

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:

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

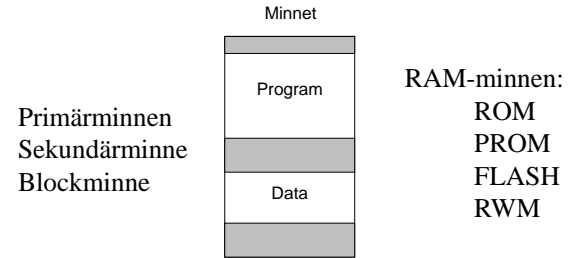
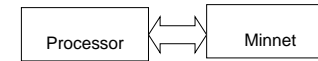
Dagens mål, Du ska kunna:

- ▶ Förstå uppbyggnaden av ett minne
- ▶ Ansluta CC-register till datavägen
- ▶ Konstruera en enkel dataväg
- ▶ Kunna programmera en enkel dataväg (RTN)
- ▶ Förstå uppbyggnaden av T och JK-vipporna

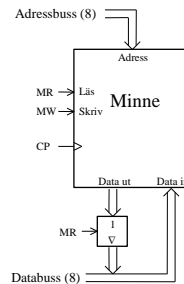
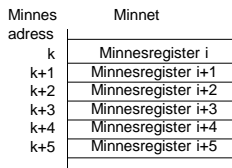
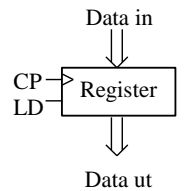
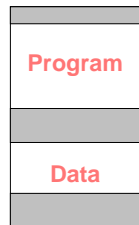
*Läs smart!
Lär dig mer!*

Ext 15

Minnet

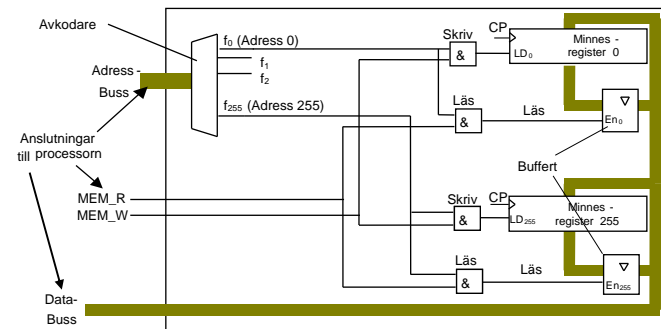


Minne



Minns Man Minnet?

Ext 15



FOKUS PÅ...

Fo7

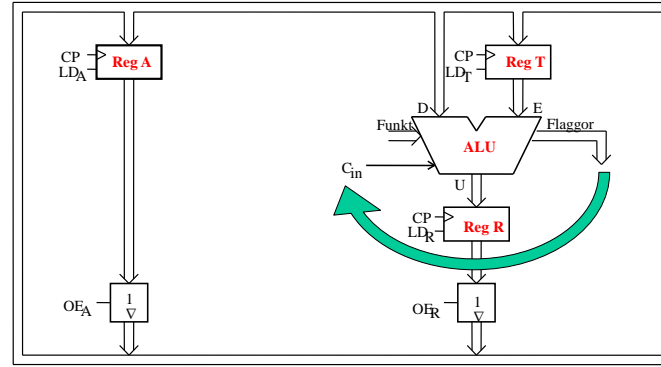
Dagens mål, Du ska kunna:

- ▶ Förstå uppbyggnaden av ett minne
- ▶ **Ansluta CC-register till datavägen**
- ▶ Konstruera en enkel dataväg
- ▶ Kunna programmera en enkel dataväg (RTN)
- ▶ Förstå uppbyggnaden av T och JK-vipporna

