

Aktivera Kursens mål:

LV5 Fo12

- ▶ Konstruera en dator mha grindar och programmera denna
- ▶ Använda en modern microcontroller

Aktivera Förra veckans mål:

- ▶ Konstruera styrenheten.... genom att....
- ▶ implementera olika maskininstruktioner i styrenheten.
- ▶ Kunna använda instruktionslistan och skriva mycket enkla assemblerprogram
- ▶ Studera olika instruktionstyper och adresseringsmoder
- ▶ Använda utvecklingsmiljön för FLEX

Veckans mål:

- ▶ Konstruera styrenheten.... genom att....
- ▶ implementera olika maskininstruktioner i styrenheten.
- ▶ Villkorliga hopp
- ▶ Subrutiner och stack
- ▶ Skriva enkla program för FLEX
- ▶ Introduktion av CPU12

Digital o Datorteknik OHLV5

**Läs smart!
Lär dig mer!**

LV5 Fo12

Dagens mål: Du ska kunna.....

- ▶ **Förstå villkorliga hopp i program**
- ▶ Implementera BEQ-instruktionen i styrenheten.
- ▶ Använda villkorliga hoppinstruktioner
- ▶ Förstå begreppen stack, stackpekare och stackinstruktioner
- ▶ Implementera PSH-instruktionen i styrenheten.
- ▶ Förstå användningen av subrutiner
- ▶ Skriva subrutiner
- ▶ "Programmera i FLEX-miljön"

Digital o Datorteknik OHLV5

Fokus på...**Villrorliga (Relativa) hopp - forts**

Arb s 140

Instruktionsformat
BRA Adr

OP-kod	Offset
--------	--------

Minnes Adress	Instruktioner i minnet
k	Maskininstruktion i
k+1	Maskininstruktion i+1
k+2	Maskininstruktion i+2
k+3	Maskininstruktion i+2
k+4	Maskininstruktion i+3
k+5	Maskininstruktion i+3
k+6	Maskininstruktion i+4
k+7	Maskininstruktion i+4
k+8	Maskininstruktion i+5
k+9	Maskininstruktion i+6
k+A	Maskininstruktion i+6

RTN-beskrivning:**PC+Offset → PC**

Digital o Datorteknik OHLV5

Villkorliga hopp:

Om PIN-koden är korrekt given
- öppna telefonen
Annars
- stäng telefonen

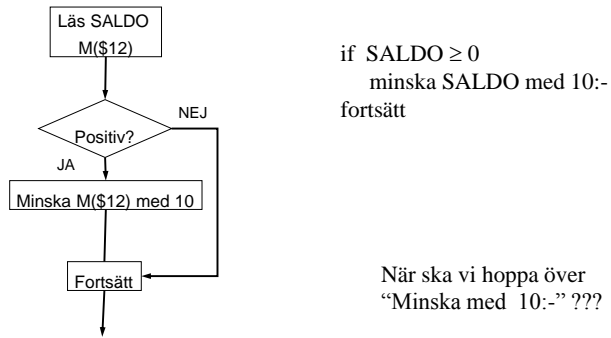
...någon addition...
Om C=1
- så hoppa till felrutin
Annars
- fortsatt beräkningen

1) Om vi inte bearbetat alla dataord
- bearbeta nästa dataord
- börja om från 1)
2) Annars fortsatt med annat arbete

