

Aktivera Kursens mål:

- ▶ Konstruera en dator mha grindar och programmera denna

LV3 Fo7

Aktivera Förra veckans mål:

- ▶ Konstruera olika kombinatoriska nät som ingår i en dator.
- ▶ Studera hur addition/subtraktion utförs (+flaggor)
- ▶ Konstruera register som används för att lagra data i datorn.
- ▶ Koppla samman register med bussar till en enkel dataväg

Veckans mål:

- ▶ Koppla samman register och ALU till en dataväg
- ▶ Förstå hur ett minne är uppbyggt och ansluta detta till datavägen
- ▶ Program och hur detta lagras i minne
- ▶ Fatta hur datorn startar och arbetar
- ▶ Räkna och mera vippor

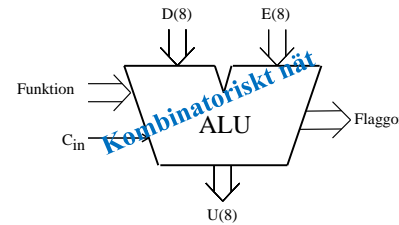
Dagens mål, Du ska kunna:

- ▶ Konstruera och använda en enkel dataväg
- ▶ Kunna programmera en enkel dataväg (RTN)
- ▶ Förstå uppbyggnaden av ett minne
- ▶ Använda en enkel dataväg med minne
- ▶ Förstå von Neumanns princip med program och minne
- ▶ Ansluta CC-register till datavägen

**Läs smart!
Lär dig mer!**

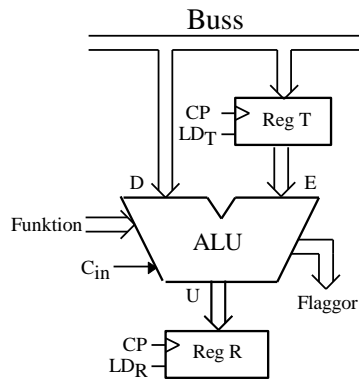
ALU:n ska anslutas – hur då?

Arb s 199

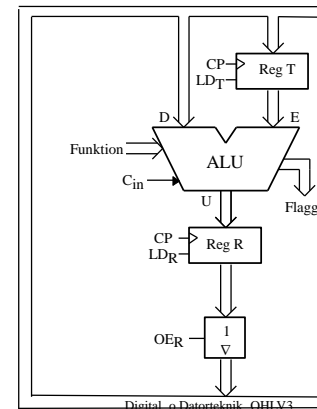


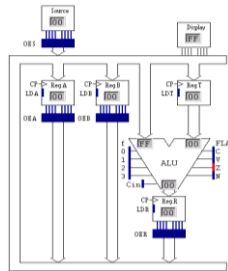
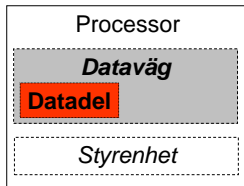
$f_3 f_2 f_1 f_0$	$U = f(D, E, F, C_{in})$	
	Operation	Resultat
0 0 0 0	Bitvis nollställning	0
0 0 0 1		D
0 0 1 0		E
0 0 1 1	Bitvis invertering	D_{ix}
0 1 0 0	Bitvis invertering	E_{ix}
0 1 0 1	Bitvis OR	D OR E
0 1 1 0	Bitvis AND	D AND E
0 1 1 1	Bitvis XOR	D XOR E
1 0 0 0	$D + 0 + C_{in}$	$D + C_{in}$
1 0 0 1	$D + FFH + C_{in}$	$D - 1 + C_{in}$
1 0 1 0	$D + D + C_{in}$	$D + E + C_{in}$
1 0 1 1	$D + D + C_{in}$	$2D + C_{in}$
1 1 0 0	$D + E_{ix} + C_{in}$	$D - E - 1 + C_{in}$
1 1 0 1	Bitvis nollställning	0
1 1 1 0	Bitvis nollställning	0
1 1 1 1	Bitvis ettställning	FFH

Användning av resultatregister (R) för lagring av utdata från ALU.



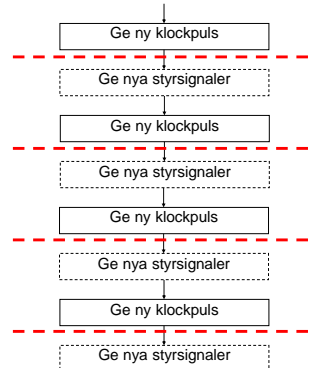
Anslutning av resultatregister (R) till buss.





Digital o Datorteknik OHLV3

Enkel dataväg Arb s 74



Fokus på...

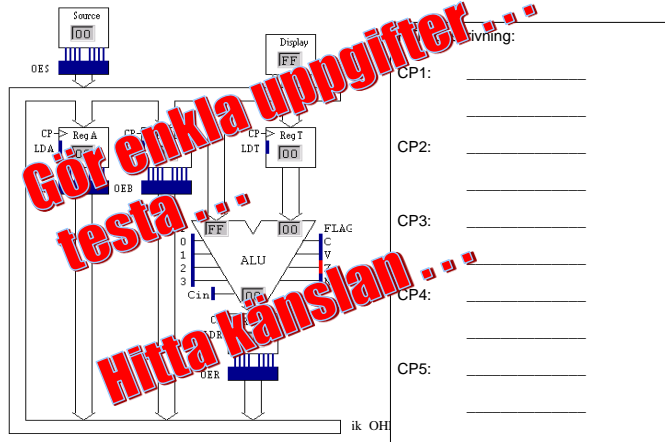
Fo7

Dagens mål:

- ▶ Konstruera och använda en enkel dataväg
- ▶ **Kunna programmera en enkel dataväg (RTN)**
- ▶ Förstå uppbyggnaden av ett minne
- ▶ Använda en enkel dataväg med minne
- ▶ Förstå von Neumanns princip med program och minne
- ▶ Ansluta CC-register till datavägen

**Läs smart!
Lär dig mer!**

Ge en sekvens av styrsignaler som utför:



Fokus på...