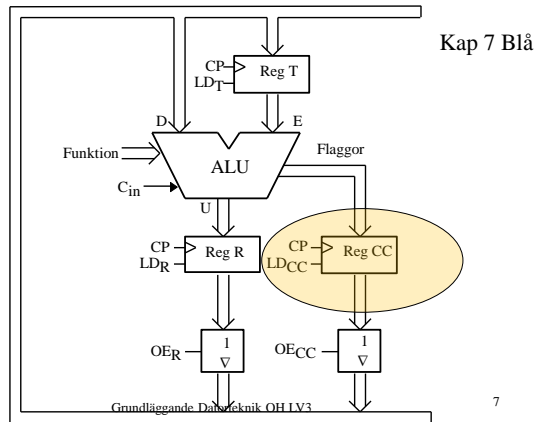
**Läs smart!
Lär dig mer!**

T, R och ALU är på 8 bitar

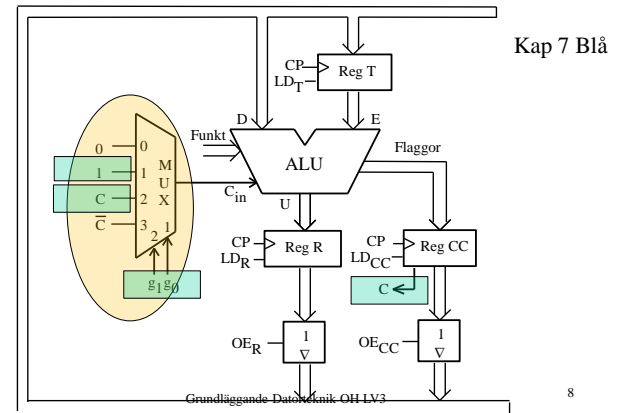
Hur utföra 16-bitars addition??? $X + 1234_{16} = ?$



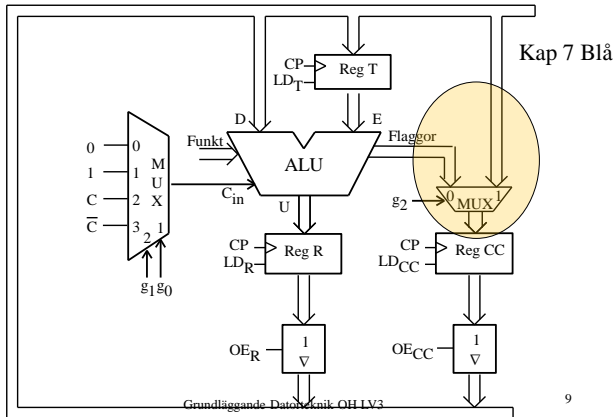
Inkoppling av flaggregister mellan ALU och buss.



Inkoppling av väljare (multiplexer) för val av C_{in}.



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



FOKUS PÅ...

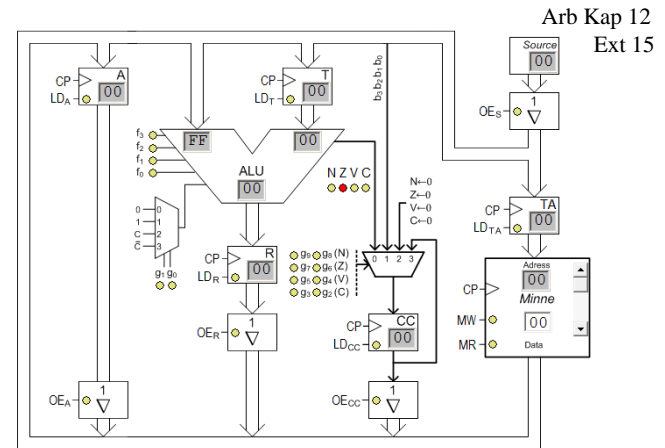
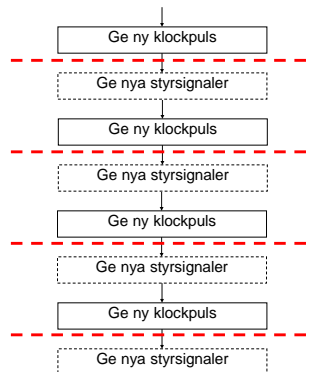
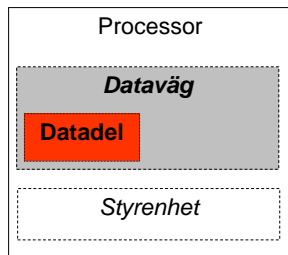
Fo7

Dagens mål, Du ska kunna:

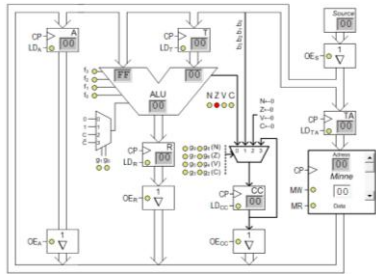
- ▶ Förstå uppbyggnaden av ett minne
- ▶ Ansluta CC-register till datavägen
- ▶ **Konstruera en enkel dataväg**
- ▶ **Kunna programmera en enkel dataväg (RTN)**
- ▶ Förstå uppbyggnaden av T och JK-vipporna

