

Digital- och datorteknik

Välkommen!

Kursens mål:

Fatta hur en dator är uppbyggd (HDW)
Fatta hur du programmerar den (SW)
Fatta hur HDW o SW samverkar

... och därmed kunna använda en
modern μ -processor / μ -controller

Digital teknik

Dator teknik

Lärandemål:

1. Talsystem, binära koder och datoraritmetik
2. Switchnätalgebra
3. Kombinatoriska nät
4. Sekvensnät
5. Datorkonstruktion (von Neumann-datorn)

Studieportalen

Lärandemål:

1) Talsystem, binära koder och datoraritmetik

1. Konvertering mellan olika talsystem
2. Utifrån given problemställning applicera binära koder så som NBC, NBCD, ASCII, Gray, Excess, felupptäckande koder, tecken/belopp och komplementkoder.
3. Redogöra för och tillämpa binär aritmetik (addition och subtraktion).

Talsystem
Basen 2, 8, 10, 16

Positionssystem
Ex: 214
(421)

$\beta = 2$	$\beta = 8$	$\beta = 10$	$\beta = 16$
binärt	oktalt	decimalt	hexa-decimalt
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6
111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F
10000	20	16	10
10001	21	17	11

Digital och Datorteknik fo1

S2.4

5

GRAY-KOD

Tabell 2.2. Graykoder.

S2.16

Decimal ordning	Kodord i trebitars Graykod	Kodord i fyrbitsars Graykod
0	000	0000
1	001	0001
2	011	0011
3	010	0010
4	110	0110
5	111	0111
6	101	0101
7	100	0100
8		1100
9		1101
10		1111
11		1110
12		1010
13		1011
14		1001
15		1000

Digital och Datorteknik fo18

6

NBCD-kod

Skriv (563,782)
på NBCD-kod.

Decimal siffra	NBCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Digital och Datorteknik fo18

S2.19

7

Alfanumeriska koder

ASCII-koden.	0	1	2	3	4	5	6	7	$b_7b_6b_5$	$b_4b_3b_2b_1b_0$
NUL	DLE	SP	@	P	'	p			0	0000
SOH	DC1	!	1	A	Q	a	q		0	0001
STX	DC2	"	2	B	R	b	r		0	0010
ETX	DC3	#	3	C	S	c	s		0	0011
EOT	DC4	\$	4	D	T	d	t		0	1000
ENQ	NAK	%	5	E	U	e	u		0	1001
ACK	SYN	&	6	F	V	f	v		0	1100
BEL	ETB	'	7	G	W	g	w		0	1101
BS	CAN	(8	H	X	h	x		1	0000
HT	EM)	9	I	Y	i	y		1	0001
LF	SUB	*	:	J	Z	j	z		1	0100
VT	ESC	+	;	K	[k	{	ä	1	0101
FF	FS	,	<	L	\	l	ö		1	1000
CR	GS	-	=	M]	m	}	å	1	1001
S0	RS	.	>	N	^	n	~		1	1100
S1	US	/	?	O	_	o			1	1101
										1111
										RUBOUT (DEL)

S2.29

0
1
2
3
4
5
6
7
8
9
A
B
C
D
E
F

Ex
"A" = 41₁₆

8

Tabell 2.1. Excess- 2^{n-1} kodning vid $n = 4$.

Excess-kod

Ex avkoda:

1 1 0 0₂ Excess- 2^{n-1}

$n=4$

$$2^3 + 2^2 + 0 + 0 - 2^3 = 4_{10}$$

Nivå k Φ	Kodord i excess- 2^{n-1} -kod ($n=4$)
-8 Φ	0000
-7 Φ	0001
-6 Φ	0010
-5 Φ	0011
-4 Φ	0100
-3 Φ	0101
-2 Φ	0110
-1 Φ	0111
0	1000
1 Φ	1001
2 Φ	1010
3 Φ	1011
4 Φ	1100
5 Φ	1101
6 Φ	1110
7 Φ	1111

