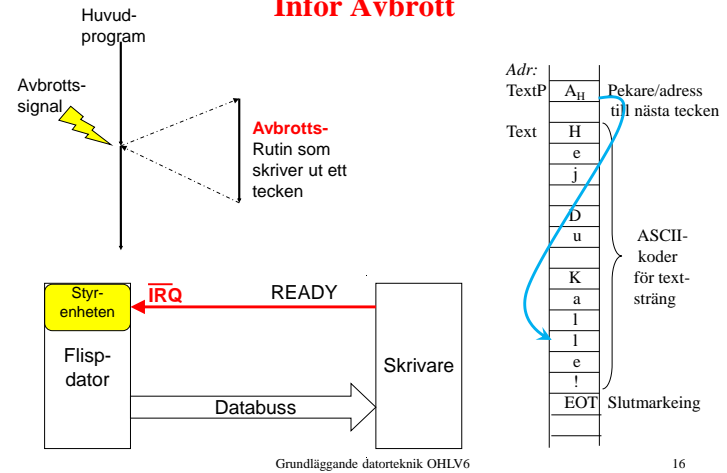
Grundläggande datorteknik OHLV6

**Inför Avbrott**



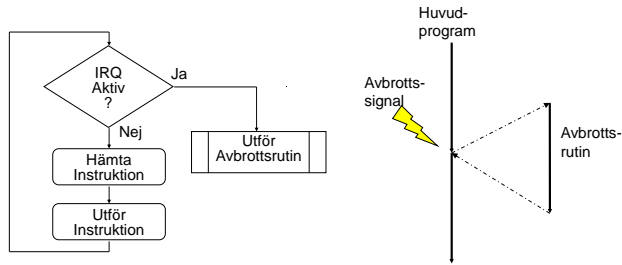
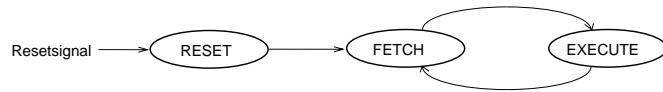
Grundläggande datorteknik OHLV6

**Inför Avbrott**



Grundläggande datorteknik OHLV6

## Avbrott – forts. Hur jobbar processorn?



Grundläggande datorteknik OHLV6

17

## Avbrott - forts

S

### HITILLS:

Programmeraren (DU) bestämmer när:

**JSR Printlchar      Skriv ett tecken**

### VID AVBROT:

Skrivaren bestämmer när:

**Printlchar      Skriv ett tecken**

### VAD KRÄVS?

- spara undan sin PC
- starta en rutin som skriver ut *ett* tecken till skrivaren för att
- sedan återstarta huvudprogrammet

Grundläggande datorteknik OHLV6

18

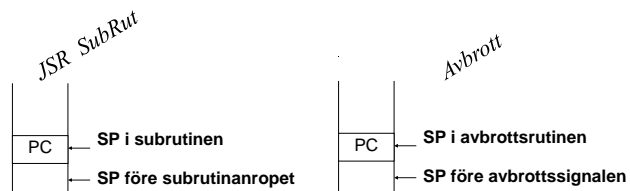
### VAD KRÄVS?

## Avbrott – forts.

S

1. spara undan sin PC
2. starta en rutin som skriver ut *ett* tecken till skrivaren för att
3. sedan återstarta huvudprogrammet

1) **Spara undan sitt PC** (jfr JSR SubRut)



Grundläggande datorteknik OHLV6

19

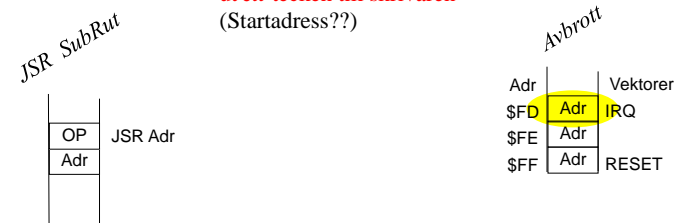
### VAD KRÄVS?