Vid villkorliga hopp används **relativa hoppinstruktioner**

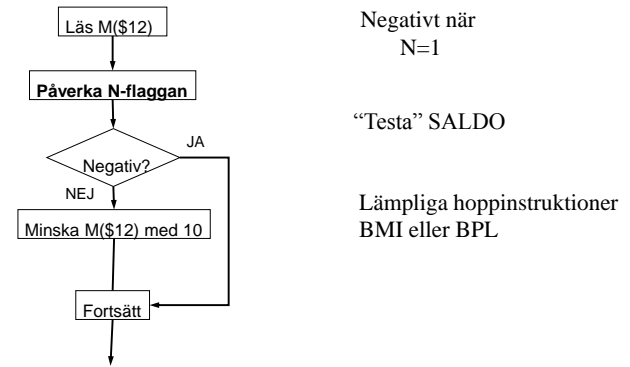
Digital o Datorteknik OHLV5

Villkorliga Hopp -Instruktioner *Branch instruktioner*



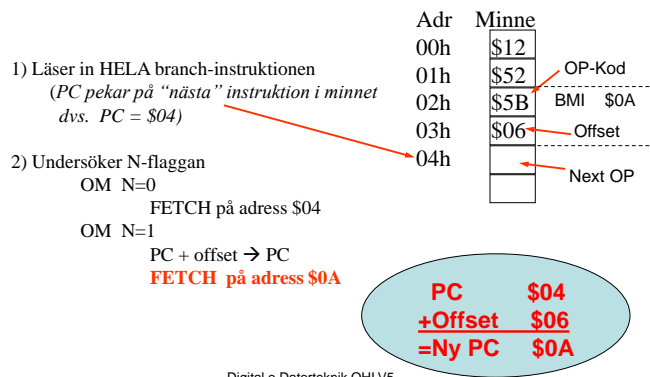
Digital o Datorteknik OHLV5

Villkorliga hopp -Instruktioner



Digital o Datorteknik OHLV5

Vad gör processorn vid BMI ?



Digital o Datorteknik OHLV5

LV5 Fo12

Dagens mål: Du ska kunna.....

- ▶ Förstå villkorliga hopp i program
- ▶ **Implementera BEQ-instruktionen i styrenheten.**
- ▶ Använda villkorliga hoppinstruktioner
- ▶ Förstå begreppen stack, stackpekare och stackinstruktioner
- ▶ Implementera PSH-instruktionen i styrenheten.
- ▶ Förstå användningen av subrutiner
- ▶ Skriva subrutiner
- ▶ ”Programmera i FLEX-miljön”

Fokus på...

Digital o Datorteknik OHLV5

LV5 Fo12

Relativa villkorliga hopp – Upg 115 ^{Arb s 144}

Om Z = 1

PC+Offset → PC, FETCH

Annars

FETCH

Testa först i

FLEX-simulatorn

Adress
(Hex)

20
21
22
23
24
25
26
27

LDA #\$FF

INCA

BEQ \$22

BRA \$20

Digital o Datorteknik OHLV5

Dagens mål: Du ska kunna.....

- ▶ Förstå villkorliga hopp i program
- ▶ Implementera BEQ-instruktionen i styrenheten.
- ▶ **Använda villkorliga hoppinstruktioner**
- ▶ Förstå begreppen stack, stackpekare och stackinstruktioner
- ▶ Implementera PSH-instruktionen i styrenheten.
- ▶ Förstå användningen av subrutiner
- ▶ Skriva subrutiner

- ▶ "Programmera i FLEX-miljön"

Fokus på...

Digital o Datorteknik OHLV5

Villkorliga hopp

Ext 9

Instruktionsuppsättningen för FLEX-processorn har ett antal villkorliga hoppinstruktioner.

De kan indelas i följande tre grupper:

1. Enkla hoppvillkor.
2. Hoppvillkor för tal utan inbyggt tecken.
3. Hoppvillkor för tal med inbyggt tecken.
(2-komplementrepresentation)

Digital o Datorteknik OHLV5

Villkorliga hopp - forts

Ext 9

1. Enkla hoppvillkor.

Vid de enkla villkorliga hoppen testas innehållet i en av flaggvipporna N, Z, V eller C och hoppet utförs om villkoret är uppfyllt, dvs den aktuella flaggvippans värde, är 0 resp 1.

Digital o Datorteknik OHLV5

Villkorliga hopp - forts

Ext 9

2. Hoppvillkor för tal utan inbyggt tecken.

Förutsätt att flaggorna har påverkats av en subtraktion $X - Y$ enligt:

LDA	XVALUE	Läs X från minnet till A
CMPA	#Y	Låt skillnaden $X - Y$ påverka flaggorna
B(Villkor)	Hoppadress	Utför hoppet om villkoret är uppfyllt

X och Y är 8-bitars tal som tillhör intervallet $[0, 255]$.