Digital och Dator teknik fo1

S2.15

Def 2-Komplement:

Arb s 36-39

Pos: $Y = Y$

Neg: $(-Y) = 2^n - |Y| = Y_{2K}$

Att tvåkomplementera:

Ex 4 bit: $2^n = 2^4 = 16$

$$2^n - Y = 2^n - 1 - Y + 1 = 16 - 1 - Y + 1 = 15 - Y + 1 (= Y_{1K} + 1)$$

Ex $Y=6$: 0110 Hitta $(-Y)$

$$\begin{array}{r} 15_{10} \\ - Y \\ \hline \end{array}$$

$$\begin{array}{r} 1111 \\ - 0110 \\ \hline \end{array}$$

$$= Y_{1K} = 1001$$

INVERSEN! Def 1komp! Y_{1K}

$$\text{addera } 1 \quad +0001$$

$$= Y_{2K} = 1010$$

$$Y_{1K} + 1 = Y_{2K}$$

$$(-Y = 1010)$$

$$(-6 = 1010)$$

Att subtrahera:

$$X - Y = X + Y_{2K} = X + Y_{1K} + 1$$

Digital och Dator teknik fo18

10

Lärandemål:

2) Switchnätalgebra

1. Definiera grundläggande logiska operationer och dess motsvarande logiska grindar.
2. Tillämpa den booleska algebrans räknelagar.
3. Utföra algebraisk förenkling av booleska uttryck.
4. Visa likhet/olikhet mellan booleska uttryck.

Digital och Dator teknik fo18

11

Sammanfattning

•Boolesk algebra

$1+0=1$; $1 \cdot 0=0$; $1+1=1$; $1 \cdot 1=1$; $1'=0$; $0'=1$
de Morgans lagar

•Funktionstabell

Sätt upp tabell med alla kombinationer av invariabler

•Binär evaluering

Gör kolumner för deluttrycken

Digital och Dator teknik fo18

12

Rep: Satser inom Boolesk algebra.

S3.8

1. Kommutativa lagarna

$$x + y = y + x$$

$$x \cdot y = y \cdot x$$

2. Distributiva lagarna

$$x \cdot (y + z) = x \cdot y + x \cdot z$$

$$x + (y \cdot z) = (x + y) \cdot (x + z)$$

7. Associativa lagarna

$$x + (y + z) = (x + y) + z$$

$$x \cdot (y \cdot z) = (x \cdot y) \cdot z$$

8. De Morgans lagar

$$(x + y)' = x' \cdot y'$$

$$(x \cdot y)' = x' + y'$$

$$3. \quad x + 0 = x$$

$$x \cdot 1 = x$$

$$4. \quad x + x' = 1$$

$$x \cdot x' = 0$$

$$5. \quad x + 1 = 1$$

$$x \cdot 0 = 0$$

$$6. \quad x + x = x$$

$$x \cdot x = x$$

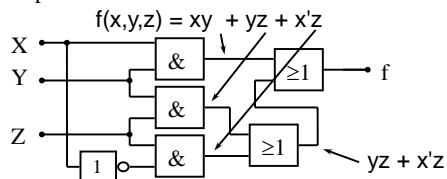
$$9. \quad (x')' = x$$

Lärandemål:

3) Kombinatoriska nät

1. Realisera logiska uttryck med grindnät.
2. Beskriva, analysera och konstruera kombinatoriska nät med hjälp av funktionstabeller och boolesk algebra.
3. Kunna minimera logiska uttryck för realisering i kombinatoriska nät.