**Läs smart!
Lär dig mer!**

Enkel dataväg



Arb Kap 12
Ext 15



Ge en sekvens av styrsignaler som utför:

RTN	CP	Source	OE _S	OE _A	OE _D	OE _{CC}	LD _A	LD _D	LD _{LA}	LD _{LC}	f ₁	f ₂	f ₃	f ₄	BrZ	B	MR	MW

Ge en sekvens av styrsignaler som utför:



**Gör enkla uppgifter...
testa...
Hitta känslan...**

övning:

- CP1: _____
- CP2: _____
- CP3: _____
- CP4: _____
- CP5: _____

FOKUS PÅ...

F07

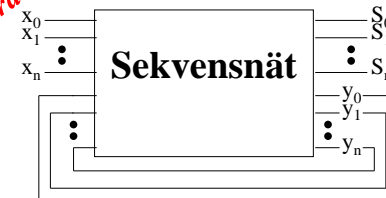
Dagens mål, Du ska kunna:

- ▶ Förstå uppbyggnaden av ett minne
- ▶ Ansluta CC-register till datavägen
- ▶ Konstruera en enkel dataväg
- ▶ Kunna programmera en enkel dataväg (RTN)
- ▶ Förstå uppbyggnaden av T och JK-vipporna

**Läs smart!
Lär dig mer!**

S5.1

Aktivera

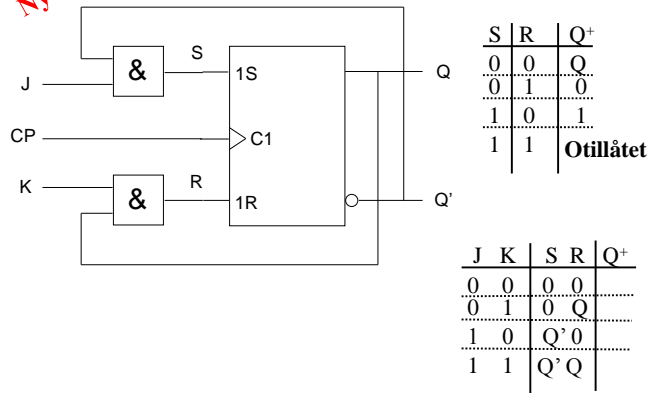


- Återkoppling
- Minnesfunktion
- **RÄKNARE**

Ny vippan

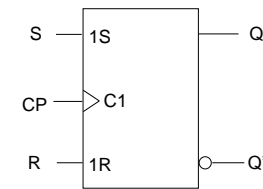
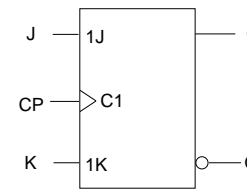
Principen för JK-vippan:

S5.15



JK - Vippan

Jfr med SR-vippan



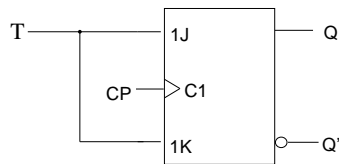
J	K	Q ⁺
0	0	Q
0	1	0
1	0	1
1	1	Q'

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

Ny vippan

T - Vippan

S5.16



Funktionstabell

J	K	Q ⁺
0	0	Q
0	1	0
1	0	1
1	1	Q'

Funktionstabell

T	Q ⁺
0	Q
1	Q'

Asynkrona ingångar för ettställning och nollställning

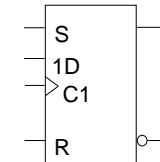
S5.17

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.

Veckans mål:

- ▶ Koppla samman register och ALU till en dataväg
- ▶ Förstå hur ett minne är uppbyggt, ansluta detta till datavägen
- ▶ Program och hur detta lagras i minne
- ▶ Fatta hur datorn startar och arbetar
- ▶ Kunna analysera och konstruera räknare

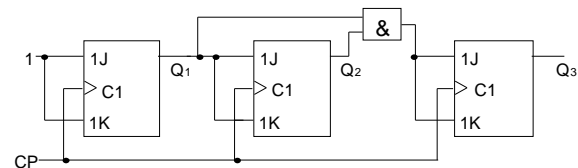
Dagens mål. Du ska kunna:

- ▶ Analysera räknare
- ▶ Förstå von Neuman's princip och hur program är uppbyggt
- ▶ Beskriva en dataväg med adressdel
- ▶ Beskriva Processorns Arbetsätt
 - ▶ Ange styrsignalsekvens för RESET
 - ▶ Ange styrsignalsekvens för FETCH
 - ▶ Ange styrsignalsekvenser för olika EXECUTE-faser
- ▶ Ta fram Excitationstabeller för vippor

**Läs smart!
Lär dig mer!**

LV3 Fo8

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**

Fokus på...