## Avbrott – forts.

S

1. spara undan sin PC
2. starta en rutin som skriver ut *ett* tecken till skrivaren för att
3. sedan återstarta huvudprogrammet

2) **starta upp en rutin som skriver ut *ett* tecken till skrivaren (Startadress??)**

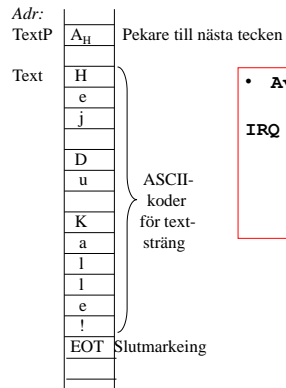


Grundläggande datorteknik OHLV6

20

## Avbrott - forts

s



```

• Avbrottsrutin

IRQ  ldx  TextP  Läs pekare till nästa
      lda  ,x+   Skriv nästa tecken
      sta  Printer
      stx  TextP  Spara nya pekare
      rti
    
```

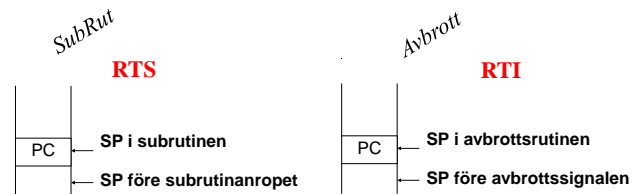
## Avbrott – forts.

S

### VAD KRÄVS?

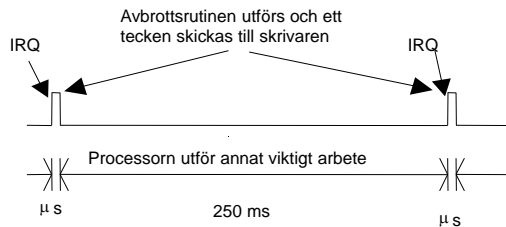
1. spara undan sin PC
2. starta en rutin som skriver ut *ett* tecken till skrivaren för att
3. sedan återstarta huvudprogrammet

3) sedan återstarta huvudprogrammet

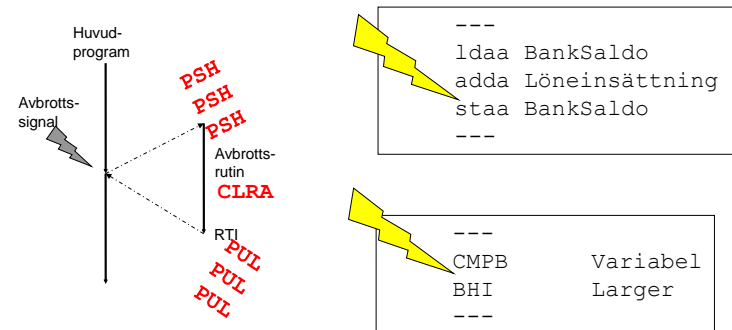


## Avbrott - forts

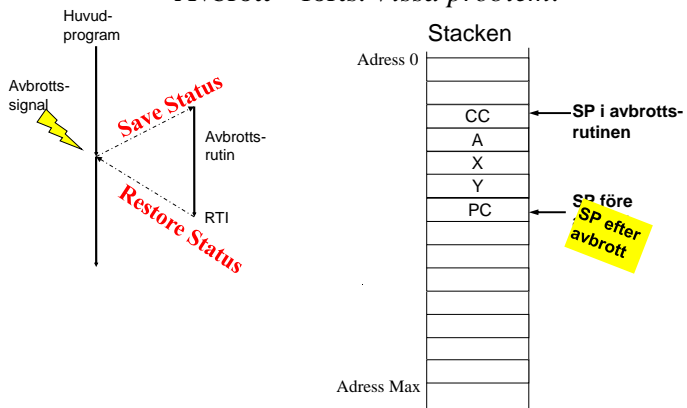
s



## Avbrott – forts. Vissa problem.



## Avbrott – forts. *Vissa problem.*



Grundläggande datorteknik OHLV6

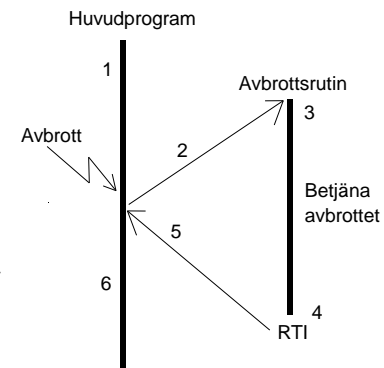
25

## Avbrott – forts.

S

Vi summerar händelseförloppet vid ett avbrott.