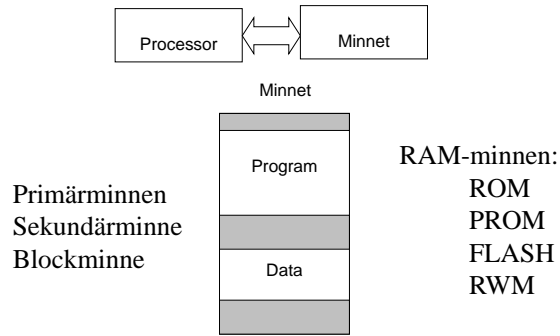
Fo7

Dagens mål:

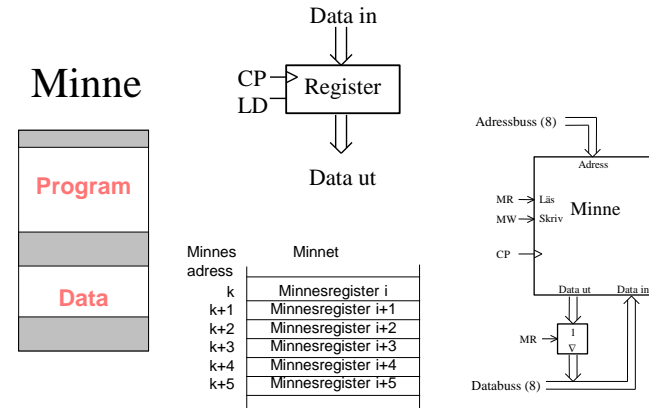
- ▶ Konstruera och använda en enkel dataväg
- ▶ Kunna programmera en enkel dataväg (RTN)
- ▶ **Förstå uppbyggnaden av ett minne**
- ▶ Använda en enkel dataväg med minne
- ▶ Förstå von Neumanns princip med program och minne
- ▶ Ansluta CC-register till datavägen

**Läs smart!
Lär dig mer!**

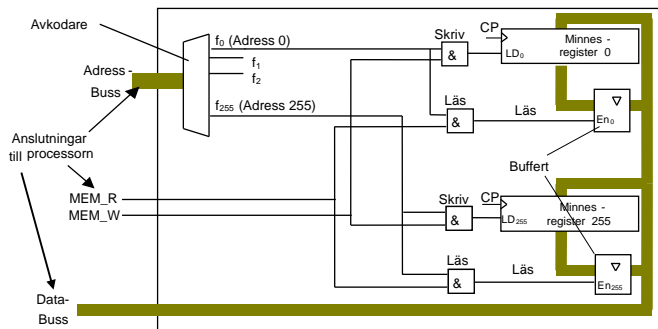
Minnet



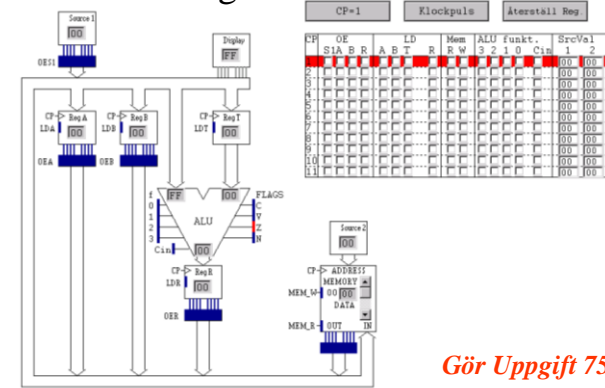
Minne



Minns Man Minnet?



Enkel dataväg med RWM



Gör Uppgift 75

FOKUS PÅ...

Fo7

Dagens mål:

- ▶ Konstruera och använda en enkel dataväg
- ▶ Kunna programmera en enkel dataväg (RTN)
- ▶ Förstå uppbyggnaden av ett minne
- ▶ Använda en enkel dataväg med minne
- ▶ Förstå von Neumans princip med program och minne
- ▶ Ansluta CC-register till datavägen

**Läs smart!
Lär dig mer!**

Arb s 92

Program och minne

- **John Louis Von Neumann** (1903-1957)

”Det lagrade programmets princip”, dvs program och data i samma minne.



adress hex	innehåll hex	operation
07	01	
08	08	- load the A-register with the contents of this address (01H)
0A	28	- add to the A-register the contents of this address (02H)
0C	61	- branch, if carry=1 to the address = this number (07H) + address of next OPCODE (0EH) (= 15H)
0E	13	- store the contents of the A-register on this address (04H)
10	47	- clear the A-register
11	13	- store the contents of acc A (00H) on this address (03H)
12	03	- jump to this address (03H)
13	59	- jump to this address (18H)
14	18	this address (18H)
16	94	- store the contents of the A-register on this address (04H)
17	0F	- load the A-register with this number (01H)
19	13	- store contents of the A-register (01H) on this address (03H)
1A	93	on this address (03H)
1R	0R	

Maskinprogram i minnet	Tillhörande assemblerprogram		
0C	10	LDAB #S23	Instruktion
0D	23		
0E	28	ADDB \$F3	Adress
0F	F3		
10	02	TFR B,A	Data
11	4F	CMPB #S03	
12	03		
13	61	BLO \$29	
14	13		

Arb s 92

Program o Minne - forts

Instruktionsformat

INCA

OP-kod

LDAA Adr

OP-kod Adr

Maskinprogram

00001111₂
00001011₂
00111111₂
11111110₂
00011001₂
01000001₂
01001010₂

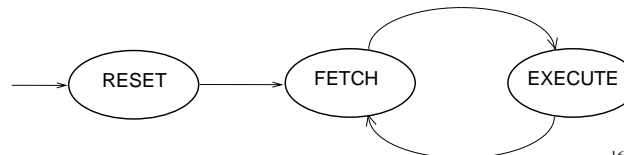
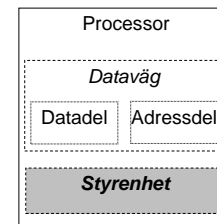
Maskinprogram

0F₁₆
0B₁₆
3F₁₆
FE₁₆
19₁₆
41₁₆
4A₁₆

Processorns arbetssätt

Arb s 95

Maskinprogram i minnet	Tillhörande assemblerprogram	
Adr		
0C	10	LDAB #S23
0D	23	
0E	28	ADDB \$F3
0F	F3	
10	02	TFR B,A
11	4F	CMPB #S03
12	03	
13	61	BLO \$29
14	13	



FOKUS PÅ...