Fo8

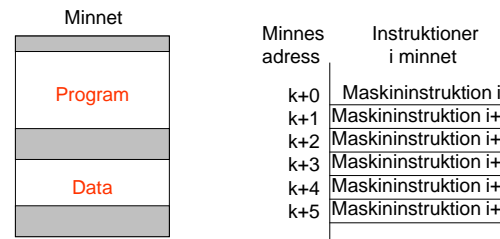
Dagens mål. Du ska kunna:

- ▶ Analysera räknare
- ▶ **Förstå von Neuman's princip och hur program är uppbyggt**
- ▶ Beskriva en dataväg med adressdel
- ▶ Beskriva Processorns Arbetsätt
 - ▶ Ange styrsignalsekvens för RESET
 - ▶ Ange styrsignalsekvens för FETCH
 - ▶ Ange styrsignalsekvenser för olika EXECUTE-faser
- ▶ Ta fram Excitationstabeller för vippor

**Läs smart!
Lär dig mer!**

Arb s 92

Program o Minne



Program o Minne - forts

Instruktionsformat



Maskinprogram

Maskinprogram	Maskinprogram
00001111 ₂	0F ₁₆
00001011 ₂	0E ₁₆
00111111 ₂	3F ₁₆
11111110 ₂	FE ₁₆
00011001 ₂	19 ₁₆
01000001 ₂	41 ₁₆
01001010 ₂	4A ₁₆

Fokus på...

Fo8

Dagens mål. Du ska kunna:

- ▶ Kunna analysera räknare
- ▶ Förstå von Neuman's princip och hur program är uppbyggt
- ▶ **Dataväg med adressdel**
- ▶ Beskriva Processorns Arbetssätt
 - ▶ Ange styrsignalsekvens för RESET
 - ▶ Ange styrsignalsekvens för FETCH
 - ▶ Ange styrsignalsekvenser för olika EXECUTION-faser
- ▶ Ta fram Excitationstabeller för vipp

**Läs smart!
Lär dig mer!**

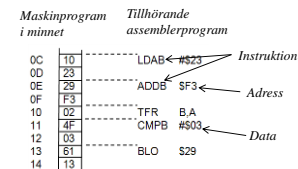
Program och minne

- **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)
09	91	
0A	28	- add to the A-register the contents of this address (02H)
0B	92	
0C	61	- branch, if carry=1, to the address = this number (07H) + address of next OPCODE (0EH) (= 15H)
0D	07	
0E	13	- store the contents of the A-register on this address (04H)
0F	94	
10	47	- clear the A-register
11	13	- store the contents of acc A (00H) on this address (03H)
12	03	- jump to
13	59	
14	1B	- store this address (1BH)
15	13	- store the contents of the A-register on this address (04H)
16	94	
17	0F	- load the A-register with
18	01	- this number (01H)
19	13	- store contents of the A-register (01H)
1A	93	- on this address (03H)
1R	0R	



Fokus på...

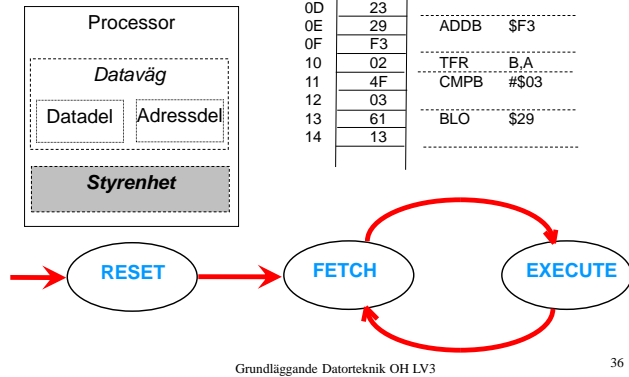
Fo8

Dagens mål. Du ska kunna:

- ▶ Kunna analysera räknare
- ▶ Förstå von Neuman's princip och hur program är uppbyggt
- ▶ Dataväg med adressdel
- ▶ **Beskriva Processorns Arbetssätt**
 - ▶ Ange styrsignalsekvens för RESET
 - ▶ Ange styrsignalsekvens för FETCH
 - ▶ Ange styrsignalsekvenser för olika EXECUTION-faser
- ▶ Ta fram Excitationstabeller för vipp

**Läs smart!
Lär dig mer!**

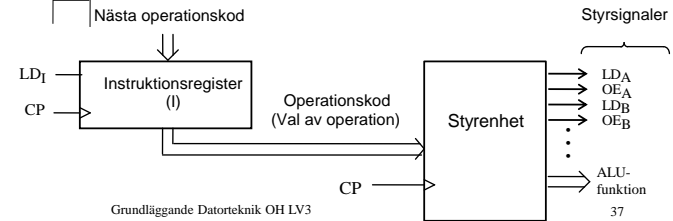
Processorns arbetsätt



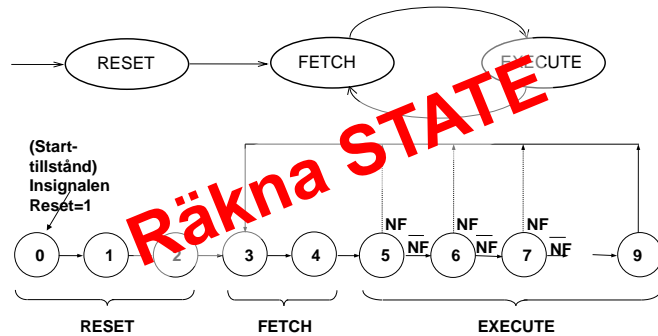
Maskinprogram i minnet		Tillhörande assemblerprogram	
Adr.			
0C	10	LDB	#\$23
0D	23		
0E	29	ADDB	\$F3
0F	F3		
10	02	TFR	B,A
11	4F	CMPB	#\$03
12	03		
13	61	BLO	\$29
14	13		

Maskinprogram i minnet		
Adr.		
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	

Styrsignalgenerering för den databehandlande enheten.



Processorns arbetsätt



Räkna STATE

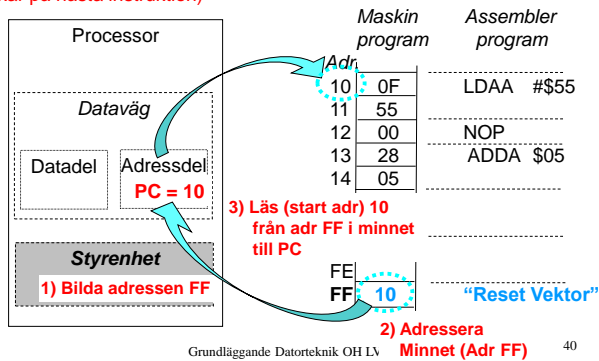
Dagens mål. Du ska kunna:

- ▶ Kunna analysera räknare
- ▶ Program och minne, von Neuman
- ▶ Dataväg med adressdel
- ▶ Beskriva Processorns Arbetsätt
 - ▶ Ange styrsignalsekvens för RESET
 - ▶ Ange styrsignalsekvens för FETCH
 - ▶ Ange styrsignalsekvenser för olika EXECUTE-faser
- ▶ Ta fram Excitationstabeller för vippor

**Läs smart!
Lär dig mer!**

Processorns arbets sätt - RESET Arb s 95

PC: Programräknare
(Pekar på nästa instruktion)



Fokus på...

Fo8

Dagens mål. Du ska kunna:

- ▶ Kunna analysera räknare
- ▶ Förstå von Neuman's princip och hur program är uppbyggt
- ▶ Dataväg med adressdel
- ▶ Beskriva Processorns Arbets sätt
 - ▶ Ange styrsignalsekvens för RESET
 - ▶ Ange styrsignalsekvens för FETCH
 - ▶ Ange styrsignalsekvenser för olika EXECUTIVE
- ▶ **Ta fram Excitationstabeller för vinn smart!**

**Läs smart!
Lär dig mer!**

Excitationstabeller Kmp s 5.18

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

Funktionstabell

D	Q ⁺
0	0
1	1

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

Excitationstabell

Q	Q ⁺	D
0	0	0
0	1	1
1	0	0
1	1	1

Excitationstabeller - forts Kmp s 5.18

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

Funktionstabell

T	Q ⁺
0	Q
1	Q'

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

Excitationstabell

Q	Q ⁺	T
0	0	0
0	1	1
1	0	1
1	1	0

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

Funktionstabell

J	K	Q^+
0	0	0
0	1	0
1	0	1
1	1	Q'

Vad skall ingången vara om tillståndsändringen i $Q \rightarrow Q^+$ är.....