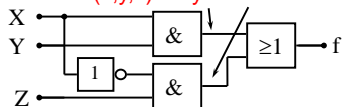
Grindnät för exempel Kalle



Disjunktiv (minimal) form

(Summa av Produkter)

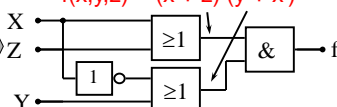
$$f(x,y,z) = xy + x'z$$



Konjunktiv (minimal) form

(Produkt av summor)

$$f(x,y,z) = (x + z)(y + x')$$

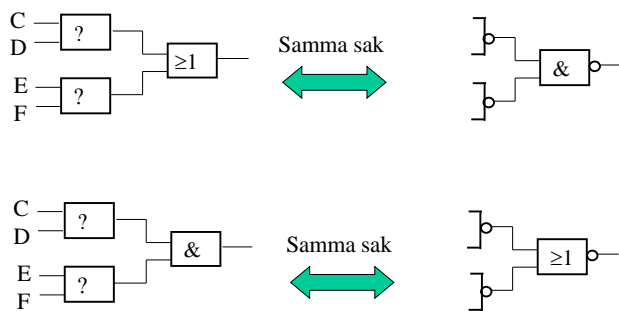


Sammanfattning

- NORMAL form \Rightarrow Funktionstabell
- MINIMAL form \Rightarrow Kranaughdiagram
- DISJUNKTIV (*normal / minimal*) form
 - Σ av Prod Ex: $(x'y) + (xw) + (xyw)$
 - Ettor
 - Mintermer: $(1 \cdot 1 \cdot 1) = 1$
 - NAND / NAND - logik
- KONJUNKTIV (*normal / minimal*) form
 - Prod av Σ :or Ex: $(x+z)(x'+z+w)(z'+w')$
 - Nollor
 - Maxtermer: $(0+0+0) = 0$
 - NOR / NOR - logik

DSP E eller LEEDS P

Sammanfattning NAND- och NOR-logik

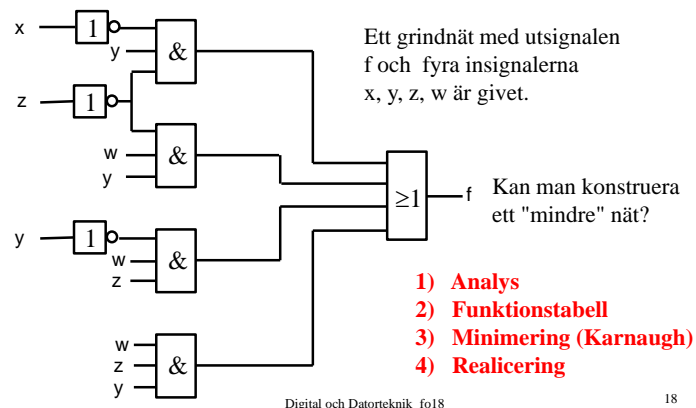


Digital och Dator teknik fo18

17

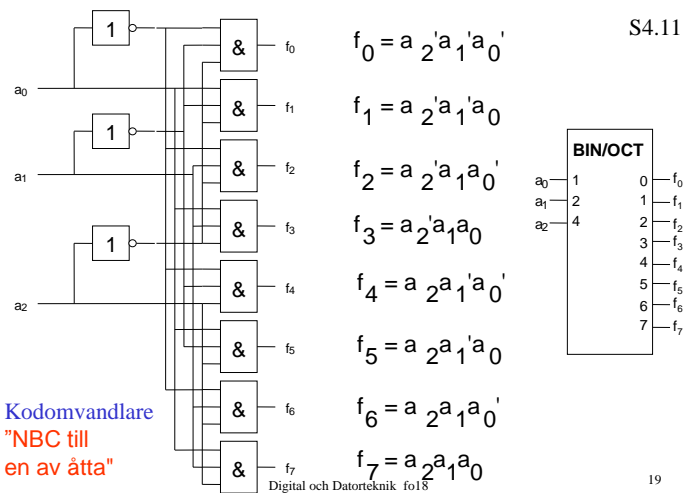
Praktikfall, minimering av grindnät

Ext4



Digital och Dator teknik fo18

18



Digital och Dator teknik fo18

19

En delmängd av veckans mål:

Fo4

- ▶ Konstruera de olika kombinatoriska nät som ingår i en dator. Exempel på sådana nät är väljare, kodomvandlare och ALU (beräkningsenheten i processorn).
- ▶ Studera hur addition/subtraktion utförs

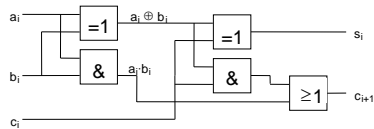
Dagens mål:

- ▶ Kodomvandlare (en kod IN → annan kod UT)
- ▶ Don't care – termer (ger färre grindar)
- ▶ Väljare (många signaler IN + styrsig → en signal UT)
- ▶ Fördelare (en signal IN + styrsig → många signaler UT)
- ▶ Heladderare (adderar $x+y+c_{in}=s_{ut}$ och c_{ut})
- ▶ Koda tal (2-komplementsrepresentationen)

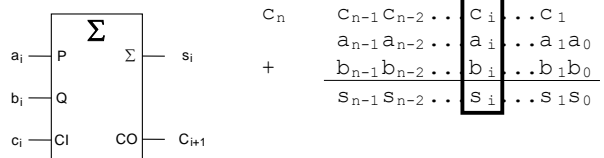
Digital och Dator teknik fo18

20

**Läs minst en
lär dig mer!**



Prosamsymbolen för en heladderare.



Def Flaggor

Statusflaggor ut från ALU:n som indikerar om resultatet blev rätt eller fel

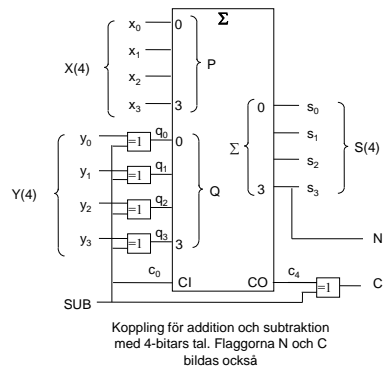
- C** Carry Tal utan tecken [0,15] (ADD : minnessiffr; SUB: lånesiffr)
- V** Overflow Tal med tecken [-8,7]
- N** Negative Tal med tecken [-8,7]
- Z** Zero Tal med och utan tecken

- C=1:** Resultatet av operationen blev *fel* för en operation *utan tecken*
- V=1:** Resultatet av operationen blev *fel* för en operation *med tecken*
- N=1:** Resultatet av operationen blev *negativt* för en operation *med tecken*
- Z=1:** Resultatet av operationen blev *noll*

Lärandemål:

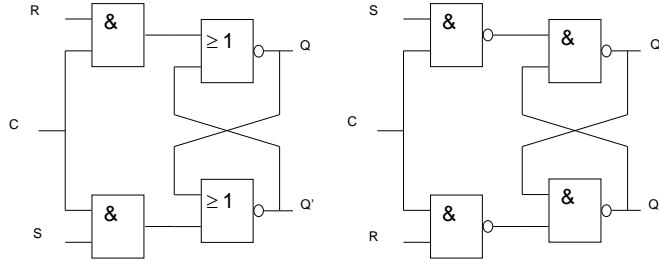
4) Sekvensnät

1. Analysera och konstruera synkrona tillståndsmaskiner med hjälp av tillståndstabeller och tillståndsgrafer.
2. Använda D-, T- och JK- vippor för konstruktion av minneselement och räknare.