Fo7

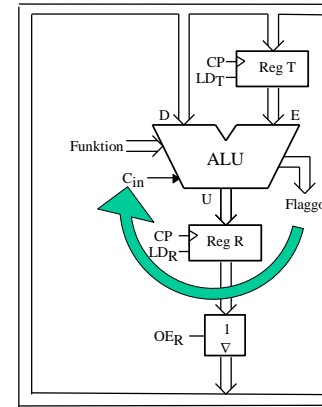
Dagens mål:

- ▶ Konstruera och använda en enkel dataväg
- ▶ Kunna programmera en enkel dataväg (RTN)
- ▶ Förstå uppbyggnaden av ett minne
- ▶ Använda en enkel dataväg med minne
- ▶ Förstå von Neumanns princip med program och minne
- ▶ Ansluta CC-register till datavägen

**Läs smart!
Lär dig mer!**

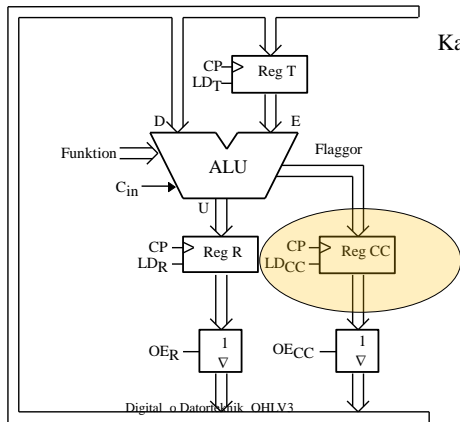
Fånga C-flaggan.

Kap 7 Blå



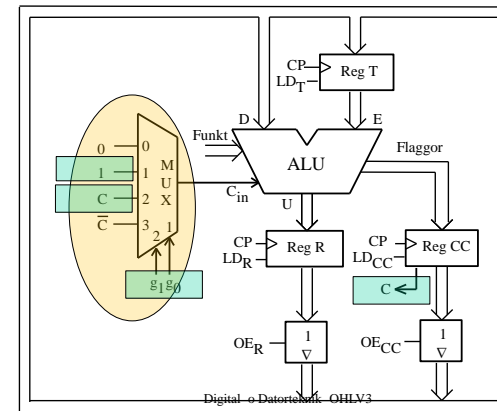
Inkoppling av flaggregister mellan ALU och buss.

Kap 7 Blå

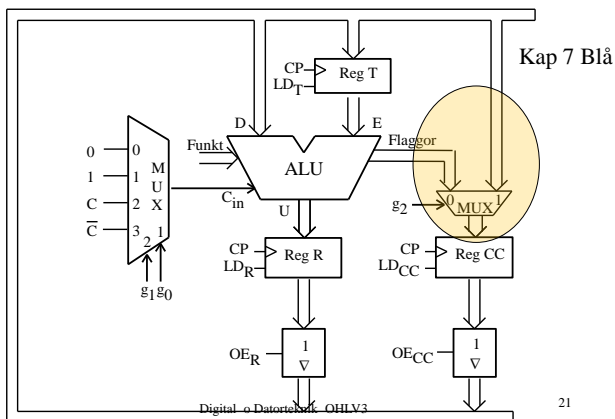


Inkoppling av väljare (multiplexer) för val av Cin.

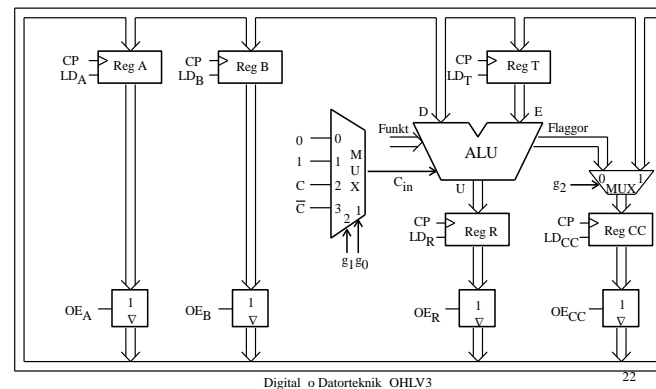
Kap 7 Blå



Inkoppling av väljare (mux) för val av indata till CC-registret.



Logiknät för databehandling med aritmetik/logikenhet (ALU).



Veckans mål:

- Koppla samman register och ALU till en dataväg
- Förstå hur ett minne är uppbyggd, ansluta detta till datavägen
- Program och hur detta lagras i minne
- Fatta hur datorn startar och arbetar
- Räknare och mera vippor

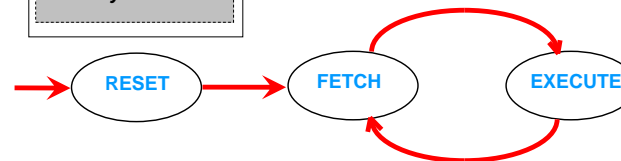
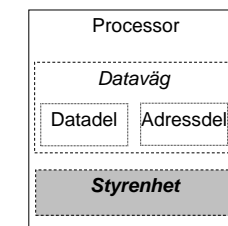
LV3 Fo8

Dagens mål. Du ska kunna:

- ▶ Beskriva Processorns Arbetsätt
 - ▶ Ange styrsignalsekvens för RESET
 - ▶ Ange styrsignalsekvens för FETCH
 - ▶ Ange styrsignalsekvenser för olika EXECUTE
- ▶ Konstruera och använda JK- och T-vippor
- ▶ Kunna analysera räknare
- ▶ Ta fram Excitationstabeller

**Läs smart!
Lär dig mer!**

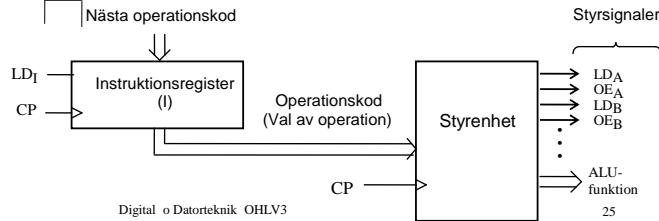
Processorns arbetsätt



Maskinprogram i minnet	Tillhörande Arb s 95 assemblerprogram
0C 10	LDB # \$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	