Excitationstabell

Q	Q^+	J	K
0	0	0	-
0	1	1	-
1	0	-	1
1	1	-	0

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

Funktionstabell

S	R	Q^+
0	0	Q
0	1	0
1	0	1
1	1	*

Vad skall ingången vara om tillståndsändringen i $Q \rightarrow Q^+$ är.....

Excitationstabell

Q	Q^+	S	R
0	0	0	-
0	1	1	0
1	0	0	1
1	1	-	0

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
- ▶ Konstruera en Fast kopplat styrenhet till Flisp
 - ▶ Implementera RESET-fasen i Flisp
 - ▶ Implementera FETCH-fasen i Flisp
 - ▶ Implementera olika EXECUTE-faser i Flisp

**Läs smart!
Lär dig mer!**

	UTSIGNALER								INSIGNALER				
	Detta tillstånd				Nästa tillstånd				T_3	T_2	T_1	T_0	
Q	q_3	q_2	q_1	q_0	Q^+	q_3^+	q_2^+	q_1^+	q_0^+				
0	0	0	0	0									
1	0	0	0	1									
2	0	0	1	0									
3	0	0	1	1									
4	0	1											
5	0												
6				0									
-			1	1									
8		0	0	0									
9	1	0	0	1									
10	1	0	1	0									
11	1	0	1	1									
12	1	1	0	0									
13	1	1	0	1									
14	1	1	1	0									
15	1	1	1	1									

Alla kombinationer Av "Detta Tillstånd"

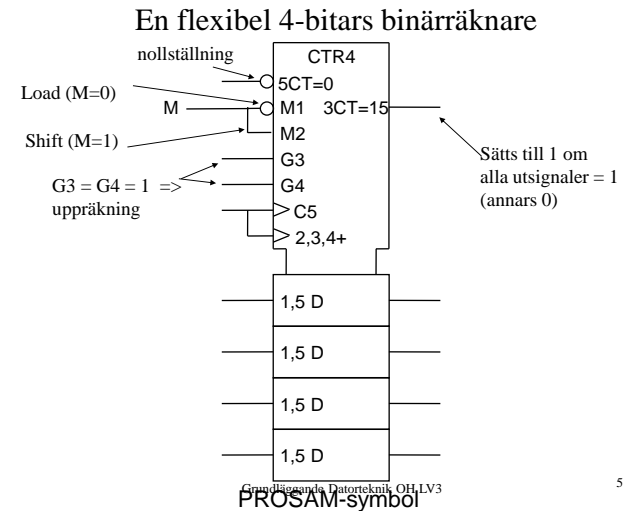
Fyll i "Nästa Tillstånd"



Arbetsgång - syntes räknare

Konstruera en räknare som räknar sekvensen ????

- 1 Rita en **tillståndsgraf**
- 2 Sätt upp en **tabell** med:
 - "Detta tillstånd" (Alla kombinationer av Q_1, Q_2, Q_3)
 - "Nästa tillstånd" (Q_1^+, Q_2^+, Q_3^+)
 - Vippornas **Insignaler**
- 3 Ange "**Nästa tillstånd**" i tabellen
- 4 Använd vippornas excitationstabell och ange **vippornas insignaler**
- 5 **Minimera** uttrycken för insignalerna
- 6 **Realisera** räknaren