Digital o Dator teknik OHLV5

Villkorliga hopp - forts

Ext 9

2. Hoppvillkor för tal utan inbyggt tecken. Flaggor C och Z

$X > Y$, $X \geq Y$, $X = Y$, $X \neq Y$, $X \leq Y$ och $X < Y$.

Relation	Villkorlig hoppinstruktion	Hoppvillkor
$X > Y$	BHI (Branch if X is higher than Y)	$C \bullet Z'$
$X \geq Y$	BHS (Branch if X is higher or same as Y)	C'
$X = Y$	BEQ (Branch if X is equal to Y)	Z
$X \neq Y$	BNE (Branch if X is not equal to Y)	Z'
$X \leq Y$	BLS (Branch if X is lower or same as Y)	$(C \bullet Z)' = C + Z$
$X < Y$	BLO (Branch if X is lower than Y)	C

Digital o Dator teknik OHLV5

Villkorliga hopp - forts

Ext 9

3. Hoppvillkor för tal med inbyggt tecken. (2-komplementstal)

Förutsätt att flaggorna har påverkats av en subtraktion $X - Y$ enligt:

LDA	XVALUE	Läs X från minnet till A
CMPA	#Y	Låt skillnaden $X - Y$ påverka flaggorna
B(Villkor)	Hoppadress	Utför hoppet om villkoret är uppfyllt

X och Y är 8-bitars tal som tillhör intervallet $[-128, 127]$.

Digital o Dator teknik OHLV5

Villkorliga hopp - forts

Ext 9

3. Hoppvillkor för tal med inbyggt tecken.

X och Y är 8-bitars tal som tillhör intervallet $[-128, 127]$.

Flaggor N, V och Z

$X > Y$, $X \geq Y$, $X = Y$, $X \neq Y$, $X \leq Y$ och $X < Y$.

Relation	Villkorlig hoppinstruktion	Hoppvillkor HV
$X > Y$	BGT (Branch if X is greater than Y)	$(N \oplus V)' \bullet Z'$
$X \geq Y$	BGE (Branch if X is greater than or equal to Y)	$(N \oplus V)'$
$X = Y$	BEQ (Branch if X is equal to Y)	Z
$X \neq Y$	BNE (Branch if X is not equal to Y)	Z'
$X \leq Y$	BLE (Branch if X is less than or equal to Y)	$((N \oplus V)' \bullet Z)' = (N \oplus V) + Z$
$X < Y$	BLT (Branch if X is less than Y)	$N \oplus V$

Digital o Dator teknik OHLV5

Uppgift

Två variabler P och Q är lagrade i minnet.
Jämför talen och skriv det största till variabeln R.

P är placerad på adress 20_{16} i minnet.
Q är placerad på adress 21_{16} i minnet.
R är placerad på adress 22_{16} i minnet.

Digital o Dator teknik OHLV5

LV5 Fo12

Dagens mål: Du ska kunna.....

- ▶ Förstå villkorliga hopp i program
 - ▶ Implementera BEQ-instruktionen i styrenheten.
 - ▶ Använda villkorliga hoppinstruktioner
 - ▶ **Förstå begreppen stack, stackpekare och stackinstruktioner**
 - ▶ Implementera PSH-instruktionen i styrenheten.
 - ▶ Förstå användningen av subrutiner
 - ▶ Skriva subrutiner
- ▶ "Programmera i FLEX-miljön"

FOKUS PÅ...