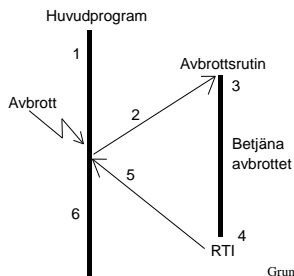
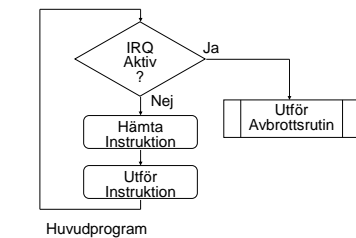
- 1) Huvudprogrammet kör
- 2) Hopp till avbrottsrutin
- 3) Save status
- 4) Hämta avbrottsvektor
- 5) Återhopp till huvudprogram
- 6) Restore status
- 7) Huvudprogrammet fortsätter



Grundläggande datorteknik OHLV6

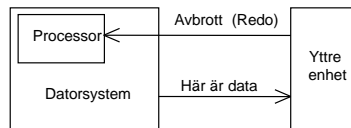
26

## Avbrott – forts. *Flera problem.*



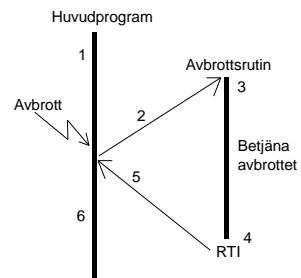
Grundläggande datorteknik OHLV6

27

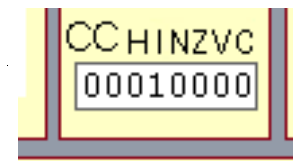


## Avbrott – forts..

S32



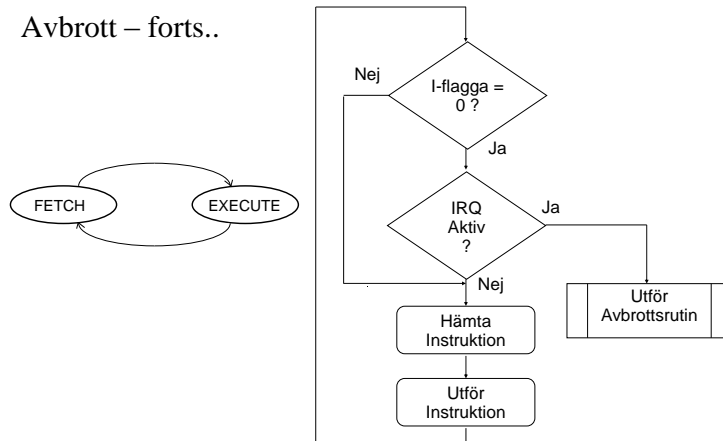
Lösning:  
Ny flagga i CC-registret



Grundläggande datorteknik OHLV6

28

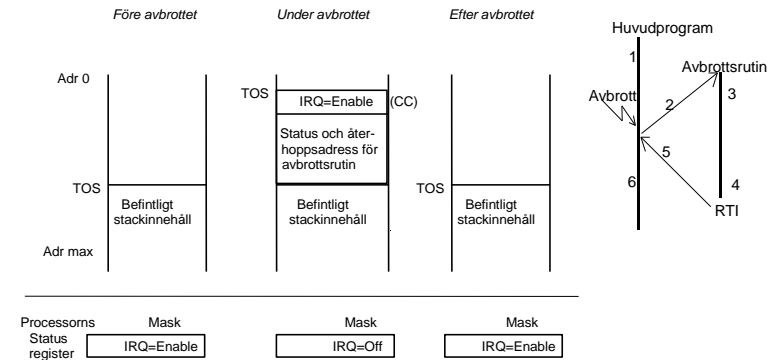
## Avbrott – forts..