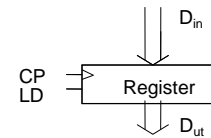
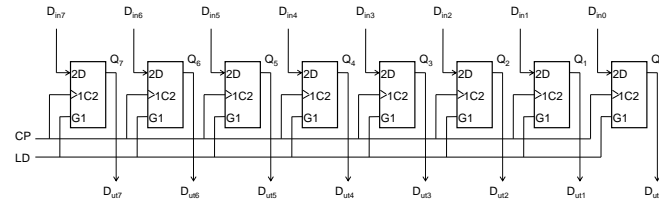


Grindad SR-latch

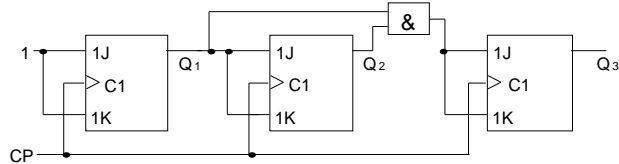
Ofta förses SR-latchar med en tredje ingång, till vilken en **styrpuls C** ansluts
 Härvid erhålls en s.k **grindad SR-latch** .



Register och bussar

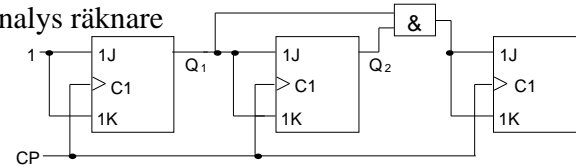


Arbetsgång - analys räknare s5.25 ->



- 1 Studera kopplingen och **bestäm vippornas insignaler** ($T_1=, T_2=, T_3=$)
- 2 Sätt upp en tabell med
 - "**Detta tillstånd**" (Alla kombinationer av Q_1, Q_2, Q_3)
 - **Insignaler** (T_1, T_2, T_3)
 - "**Nästa tillstånd**" (Q_1^+, Q_2^+, Q_3^+)
- 3 Ange insignalernas värden i tabellen och----
- 4 ange vad "nästa tillstånd" blir
- 5 Rita slutligen en **tillståndsgraf**

Analys räknare



- 1) $T_1=1$
- 2) $T_2= Q_1$
- 3) $T_3= Q_1 Q_2$

Funktionstabell

T	Q^+
0	Q
1	Q'

Detta Tillstånd			Insignaler			Nästa Tillstånd		
Q_3	Q_2	Q_1	T_3	T_2	T_1	Q_3^+	Q_2^+	Q_1^+
0	0	0	0	0	1	0	0	1
0	0	1	0	1	1	0	1	0
0	1	0	0	0	1	0	1	1
0	1	1	1	1	1	0	0	0
1	0	0	0	0	1	1	0	1
1	0	1	0	1	1	1	1	0
1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0

Arbetsgång - syntes räknare

Konstruera en räknare som räknar sekvensen ????

Utsignaler				Insignaler							
Detta tillstånd Q		Nästa tillstånd Q'		J ₃ K ₃		J ₂ K ₂		J ₁ K ₁		J ₀ K ₀	
q ₃	q ₂	q ₁	q ₀	q ₃ '	q ₂ '	q ₁ '	q ₀ '				
0	0	0	0								
0	0	0	1								
0	0	1	0								
0	0	1	1								
0	1	0									
0	1										
0	1	1									
0	1	1	1								
1	0	1	1								
1	1	0	0								
1	1	0	1								
1	1	1	0								
1	1	1	1								

Alfa kombinationer av "Detta Tillstånd"

Fyll i "Nästa Tillstånd"



- 1 Rita en **tillståndsgraf**
- 2 Sätt upp en **tabell** med:
 - "Detta tillstånd" (Alla kombinationer av Q₁, Q₂, Q₃)
 - "Nästa tillstånd" (Q₁', Q₂', Q₃')
 - Vippornas **Insignaler**
- 3 Ange "**Nästa tillstånd**" i tabellen
- 4 Använd vippornas excitationstabell och ange **vippornas insignaler**
- 5 **Minimera** uttrycken för insignalerna
- 6 **Realisera** räknaren