Digital o Dator teknik OHLV5

Exempel IN- o UT-matning

Skriv ett program som hela tiden läser inporten,
om b_6 av inporten noll, skriv 7 till utporten
annars skriv 22 till utporten

Använd ML4 In/Ut

Programmets startadress: \$60

Digital o Dator teknik OHLV5

STACK och STACKPEKARE Arb s 137

STACK: Ett minnesutrymme
Används för att lagra temporära
data (registerinnehåll och
återhopsadresser)

STACKPEKARE: Ett register (**Reg S**)
som pekar på det senast ditlagda

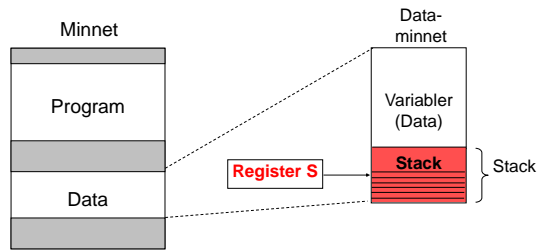
INSTRUKTIONER:

PSH: Placera ett registerinnehåll **PÅ** stacken
PUL: Hämta **FRÅN** stacken **TILL** ett register

Digital o Dator teknik OHLV5

Stacken - forts

Arb s 137



Stacken: ett minnesutrymme som vi temporärt utnyttjar

Digital o Datorteknik OHLV5

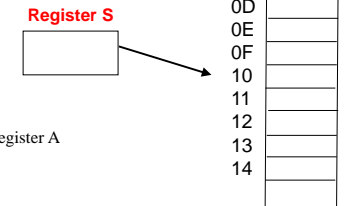
Stacken – forts – några instruktioner Arb s 137

PSHA.

Innehållet i register A skrivs till stacken (till minnet).

RTN-beskrivningen:

- 1) $S-1 \rightarrow S$
- 2) $A \rightarrow M(S)$

**PULA.**

Hämta ett dataord från stacken till register A

RTN-beskrivningen:

- 1) $M(S) \rightarrow A$
- 2) $S+1 \rightarrow S$

Digital o Datorteknik OHLV5

LV5 Fo12

Uppgift

Definiera en stack som börjar på adress $7F_{16}$.

Placera sedan följande på stacken:

$3B_{16}$, 12_{16} , 66_{16} och $F8_{16}$.

Använd register B

Digital o Datorteknik OHLV5

Dagens mål: Du ska kunna.....

- ▶ Förstå villkorliga hopp i program
 - ▶ Implementera BEQ-instruktionen i styrenheten.
 - ▶ Använda villkorliga hoppinstruktioner
 - ▶ Förstå begreppen stack, stackpekare och stackinstruktioner
 - ▶ **Implementera PSH-instruktionen i styrenheten.**
 - ▶ Förstå användningen av subrutiner
 - ▶ Skriva subrutiner
- ▶ "Programmera i FLEX-miljön"

Fokus på...

Digital o Datorteknik OHLV5

LV5 Fo12

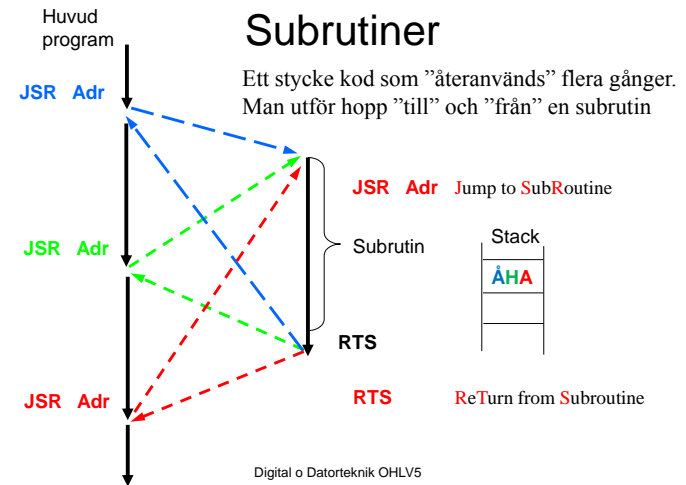
Dagens mål: Du ska kunna.....

- ▶ Förstå villkorliga hopp i program
- ▶ Implementera BEQ-instruktionen i styrenheten.
- ▶ Använda villkorliga hoppinstruktioner
- ▶ Förstå begreppen stack, stackpekare och stackinstruktioner
- ▶ Implementera PSH-instruktionen i styrenheten.
- ▶ **Förstå användningen av subrutiner**
- ▶ Skriva subrutiner

- ▶ "Programmera i FLEX-miljön"

FOKUS PÅ...