Grundläggande datorteknik OHLV6

29

## Avbrott – forts.. Bild av stacken före, under o efter avbrott <sup>S</sup>



Grundläggande datorteknik OHLV6

30

## LV6 Fo14

### Dagens mål. Du ska :

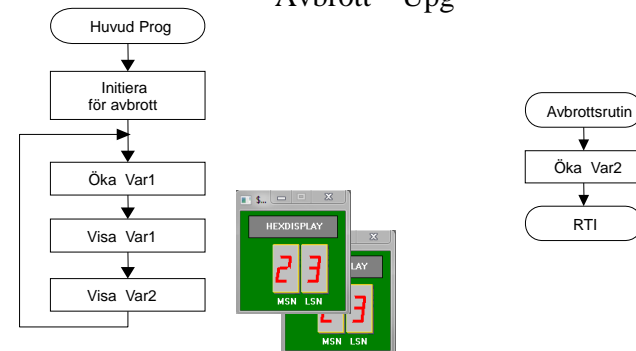
- ▶ Kunna ansluta In- och Utportar till Flisp
- ▶ Förstå problem i samband med in-och utmatning
  - ▶ Beskriva Villkorlig/Ovillkorlig överföring
  - ▶ Förstå enkla IO-program som utnyttjar Busy Wait / Polling - principer
- ▶ Förstå avbrott
  - ▶ **Skriva mycket enkla avbrottsrutiner**

**Läs lurtt!  
Lär dig mere!**

Grundläggande datorteknik OHLV6

31

## Avbrott – Upp



Grundläggande datorteknik OHLV6

32



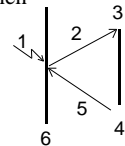
## Avbrott - Sammanfattning

Begrepp:

- Avbrottsingång på processorn IRQ (Interrupt)
- I-flagga i CC-registret (Maskera avbrott, brandvägg)
- Avbrottsvektor (Startadress för avbrottsrutin)
- Save Status (Placera alla regs på stacken)
- Avbrottsrutin ("subrutin" avslutad med RTI)
- Återhopp från avbrottsrutin (RTI)
- Restore Status (Återhämta alla regs från stack)

## Avbrott - Sammanfattning

1. **OM I-flaggan=0:** Processorn känner att IRQ är aktiverad och slutför utförandet av pågående instruktion.
2. Processorn sparar huvudprogrammets återhopsadress och övriga registerinnehåll på stacken, *save status*. Därefter läser processorn startadressen för avbrottsrutinen från IRQ-vektorn (från adress \$FD). Denna startadress placeras i PC. I-flaggan ETT-ställs
3. Avbrottsrutinen startas (med I-flaggan=1).
4. Avbrottsrutinen avslutas med instruktionen RTI som får processorn att utföra *restore status*, dvs registerinnehållen återställs från stacken (med gamla I=0).
5. Återhopp till huvudprogram.
6. Därmed återstartas huvudprogrammet där det blev avbrutet



### Veckans mål:

- ▶ Ansluta In- och Utportar till Flisp
- ▶ In-och utmatning
- ▶ Avbrott
- ▶ Adressavkodning
- ▶ Ansluta minnen och I/O-moduler

### Dagens mål: Du ska kunna....

- ▶ Skilja på olika minnestyper
- ▶ Förklara principer för olika bussprotokoll
- ▶ Förstå begreppen Timing och VM (Valid Memory Adress)
- ▶ Konstruera adressavkodningslogik ....
  - ▶ 1) Fullständig Adr Avk
  - ▶ 2) Ofullständig Adr Avk
  - ▶ Processorns adressrum

LV6 Fo15

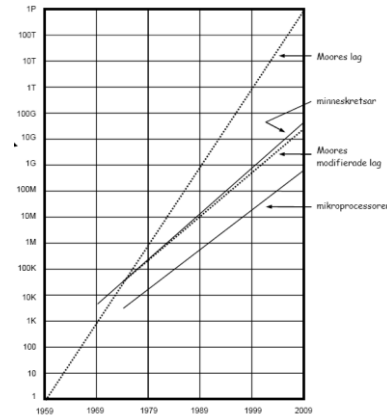
Läs lurtt!  
Lär dig mer!

## Lite om Minnen

•"RAM"	Random Access Memory
•RWM	Read Write Memory
•SRAM	Statiskt RAM
•DRAM	Dynamiskt
•ROM	Read Only Memory
•PROM	Programmable ROM
•EPROM	Erasable PROM
•EEPROM	Electrically EPROM
•FLASH	

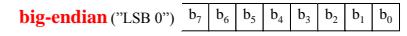
# Mikroelektronikens utveckling

Antalet transistorer som ryms på en kiselbricka....