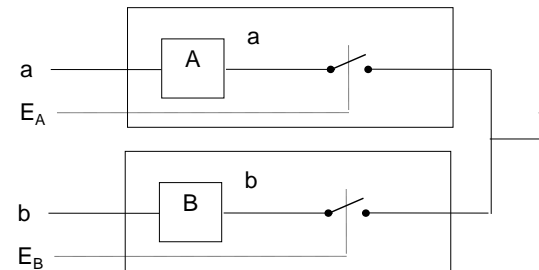
Lärandemål:

Arb s67

5) Datorkonstruktion (von Neumann-datorn)

1. Beskriva, analysera och konstruera kombinatoriska och sekventiella nät som typiskt används för att bygga en dators centralenhet, dvs. dataväg, styrenhet, aritmetisk/logisk enhet (Arithmetic/Logical Unit).
2. Beskriva in-/ut- matningsenheter och minnessystem tillsammans med centralenheten.
3. Beskriva, analysera och konstruera en styrenhet baserad på fast kopplad logik och kunna implementera instruktions-exekvering i denna logik.
4. Kunna redogöra för styrenheter med mikroprogrammerad logik.
5. Kunna utföra elementär maskinnära programmering (maskinprogrammering och assemblerprogrammering).

"Three-State"



Aktivera "Program och minne"

Instruktionsformat

LDAA Adr



INCA



Maskinprogram

00001111₂
 00001011₂
 00111111₂
 11111110₂
 00011001₂
 01000001₂
 01001010₂

Maskinprogram

0F₁₆
 0B₁₆
 3F₁₆
 FE₁₆
 19₁₆
 41₁₆
 48₁₆

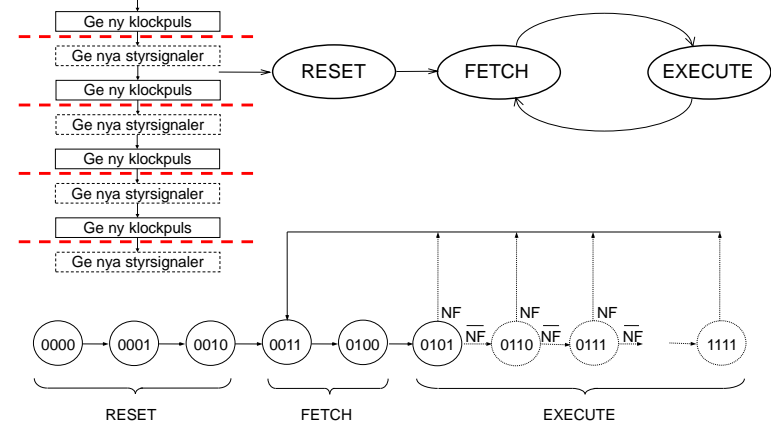
Mask. prog
i minnet

Adr:

Adr	Mask. prog	Tillhörande assemblerprog
0C	10	LDAB #\$23
0D	23	
0E	29	ADDB \$F3
0F	F3	
10	02	TFR B,A
11	4F	CMPB #\$03
12	03	
13	61	BLO \$29
14	13	

Arb s 92

Aktivera "Processorns arbetssätt"



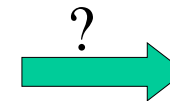
Instruktionslistan för FLEX

och för CPU12

Handassemblering

Assemblerprogram

CLRA
 NEGB
 SBCB \$0B
 LDAA #\$43



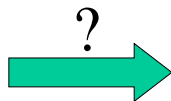
Maskinprogram

Adr	Minne
00h	\$47
01h	\$39
02h	\$35
03h	\$0B
.	.
.	.
A6h	\$28
A7h	\$23
.	.
.	.