Digital o Datorteknik OHLV5



LV5 Fo12

Dagens mål: Du ska kunna.....

- ▶ Förstå villkorliga hopp i program
- ▶ Implementera BEQ-instruktionen i styrenheten.
- ▶ Använda villkorliga hoppinstruktioner
- ▶ Förstå begreppen stack, stackpekare och stackinstruktioner
- ▶ Implementera PSH-instruktionen i styrenheten.
- ▶ Förstå användningen av subrutiner
- ▶ **Skriva subrutiner**

- ▶ "Programmera i FLEX-miljön"

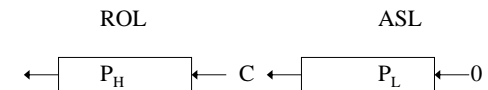
FOKUS PÅ...

Digital o Datorteknik OHLV5

Subrutin o stack - forts

Multiplitera en 16-bitars variabel på adress \$20 med två.

Adr	Minne		
20	P _H	3	0011 ←
21	P _L	6	0110 ←
		12	1100 ←



Digital o Datorteknik OHLV5

LV5 Fo12

Adr	Minne	Assembler prog		
\$80	\$3D	ASL	\$21	2*P _L
\$81	\$21			
\$82	\$40	ROL	\$20	2*P _H
\$83	\$20			
\$84	\$6A	RTS		
\$85	\$69	JSR	\$80	Mul2
\$86	\$80			
\$87	\$69	JSR	\$80	Mul2
\$88	\$80			
\$89	6A	RTS		
\$8A				

Digital o Datorteknik OHLV5

Dagens mål: Du ska kunna.....

- ▶ Förstå villkorliga hopp i program
- ▶ Implementera BEQ-instruktionen i styrenheten.
- ▶ Använda villkorliga hoppinstruktioner
- ▶ Förstå begreppen stack, stackpekare och stackinstruktioner
- ▶ Implementera PSH-instruktionen i styrenheten.
- ▶ Förstå användningen av subrutiner
- ▶ Skriva subrutiner

- ▶ "Programmera i FLEX-miljön"

Fokus på...

Digital o Datorteknik OHLV5

LV5 Fo13

Veckans mål:

- ▶ Konstruera styrenheten.... genom att....
- ▶ implementera olika maskinstruktioner i styrenheten.
- ▶ Villkorliga hopp
- ▶ Subrutiner och stack
- ▶ Skriva enkla program för FLEX
- ▶ Introduktion av CPU12

Dagens mål: Du ska kunna....

- ▶ Beskriva likheter o olikheter mellan FLEX och CPU12
- ▶ Använda Instruktionslistan för CPU12
 - ▶ Instruktionsgrupper
 - ▶ Adresseringsmoder
- ▶ Skriva enkla program för CPU12
- ▶ Använda delar av utvecklingsmiljön Eterm för MC12

Digital o Datorteknik OHLV5

**Läs smart!
Lär dig mer!**

Fo13

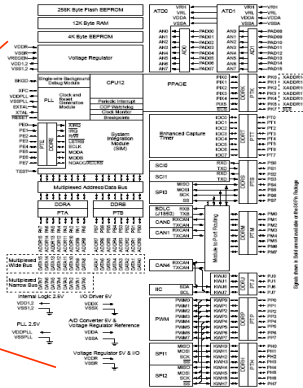
Dagens mål: Du ska kunna....

- ▶ **Beskriva likheter o olikheter mellan FLEX och CPU12**
- ▶ Använda Instruktionslistan för CPU12
 - ▶ Instruktionsgrupper
 - ▶ Adresseringsmoder
- ▶ Skriva enkla program för CPU12
- ▶ Använda delar av utvecklingsmiljön Eterm för MC12