Maskinprogram
i minnet

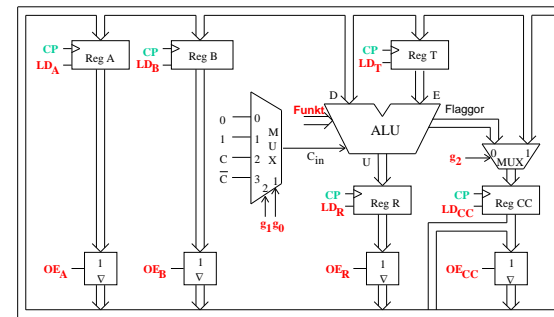
Adn			
0C	10	LDB	#\$23
0D	23		
0E	29	ADDB	\$F3
0F	F3		
10	02	TFR	B,A
11	4F	CMPB	#\$03
12	03		
13	61	BLO	\$29
14	13		



Digital o Dator teknik OHLV3

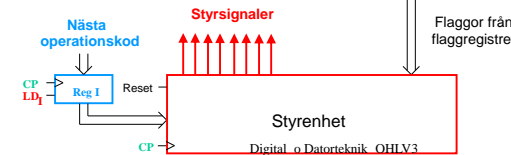
Kap 7 Blå

Styrsignalgenerering för
den databehandlande
enheten.



Kap 7 Blå

Styrenhet
och del av
dataväg



26

FOKUS PÅ...

Fo8

Dagens mål. Du ska kunna:

- ▶ Beskriva Processorns Arbetssätt
 - ▶ Ange styrsignalsekvens för RESET
 - ▶ Ange styrsignalsekvens för FETCH
 - ▶ Ange styrsignalsekvenser för olika EXECUTE-faser
- ▶ Konstruera och använda JK- och T-vippa
- ▶ Kunna analysera räknare
- ▶ Ta fram Excitationstabeller

**Läs smart!
Lär dig mer!**

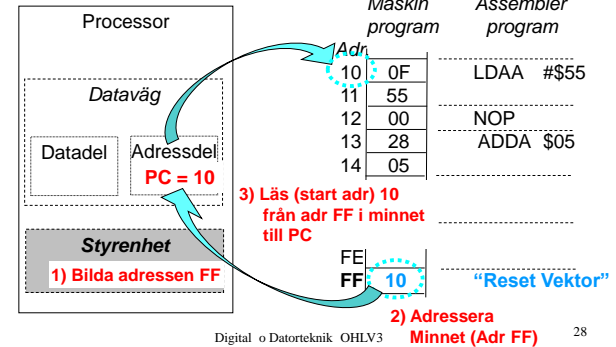
Digital o Dator teknik OHLV3

27

Processorns arbetssätt - **RESET**

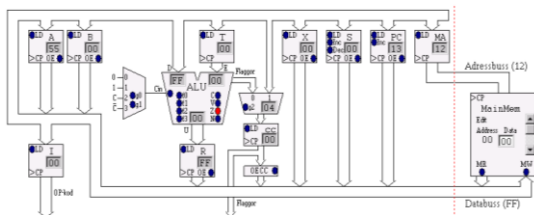
Arb s 95

PC: Programräknare
(Pekar på nästa instruktion)



Digital o Dator teknik OHLV3

28



State	RTN-beskr	Styrtsignaler	Kommentar
0	$FF_{16} \rightarrow R$	$ALU-fkn = F_0$ $LD_R=1$	ALU-funktionen väljs så att talet FF finns på ALU:ns utgång. Laddning på R-registret ettsätts så att utvärdet från ALU'n (FF_{16}) laddas i R-registret vid nästa klockpuls.
1	$R \rightarrow MA$	$OE_R=1$ $LD_{MA}=1$	Talet FF_{16} i R-registret kopplas ut på bussen. Bussinnehållet laddas i minnesadressregistret vid nästa klockpuls.
2	$M \rightarrow PC$	$MR=1$, $LD_{PC}=1$	Minnesinnehållet på adressen FF läses. Det dataord som läses placeras i PC vid nästa klockpuls. Nästa klockcykel skall vara den första i FETCH-sekvensen.

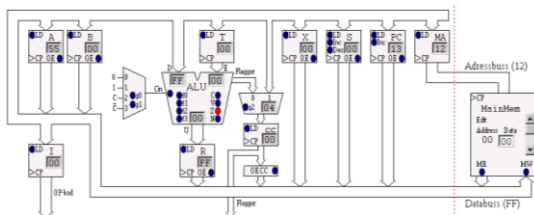


Fokus på...

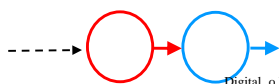
Dagens mål. Du ska kunna:

- ▶ Beskriva Processorns Arbetsätt
 - ▶ Ange styrsignalsekvens för RESET
 - ▶ **Ange styrsignalsekvens för FETCH**
 - ▶ Ange styrsignalsekvenser för olika EXECUTE-faser
- ▶ Konstruera och använda JK- och T-vippa
- ▶ Kunna analysera räknare
- ▶ Ta fram Excitationstabeller

**Läs smart!
Lär dig mer!**



State	RTN	Styrtsignaler	Kommentar
0	$PC \rightarrow MA$ $PC+1 \rightarrow PC$	$OE_{PC}=1$, $LD_{MA}=1$ $incPC=1$	Adressen till instruktionens operationskod kopieras från PC till minnesadressregistret MA. Adressen som finns i PC ökas med ett.
1	$M \rightarrow I$	$MR=1$ $LD_I=1$	Läs operationskoden från minnet. Placera den i instruktionsregistret I. Nästa state skall vara det första i EXECUTE-sekvensen.