Disassemblering

Maskinprogram

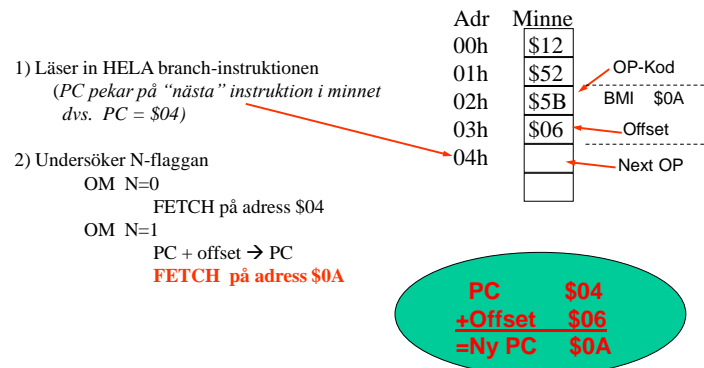
Adr	Minne
00h	\$47
01h	\$39
02h	\$35
03h	\$0B
.	.
.	.
A6h	\$28
A7h	\$23
.	.
.	.



Assemblerprogram

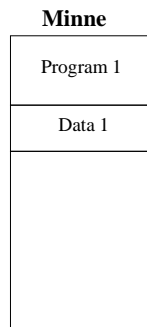
```
CLRA
NEGB
SBCB $0B
LDAA #$43
```

Vad gör processorn vid BMI ?



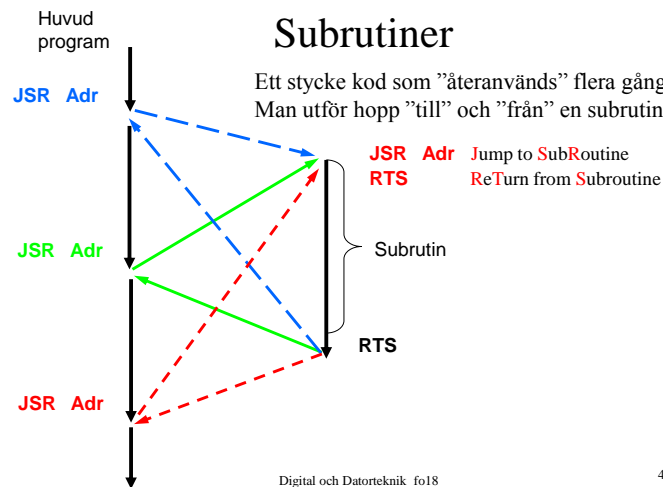
Subrutin o stack

- Stack:** Ett minnesutrymme
- Stackpekare:** Register S
Pekar på översta elementet på stacken
- Subrutin:** Ett stycke kod, avslutat med instruktionen RTS
- JSR:** Hopp TILL subrutin
- RTS:** Återhopp FRÅN subrutin



Subrutiner

Ett stycke kod som "återanvänds" flera gånger.
Man utför hopp "till" och "från" en subrutin



```

Program1.fsr
    org    $10    Definerar programs
    lda    #$12

Loop    inca
        sta    temp
        bra    Loop

temp    rmb    1    Definerar en varia
    
```

10		1.	org	\$10	Definerar programs
10	0F 12	3.	lda	#\$12	
		4.			
12	41	5.	Loop	inca	
13	13 17	6.		sta	temp
15	5A FB	7.		bra	Loop
		8.			
18	00	9.	temp	rmb	1
					Definerar en varia