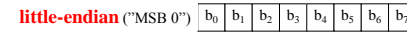


Grundläggande datorteknik OHLV6

## Bit ordning "bit endianness"

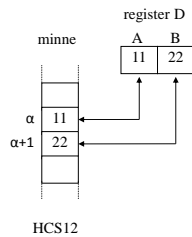


8-bitars ord där b<sub>7</sub> är den MEST signifikanta biten och b<sub>0</sub> den MINST signifikanta biten

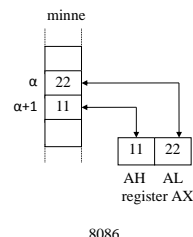


Grundläggande datorteknik OHLV6

## HCS12/Intel 8086 byte ordning



(big endian)



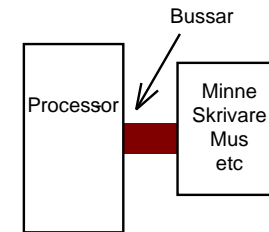
(little endian)

Grundläggande datorteknik OHLV6

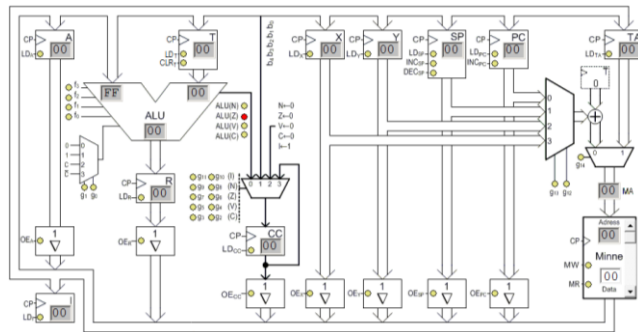
## LV6 Fo15

### Dagens mål: Du ska kunna....

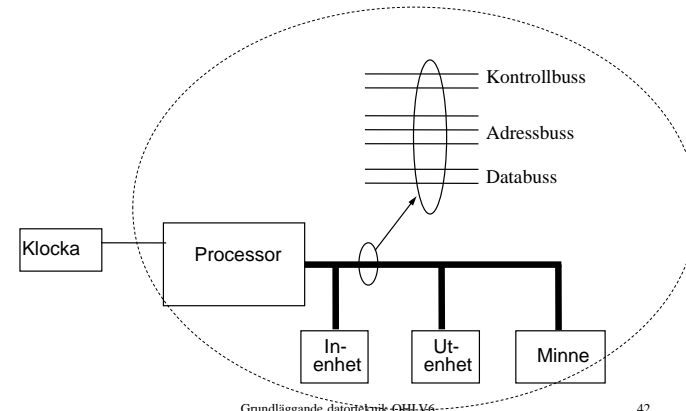
- ▶ Skilja på olika minnestyper
- ▶ **Förklara principer för olika bussprotokoll**
- ▶ Förstå begreppen Timing och VM (Valid Memory Adress)
- ▶ Konstruera adressavkodningslogik ....
  - ▶ 1) Fullständig Adr Avk
  - ▶ 2) Ofullständig Adr Avk
  - ▶ Processorns adressrum