(PC) →

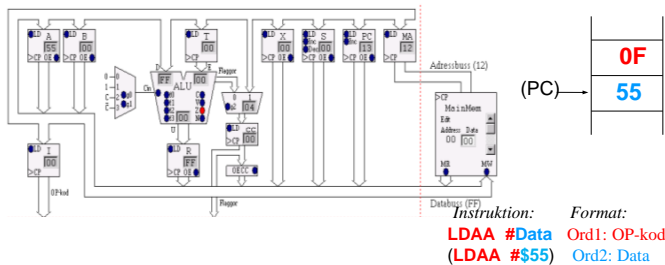


Fokus på...

Dagens mål. Du ska kunna:

- ▶ Beskriva Processorns Arbetsätt
 - ▶ Ange styrsignalsekvens för RESET
 - ▶ Ange styrsignalsekvens för FETCH
 - ▶ **Ange styrsignalsekvenser för olika EXECUTE-faser**
- ▶ Konstruera och använda JK- och T-vippa
- ▶ Kunna analysera räknare
- ▶ Ta fram Excitationstabeller

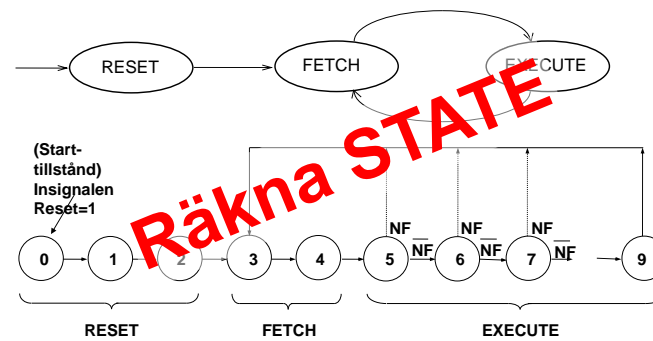
**Läs smart!
Lär dig mer!**



State	RTN	Styrsignaler	Kommentar
0	PC→MA, PC+1→PC	OE _{PC} =1, LD _{MA} =1, IncPC=1	Instruktionens datadel finns i minnesordet efter OP-koden. När EXECUTE-sekvensen inleds pekar PC på instruktionens datadel. PC kopieras därför över till minnesadressregistret MA så att datadelen kan läsas från minnet under nästa klockcykel. Innehållet i PC ökas med ett, så att PC pekar på nästa adress i minnet där OP-koden för nästa instruktion skall finnas.
1	M→A	MR=1, LD _A =1	Läs instruktionens datadel "Data" från minnet och placera den i A-registret. Nästa klockcykel ska vara den första i FETCH-sekvensen.

Arb s

Processorns arbetssätt



Digital o Dator teknik OHLV3

34

FOKUS PÅ...

Fo8

Dagens mål. Du ska kunna:

- ▶ Beskriva Processorns Arbetssätt
 - ▶ Ange styrsignalsekvens för RESET
 - ▶ Ange styrsignalsekvens för FETCH
 - ▶ Ange styrsignalsekvenser för olika EXECUTE-faser
- ▶ **Konstruera och använda JK- och T-vippa**
- ▶ Kunna analysera räknare
- ▶ Ta fram Excitationstabeller

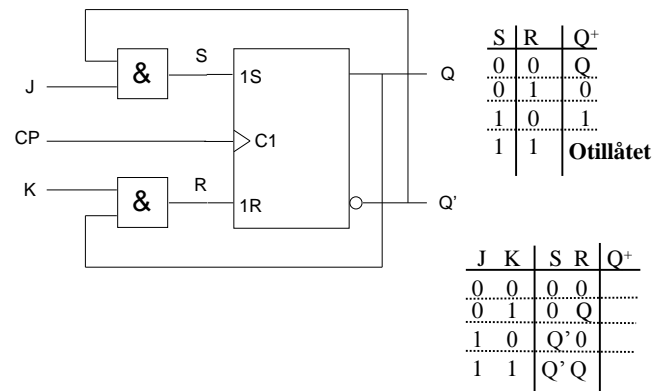
**läs smart!
läs dig mer!**

Digital o Dator teknik OHLV3

35

S5.15

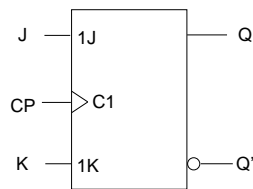
Principen för JK-vippa:



Digital o Dator teknik OHLV3

36

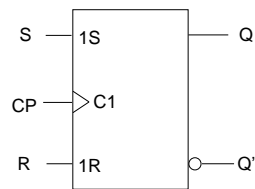
JK - Vippan



J	K	Q ⁺
0	0	Q
0	1	0
1	0	1
1	1	Q'

Digital o Datorteknik OHLV3

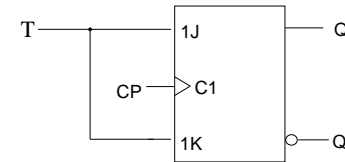
Jfr med SR-vippan



S	R	Q ⁺
0	0	Q
0	1	0
1	0	1
1	1	Otillåtet

37

T - Vippan



Funktionstabell

J	K	Q ⁺
0	0	Q
0	1	0
1	0	1
1	1	Q'

Digital o Datorteknik OHLV3

Funktionstabell

T	Q ⁺
0	Q
1	Q'

38

Asynkrona ingångar för ettställning och nollställning

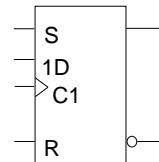
S5.17

Vippor förses ofta med extra ingångar som påverka utsignalen **oberoende** av klockpulsen och övriga insignaler.

Dessa ingångar kallas därför för **asynkrona ingångar**.

För **ett-** (eng **Preset**) och **nollställning** (eng **Clear**) av utsignalen.

Benämns S - R på samma sätt som S- och R-signalerna hos en SR-latch.



D-vippan har asynkrona S- och R-ingångar.

Digital o Datorteknik OHLV3

39

Fokus på...