Digital o Datorteknik OHLV5

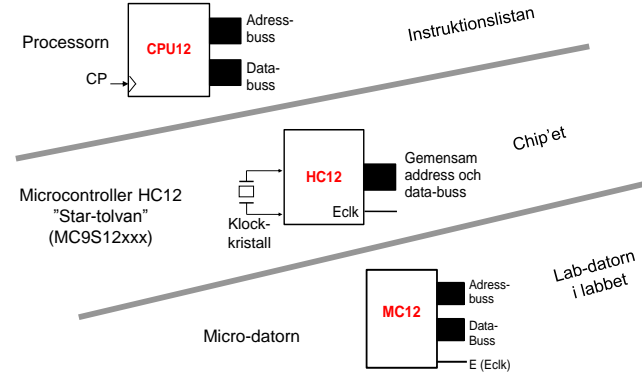
**Läs smart!
Lär dig mer!**

CPU12 / HC12 / MC12



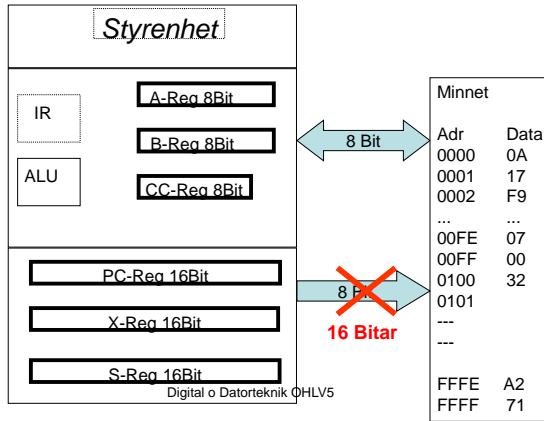
Digital o Datorteknik OHLV5

CPU12 / HC12 / MC12



Digital o Datorteknik OHLV5

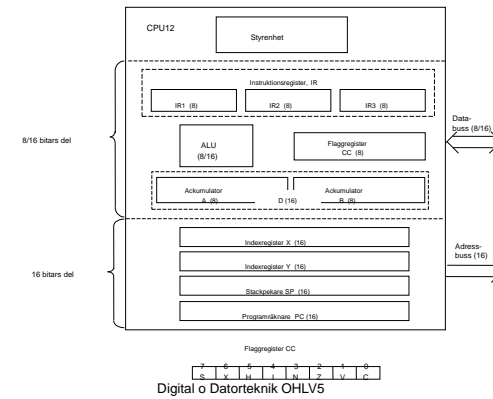
~~Flex~~ **CPU12**
Programmerarens bild.



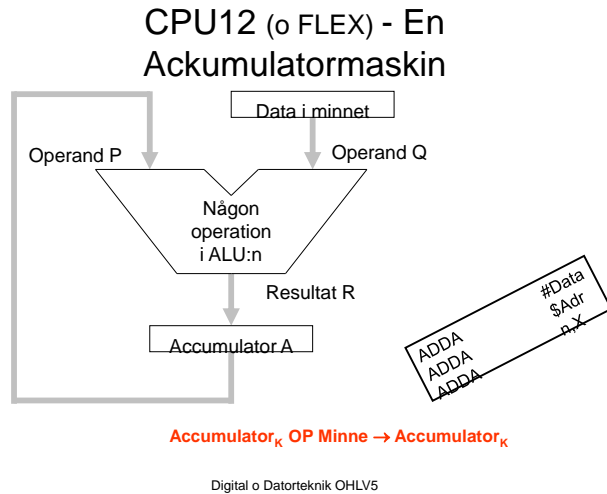
Digital o Datorteknik OHLV5

Inst s 0-1

Programmerarens bild CPU12



Digital o Datorteknik OHLV5



FLEX:
en 1-adress –maskin

Flytta data i minnet:

LDAA \$Adr1
STAA \$Adr2

Stoppa in data i minnet:

LDAA #Data
STAA \$Adr

Flytta data (byte) i minnet:

MOVB \$Adr1,\$Adr2

Stoppa in data (byte) i minnet:

MOVB #Data,\$Adr

Även 16-bitars: MOVW \$Adr1,\$Adr2

Digital o Datorteknik OHLV5

CPU12:
Även
2-adress
maskin

Fo13

Dagens mål: Du ska kunna....