Grundläggande datorteknik OHLV6

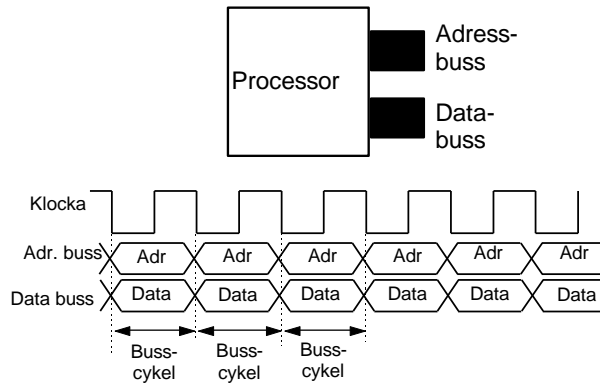


Grundläggande dator teknik OHLV6



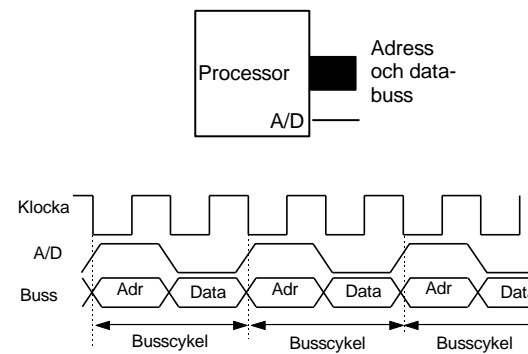
Grundläggande dator teknik OHLV6

### Icke Multiplexad Buss

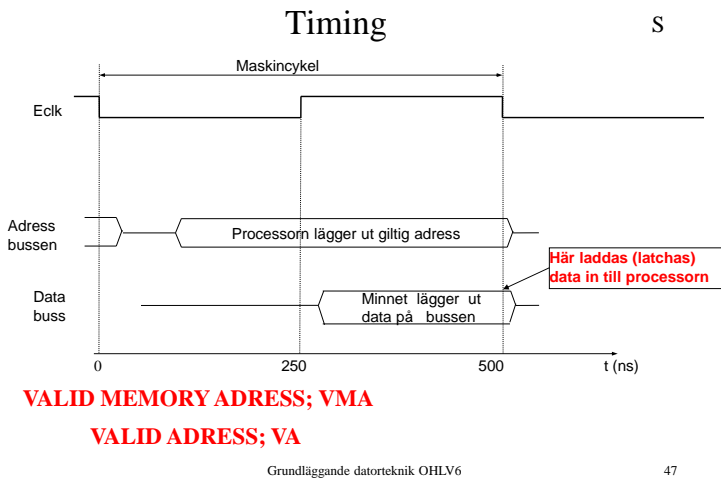
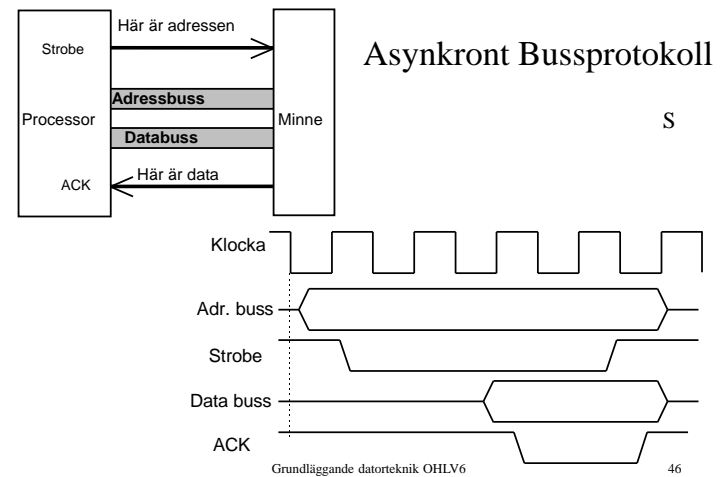
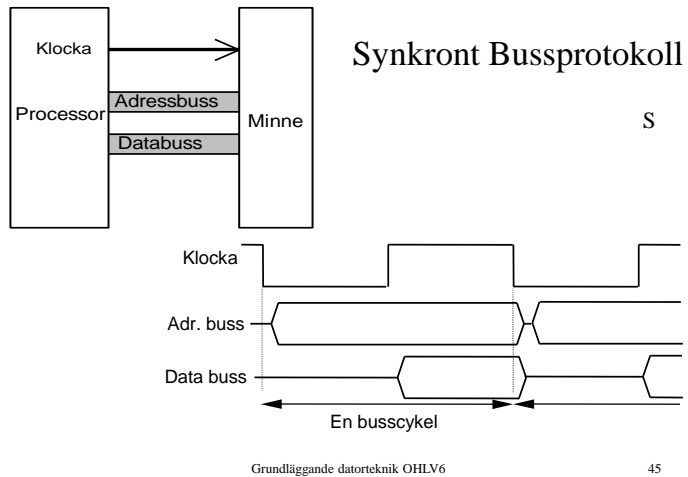