Fo8

Dagens mål. Du ska kunna:

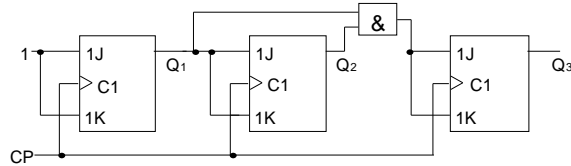
- ▶ Beskriva Processorns Arbets sätt
 - ▶ Ange styrsignalsekvens för RESET
 - ▶ Ange styrsignalsekvens för FETCH
 - ▶ Ange styrsignalsekvenser för olika EXECUTE-faser
- ▶ Konstruera och använda JK- och T-vippa
- ▶ **Kunna analysera räknare**
- ▶ Ta fram Excitationstabeller

**Läs smart!
Lär dig mer!**

Digital o Datorteknik OHLV3

40

Arbetsgång - analys räknare s5.24 ->

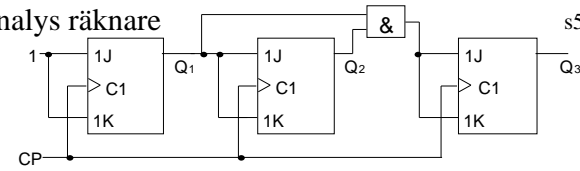


- 1 Studera kopplingen och **bestäm vippornas insignaler** ($T_1=, T_2=, T_3=$)
- 2 Sätt upp en tabell med
 - "Detta tillstånd" (Alla kombinationer av Q_1, Q_2, Q_3)
 - **Insignaler** (T_1, T_2, T_3)
 - "Nästa tillstånd" (Q_1^+, Q_2^+, Q_3^+)
- 3 Ange insignalernas värden i tabellen och----
- 4 ange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

Digital o Datorteknik OHLV3

41

Analys räknare s5.23



1)

$$T_1=1$$

$$T_2=Q_1$$

$$T_3=Q_1 Q_2$$

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	1	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

Digital o Datorteknik OHLV3

42

Analys räknare s5.26

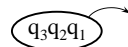
s5.26

Tillståndsdiagram:

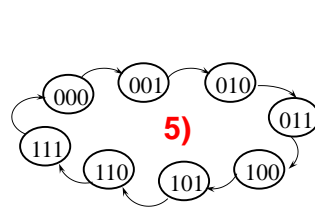
Fördelar:

Vi ser funktionen.

Vi upptäcker om vi har gjort fel!



Q ₃	Q ₂	Q ₁	Q ₃ ⁺	Q ₂ ⁺	Q ₁ ⁺
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0



Digital o Datorteknik OHLV3

43

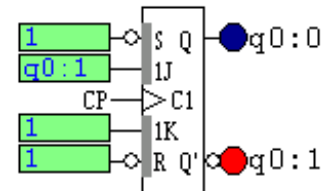
Arb s 106

Uppgift 88

Välj en JK-vippa och anslut Q'-utgången till J-ingången.

Ettställ de övriga ingångarna.

Ge ett antal klockpulser och observera att utgångarna



Gör Uppgift 88

Digital o Datorteknik OHLV3

44

Fokus på...

Fo8

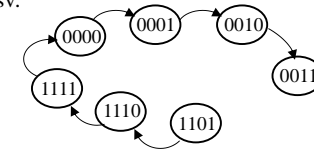
Dagens mål. Du ska kunna:

- ▶ Beskriva Processorns Arbets sätt
 - ▶ Ange styrsignalsekvens för RESET
 - ▶ Ange styrsignalsekvens för FETCH
 - ▶ Ange styrsignalsekvenser för olika EXECUTE-faser
- ▶ Konstruera och använda JK- och T-vippa
- ▶ Kunna analysera räknare
- ▶ **Ta fram Excitationstabeller**

**Läs smart!
Lär dig mer!**

Syntes Räknare

Konstruera en räknare som räknar sekvensen
0-1-2-3-...-14-15-0-1 osv.



Vi vet "Detta tillstånd"

Vi vet "Nästa tillstånd"

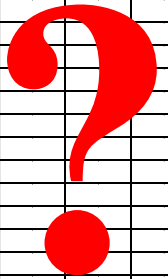
**VAD SKA VIPPORNAS INSIIGNALER VARA
FÖR ATT HAMNA I "NÄSTA TILLSTÄND"????**

⇒ NY TABELL

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	1									
0	1		0								
0	1		1								
1	0	0	0								
1	0	0	1								
1	0	1	0								
1	0	1	1								
1	1	0	0								
1	1	0	1								
1	1	1	0								
1	1	1	1								

**Alla kombinationer
Av "Detta Tillstånd"**

**Fyll i "Nästa
Tillstånd"**



Excitationstabeller - forts

Kmp s 5.18

Vad blir utgången Q⁺
om insignalen är.....

Funktionstabell

J	K	Q ⁺
0	0	Q
0	1	0
1	0	1
1	1	Q'

Vad skall ingången vara
om tillståndsändringen i
Q → Q⁺ är.....

Excitationstabell

Q	Q ⁺	J	K
0	0		
0	1		
1	0		
1	1		

Fo9

Veckans mål:

- Koppla samman register och ALU till en dataväg
- Förstå hur minne är uppbyggd, ansluta detta till datavägen
- Program och hur detta lagras i minne
- Fatta hur datorn startar och arbetar
- Räkare och mera vippor

Dagens mål. Du ska kunna:

- ▶ Konstruera räknare
- ▶ Använda räknaren 74HC163
- ▶ Konstruera en Fast kopplat styrenhet till FLEX
 - ▶ Implementera RESET-fasen i FLEX
 - ▶ Implementera FETCH-fasen i FLEX
 - ▶ Implementera olika EXECUTE-faser i FLEX

**Läs smart!
Lär dig mer!**

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												
0	1	0	1												
0	1	1	0												
0	1	1	1												
1	0	0	0												
1	0	0	1												
1	0	1	0												
1	0	1	1												
1	1	0	0												
1	1	0	1												
1	1	1	0												
1	1	1	1												

Alla kombinationer av "Detta Tillstånd" Vi vet alla "Nästa Tillstånd" Fyll i Vippornas Insignaler

Arbetsgång - syntes räknare

Konstruera en räknare som räknar sekvensen ?????