F09

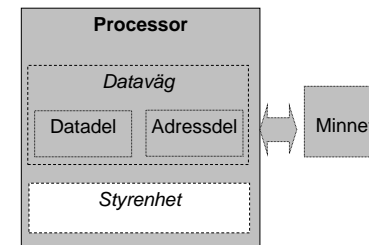
Arb s 99

Dataväg med fast styrenhet

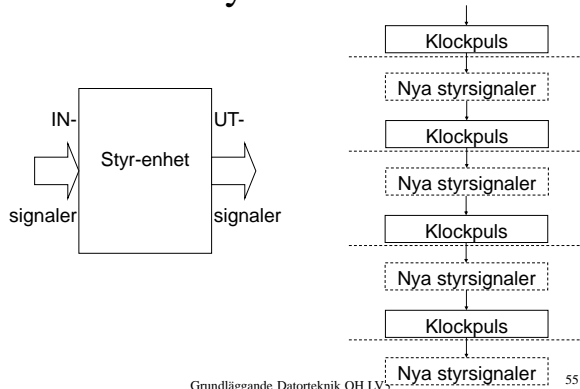
Dagens mål. Du ska kunna:

- ▶ Konstruera räknare
- ▶ **Konstruera en Fast kopplat styrenhet till Flisp**
 - ▶ Implementera RESET-fasen i Flisp
 - ▶ Implementera FETCH-fasen i Flisp
 - ▶ Implementera olika EXECUTE-faser i Flisp

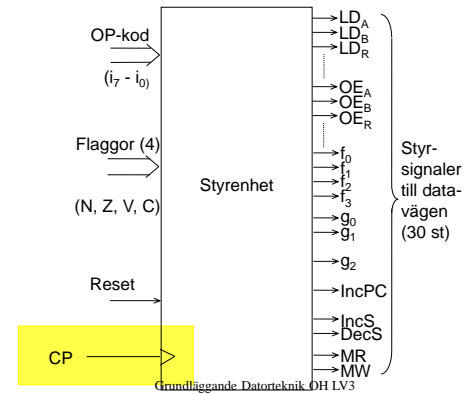
**läs smart!
läs dig mer!**



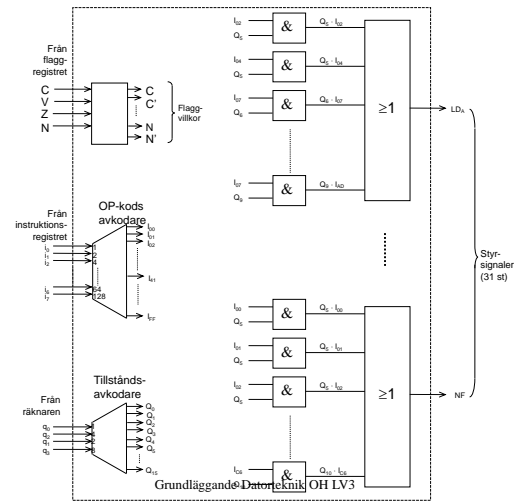
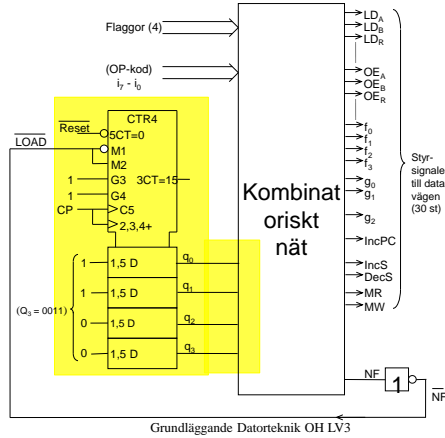
Styrenheten



Styrenheten - forts

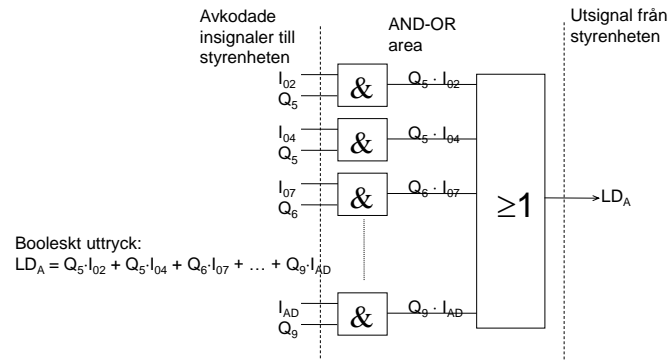


Styrenheten - forts



Styrenheten - forts

Fo9



Dagens mål. Du ska kunna:

- ▶ Konstruera räknare
- ▶ Konstruera en Fast kopplat styrenhet till Flisp
 - ▶ Implementera RESET-fasen i Flisp
 - ▶ Implementera FETCH-fasen i Flisp
 - ▶ Implementera olika EXECUTE-faser i Flisp

Läs smart!
Lär dig mer!