Grundläggande dator teknik OHLV6

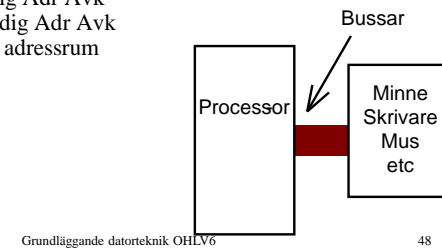
### Multiplexad Buss



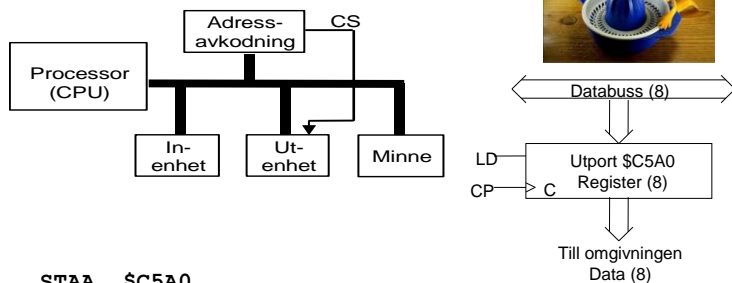
Grundläggande dator teknik OHLV6



- LV6 Fo15**
- Dagens mål: Du ska kunna....**
- ▶ Skilja på olika minnestyper
  - ▶ Förklara principer för olika bussprotokoll
  - ▶ Förstå begreppen Timing och VM (Valid Memory Address)
  - ▶ **Konstruera adressavkodningslogik ....**
    - ▶ 1) Fullständig Adr Avk
    - ▶ 2) Ofullständig Adr Avk
  - ▶ Processorns adressrum



## Adressavkodningslogik för en utport (Chip Select logik; CS-logik)



STAA \$C5A0  
STAA UTPORTADDRESS

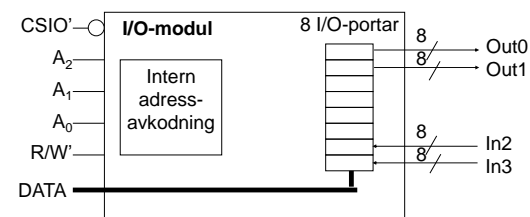
Grundläggande datorteknik OHLV6

49

Ut0 på adress \$C5A0  
Ut1 på adress \$C5A1  
Ut2 på adress \$C5A2  
Ut3 på adress \$C5A3  
In0 på adress \$C5A4  
In1 på adress \$C5A5  
In2 på adress \$C5A6  
In3 på adress \$C5A7

## Adressavkodning

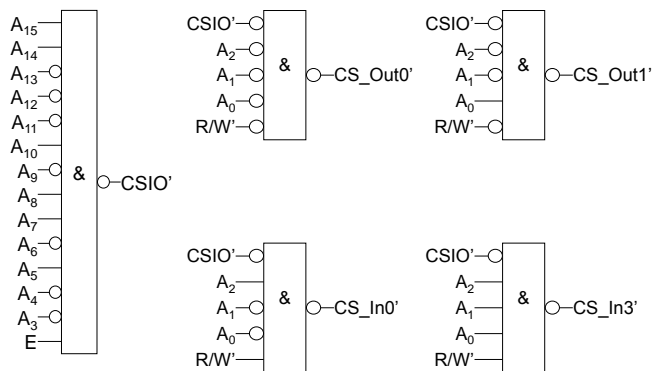
Om vi behöver flera I/O-portar ???



Grundläggande datorteknik OHLV6

50

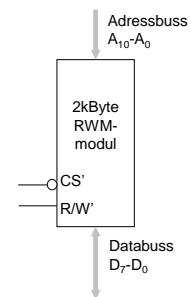
## Adressavkodning



Grundläggande datorteknik OHLV6

51

## Att ansluta en 2-kbyte RWM- modul med startadress \$4000



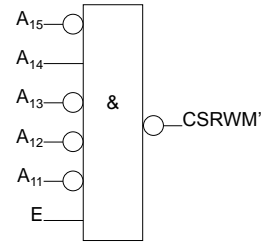
### Arbetsgång:

- "Tolka" beskrivningen av minnesmodulen
- Rita tabell
- Ange modulens första adress
- Ange modulens sista adress
- Märk ut konstanta resp varierande adressledningar
- Rita adressavkodningslogiken