KÄLLFIL

LISTFIL

Maskinkod

Adresser

Assemblerprogram

Digital och Dator teknik fo18 45

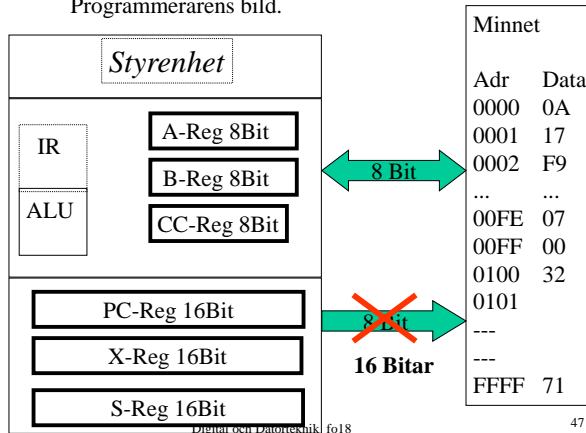
Assemblerdirektiv

[symbol]	FCB	uttryck[,uttryck[,...]]	(Form Constant Byte)
[symbol]	FDB	uttryck[,uttryck[,...]]	(Form Double Byte)
[symbol]	FCS	"teckensträng"	(Form Constant String)
[symbol]	RMB	uttryck	(Reserve Memory Bytes)
[symbol]	ORG	uttryck	(Origin)
symbol	EQU	uttryck	(Equate)

Studera nu Upg 118

FLEX CPU12

Programmerarens bild.

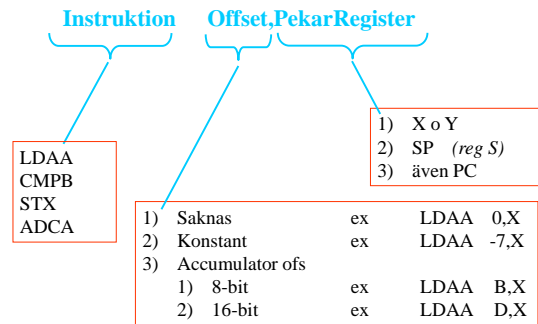


Adresseringsmoder CPU12

- Inherent INH
- Immediate IMM
- Direct (Page) DIR
- Extended EXT
- Relative REL
- Indexed IDX/IDX1/IDX2

Adresseringsmod: Indexed IDX, IDX1, IDX2

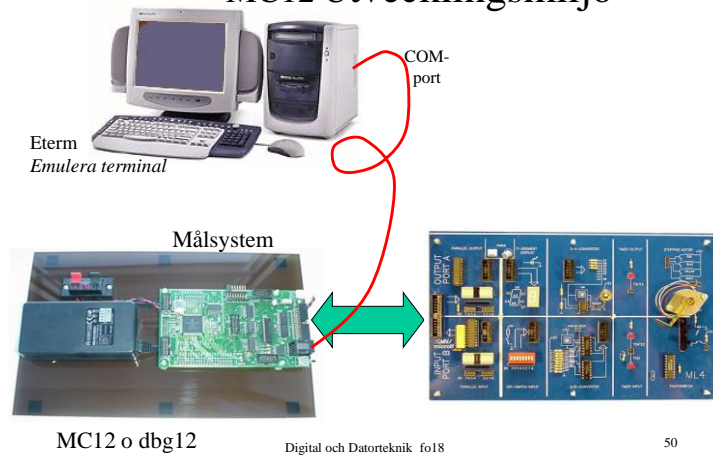
Ex) LDAA 3,X



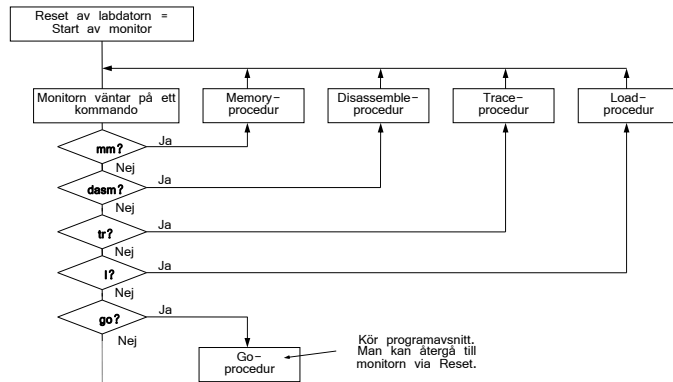
Digital och Dator teknik fo18

49

MC12 Utvecklingsmiljö



MC12 – dbg12



Digital och Dator teknik fo18

51