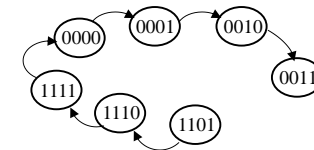
- 1 Rita en **tillståndsgraf**
- 2 Sätt upp en **tabell** med:
 - "Detta tillstånd" (Alla kombinationer av Q₁, Q₂, Q₃)
 - "Nästa tillstånd" (Q₁*, Q₂*, Q₃*)
 - Vippornas **Insignaler**
- 3 Ange "**Nästa tillstånd**" i tabellen
- 4 Använd vippornas excitationstabell och ange **vippornas insignaler**
-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

Arb s 110

Uppgift 93

Konstruera en räknare som räknar sekvensen 0-1-2-3-...-14-15-0-1 osv.

Räknaren skall realiseras med JK-vippor grindar tillgängliga i Kopplingsboxen



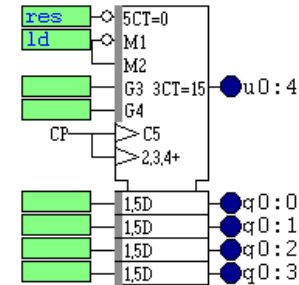
Fo9

Dagens mål. Du ska kunna:

- ▶ Konstruera räknare
- ▶ **Använda räknaren 74HC163**
- ▶ Konstruera en Fast kopplat styrenhet till FLEX
 - ▶ Implementera RESET-fasen i FLEX
 - ▶ Implementera FETCH-fasen i FLEX
 - ▶ Implementera olika EXECUTE-faser i FLEX

**Läs smart!
Lär dig mer!**

Uppgift 96 -Räknaren 74HC163



Grönt = etta

Gör uppgift 96

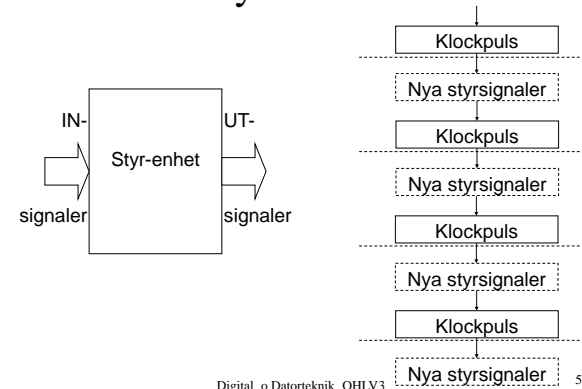
Fo9

Dagens mål. Du ska kunna:

- ▶ Konstruera räknare
- ▶ Använda räknaren 74HC163
- ▶ **Konstruera en Fast kopplat styrenhet till FLEX**
 - ▶ Implementera RESET-fasen i FLEX
 - ▶ Implementera FETCH-fasen i FLEX
 - ▶ Implementera olika EXECUTE-faser i FLEX

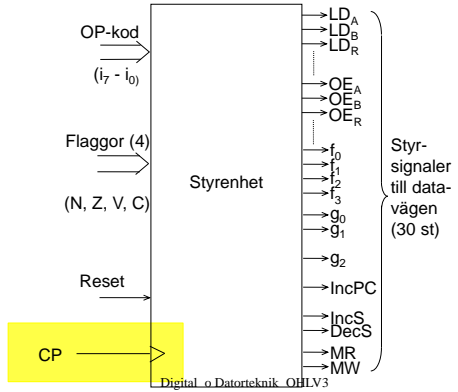
**Läs smart!
Lär dig mer!**

Styrenheten



Styrenheten - forts

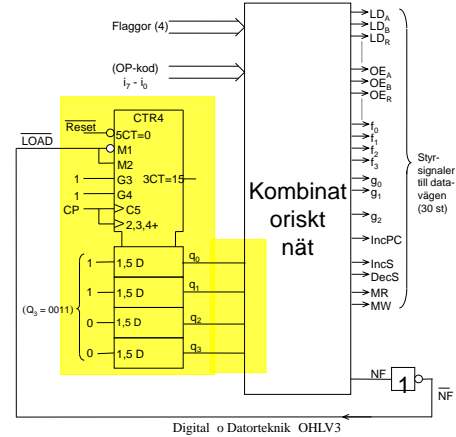
Arb s 120



57

Styrenheten - forts

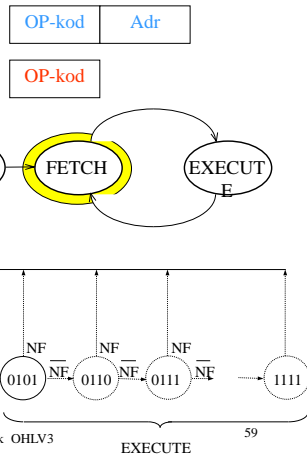
Arb s 123



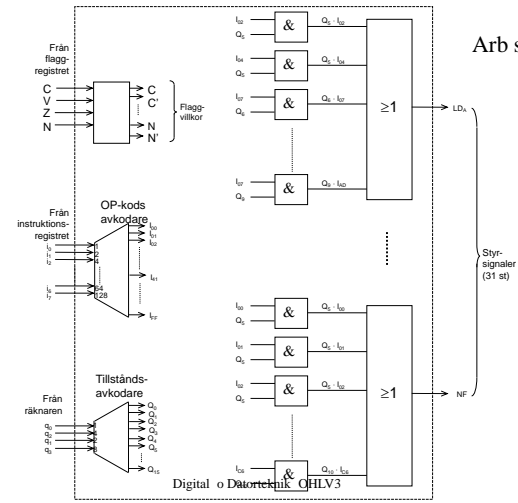
58

Mask. prog i minnet	Tillhörande assemblerprog
0C 10	LDAB #S23
0D 23	
0E 29	ADDB \$F3
0F F3	
10 02	TFR B.A
11 4F	CMPB #S03
12 03	
13 61	BLO \$29
14 13	

Aktivera



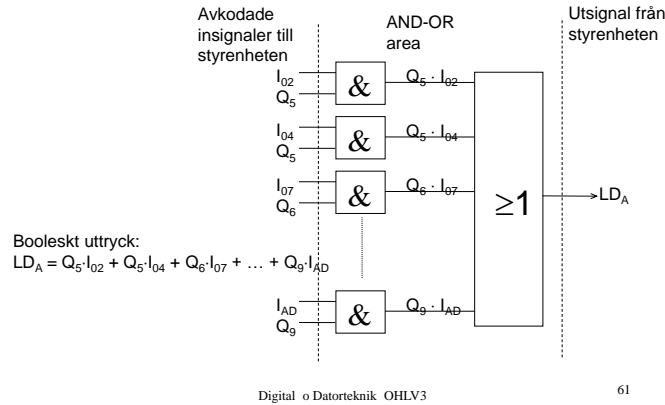
Arb s 124



Styrenheten - forts

Arb s 125

Fo9



Dagens mål. Du ska kunna:

- ▶ Konstruera räknare
- ▶ Använda räknaren 74HC163
- ▶ Konstruera en Fast kopplat styrenhet till FLEX
 - ▶ Implementera RESET-fasen i FLEX
 - ▶ Implementera FETCH-fasen i FLEX
 - ▶ Implementera olika EXECUTE-faser i FLEX

Läs smart!
Lär dig mer!