- ▶ Beskriva likheter o olikheter mellan FLEX och CPU12
- ▶ **Använda Instruktionslistan för CPU12**
 - ▶ Instruktionsgrupper
 - ▶ Adresseringsmoder
- ▶ Skriva enkla program för CPU12
- ▶ Använda delar av utvecklingsmiljön Eterm för MC12

**Läs smart!
Lär dig mer!**

Digital o Datorteknik OHLV5

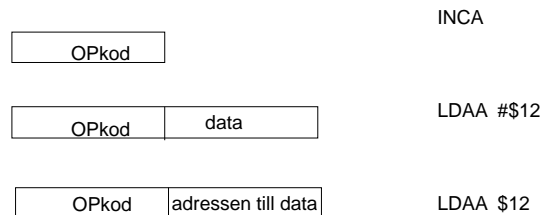
Instruktionsrepertoar

s 7

- **Instruktioner för dataflyttning:**
 - Flytta data (Kopiera data)
- **Instruktioner för aritmetik:**
 - Utföra aritmetiska operationer (DAA, MUL, DIV)
- **Instruktioner för logiska operationer:**
 - AND, OR, XOR, komplementering (invertering).
- **Instruktioner för jämförelse och test**
 - Jämföra innehållet i ett register med ett minnesinnehåll
- **Hoppinstruktioner**
 - BEQ, LBEQ, DBNE, BRSET/CLR
- **Övriga instruktioner:**
 - Instruktioner för att styra processorns funktion och arbetssätt

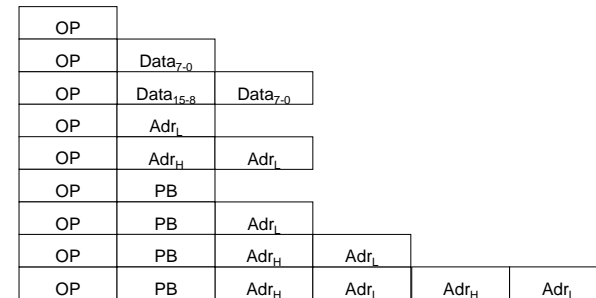
Digital o Datorteknik OHLV5

Instruktionsformat FLEX



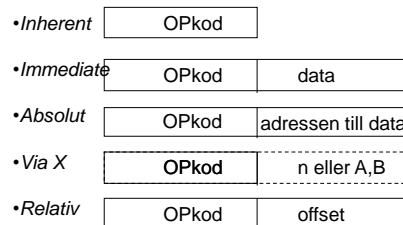
Digital o Datorteknik OHLV5

Instruktionsformat CPU12



Digital o Datorteknik OHLV5

Adresseringsmoder FLEX



Digital o Datorteknik OHLV5

Inst s 4

Adresseringsmoder CPU12

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

Digital o Datorteknik OHLV5

Adresseringsmod: **Inherent** INH

Assembler	Maskinkod	RTN
INCA	42	A + 1 → A
NEGB	50	B' + 1 → B

*ENDAST OP-kod
(Operanden är
inbyggd i OP-
koden)*

Digital o Datorteknik OHLV5

Adresseringsmod: **Immediate** IMM

Assembler	Maskinkod	RTN
LDAB #3F	C6 3F	3F → B
LDD #9AB2	CC 9A B2	9AB2 → D
LDY #1234	CD 12 34	1234 → Y
LDY #1	CD 00 01	0001 → Y

OP Dataoperand

Digital o Datorteknik OHLV5

Adresseringsmod: **Extended** EXT

Assembler	Maskinkod	RTN
LDAB \$1234	F6 12 34	M(1234) → B
LDD \$9AB2	FC 9A B2	M(9AB2):M(9AB3) → D
LDY \$1234	FD 12 34	M(1234):M(1235) → Y
LDY \$1	FD 00 01	M(0001):M(0002) → Y