Grundläggande datorteknik OHLV6

52

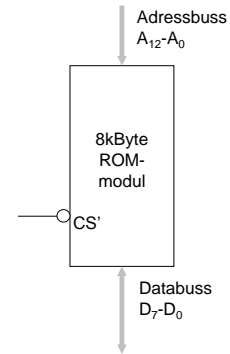
## Att ansluta en 2-kbyte RWM-modul



Grundläggande datorteknik OHLV6

53

## Att ansluta en 8-kbyte ROM-modul

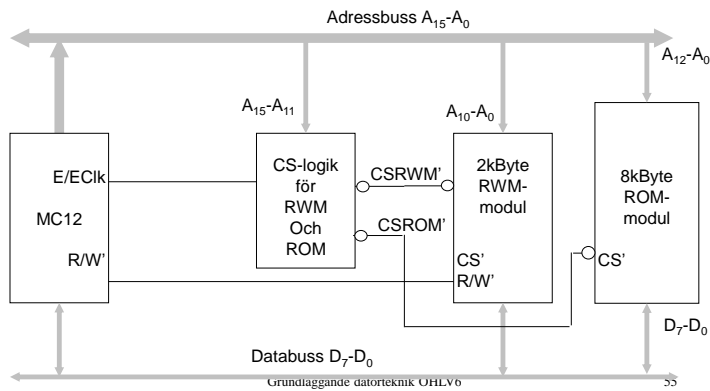


Skall placeras på de högsta adresserna (\$FFFF)

Grundläggande datorteknik OHLV6

54

## Att ansluta en 8-kbyte ROM-modul till ett befintligt system

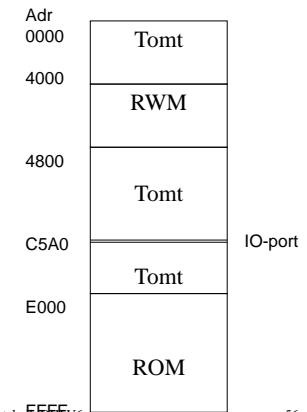


Grundläggande datorteknik OHLV6

55

## Processorns adressrum

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



Grundläggande datorteknik OHLV6

56

## Uppgift

Konstruera adressavkodningen för följande:

- 4 kByte RWM från adress 0
- 8 kByte ROM på de högsta adresserna
- En I/O-area på 256 Byte med start på adress \$6000

Du har tillgång till

- 8 kByte ROM-modul
- 4 kByte RWM-modul

Använd fullständig adressavkodning

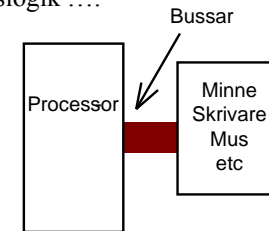
Grundläggande datorteknik OHLV6

57

LV6 Fo15

## Dagens mål: Du ska kunna....

- ▶ Skilja på olika minnestyper
- ▶ Förklara principer för olika bussprotokoll
- ▶ Beskriva MC12's bussar (anslutningar)
- ▶ Förstå begreppen Timing och VM (Valid Memory Address)
- ▶ Konstruera adressavkodningslogik ....
  - ▶ 1) Fullständig Adr Avk
  - ▶ **2) Ofullständig Adr Avk**
  - ▶ Processorns adressrum



Grundläggande datorteknik OHLV6

58

## Uppgift

Som förra.... men använd ofullständig adressavkodning!

Konstruera adressavkodningen för följande:

- 4 kByte RWM från adress 0
- 8 kByte ROM på de högsta adresserna
- En I/O-area på 256 Byte med start på adress \$6000

Du har tillgång till

- 8 kByte ROM-modul
- 4 kByte RWM-modul

Grundläggande datorteknik OHLV6

59

## Uppgift

Konstruera adressavkodningen för följande:

- En I/O-area på 512 Byte med start på adress \$0000
- RWM från I/O-areans slut till adress \$1FFF
- ROM från adress \$2000-\$FFFF

Du har tillgång till

- 64 kByte ROM-modul
- 8 kByte RWM-modul

Grundläggande datorteknik OHLV6

60