Digital o Datorteknik OHLV3

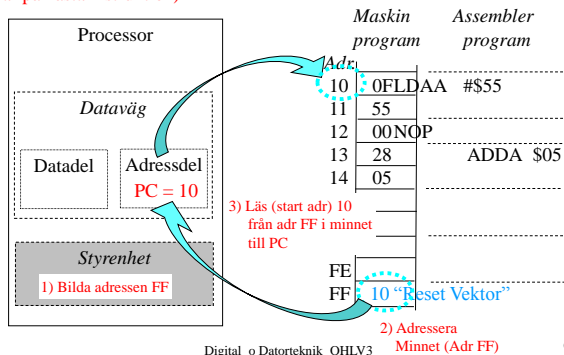
62

Processorns arbetsätt

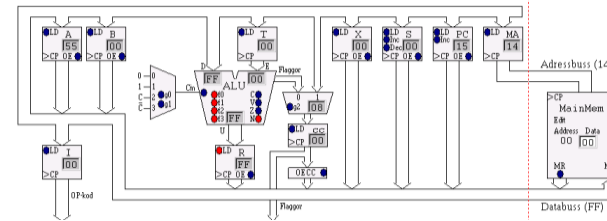
RESET

Arb s 95

PC: Programräknare
(Pekar på nästa instruktion)



Processorns arbetsätt - RESET

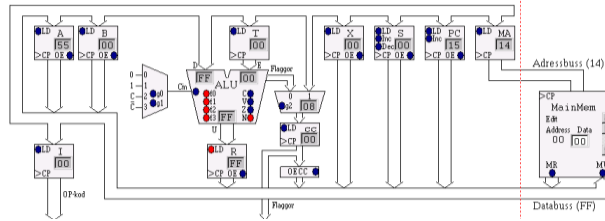


Tillstånd	Summatern	RTN-beskrivning	Styrsignaler (=1)
Q ₀	Q ₀	FF ₁₆ → R	f ₃ , f ₂ , f ₁ , f ₀ , LD _R
Q ₁	Q ₁	R → MA	OE _R , LD _{MA}
Q ₂	Q ₂	M → PC	MR, LD _{PC}

Digital o Datorteknik OHLV3

64

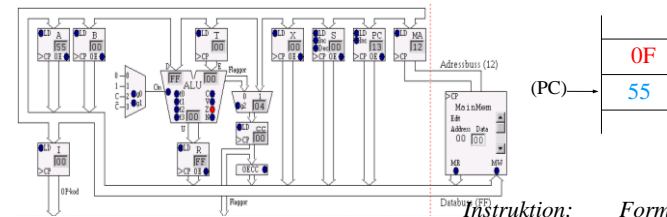
Processorns arbetssätt - FETCH



Tillstånd	Summatern	RTN-beskrivning	Styrsignaler (=1)
Q ₃	Q ₃	PC → MA PC+1 → PC	OE _{PC} , LD _{MA} IncPC
Q ₄	Q ₄	M → I	MR, LD _I

Digital o Datorteknik OHLV3

65

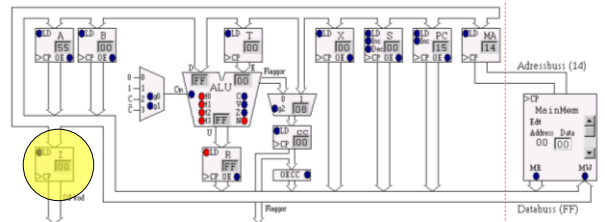


Instruktion: **Format.**
LDAA #Data Ord1: OF
(LDAA #\$55) Ord2: D

Tillstånd	Summatern	RTN-beskrivning	Styrsignaler (=1)
Q ₅	Q ₅ *I _{0F}	PC → MA PC+1 → PC	OE _{PC} , LD _{MA} IncPC
Q ₆	Q ₆ *I _{0F}	M → A	MR, LD _A , NF

Digital o Datorteknik OHLV3

66



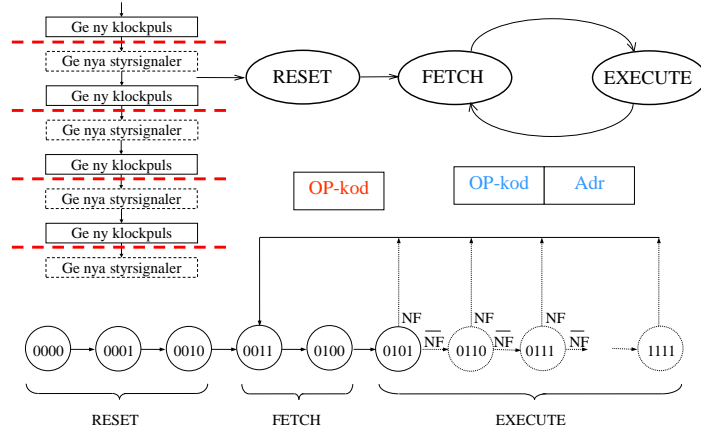
Tillstå	Summter	RTN-beskrivning	Styrsignaler (=1)
struktionsnr (OP-kod) AND State xx			
	I91 * Q5	Y → Z	
	I91 * Q6	Q → P	
	I91 * Q7	W → U	NF

LDAB #\$23
ADDB \$F3

Digital o Datorteknik OHLV3

67

Aktivera "Processorns arbetssätt"



Digital o Datorteknik OHLV3

68