OP Adressoperand

Digital o Datorteknik OHLV5

Adresseringsmod: **Relative** REL

Assembler	Maskinkod	RTN
BRA Loop	20 ofs	PC+ofs → PC
BEQ \$9AB2	27 ofs	if Z=1: PC+ofs → PC
		<i>OP offset (1 byte) → [-128,127]</i>
LBRA Stop	18 20 ofsH ofsL	PC+ofsH:ofsL → PC
LBEQ Snurra	18 27 ofsH ofsL	if Z=1: PC+ofsH:ofsL → PC
		<i>OP offset (2 byte) → [-32768,32767]</i>

Digital o Datorteknik OHLV5

Adresseringsmod: **Indexed pre/post inc/dec**

Assembler		RTN:
LDAA	2,X+	1) $M(X) \rightarrow A$ 2) $X+2 \rightarrow X$
LDAA	4,+X	1) $X+4 \rightarrow X$ 2) $M(X) \rightarrow A$
LDAA	8,-X	1) $X-8 \rightarrow X$ 2) $M(X) \rightarrow A$
LDAA	1,X-	1) $M(X) \rightarrow A$ 2) $X-1 \rightarrow X$

Digital o Datorteknik OHLV5

Uppgift

Skriv en instruktionssekvens för FLEX-processorn som nollställer bit 3-0 i alla minnesord i adressintervallet $[35_{16}, 39_{16}]$.

Använd X-registret för adressering.

Digital o Datorteknik OHLV5

Fo13

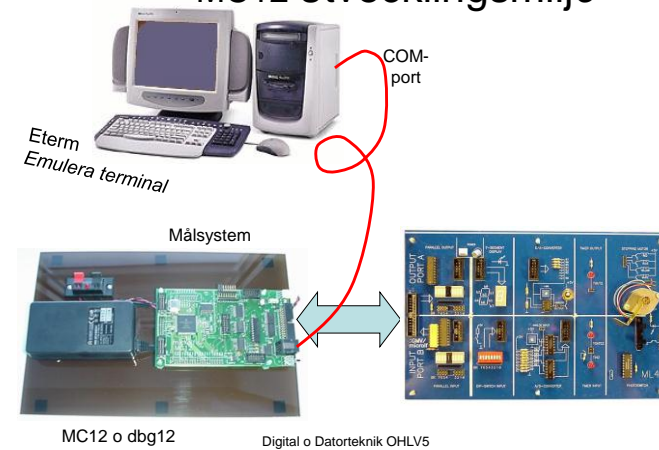
Dagens mål: Du ska kunna....

- ▶ Beskriva likheter o olikheter mellan FLEX och CPU12
- ▶ Använda Instruktionslistan för CPU12
 - ▶ Instruktionsgrupper
 - ▶ Adresseringsmoder
- ▶ **Skriva enkla program för CPU12**
- ▶ **Använda delar av utvecklingsmiljön Eterm för MC12**

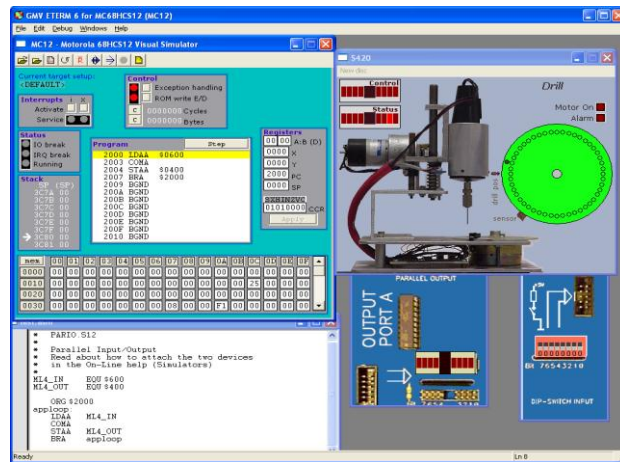
**Läs smart!
Lär dig mer!**

Digital o Datorteknik OHLV5

MC12 Utvecklingsmiljö



Digital o Datorteknik OHLV5



Digital o Datorteknik OHLV5

Räkna antal ettställda bitar i Register A (DipSwitch)
Skriv antalet (register B) till HexDisplay

Digital o Datorteknik OHLV5