

# Sekvensnät – vippor, register och bussar

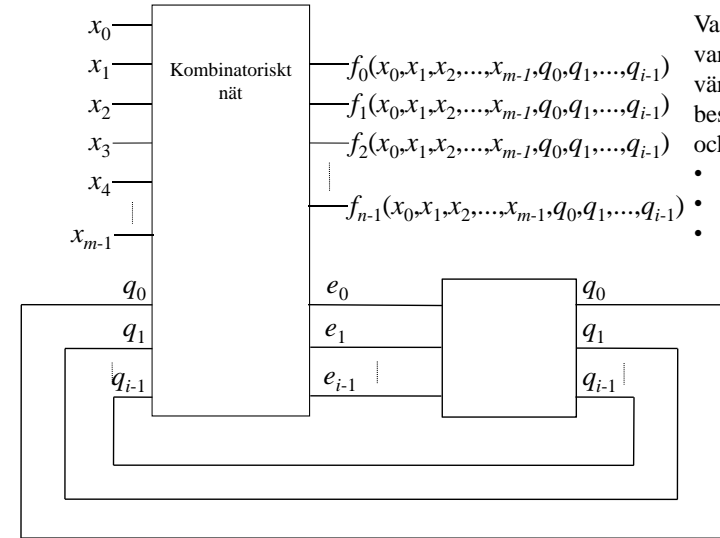
Dagens föreläsning:

Lärobok kap.5  
Arbetsbok kap 8,9,10

Ur innehållet:

- Hur fungerar en SR-latch?
- Hur konstrueras JK-, D- och T-vippor?
- Funktionstabeller och excitationstabeller
- Register som minneselement
- Register och bussar

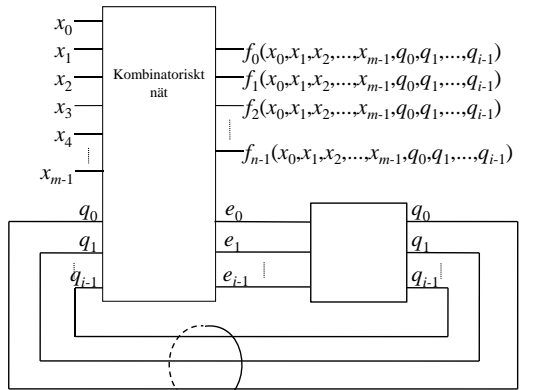
# Sekvensnät



Varje utgång har i varje ögonblick det värde som entydigt bestäms av insignaler och tillstånd.

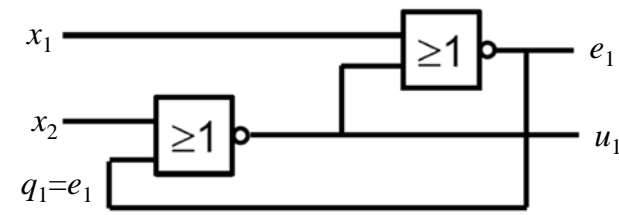
- $m$  insignaler
- $n$  utsignaler
- $2^i$  tillstånd

# Asynkront eller synkront sekvensnät

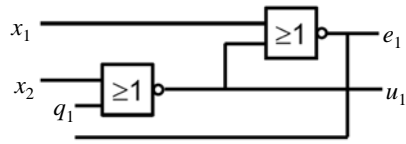


Återkoppling ("feedback")  
Omedelbar (asynkron)  
Tidsbestämd (synkron)

# Exempel: Omedelbar återkoppling



### Analys...

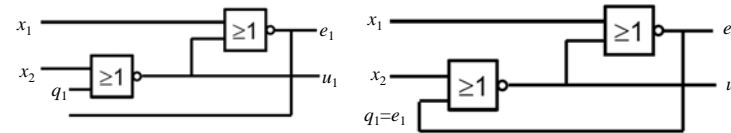


$$e_1 = [x_1 + (x_2 + q_1)]' = x_1' (x_2 + q_1)$$

$$u_1 = (x_2 + q_1)' = x_2' q_1'$$

$x_1$	$x_2$	$q_1$	$e_1$	$u_1$
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0

### ..och när vi kopplar ihop $e_1$ och $q_1$ ...



$x_1$	$x_2$	$q_1$	$e_1$	$u_1$
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0

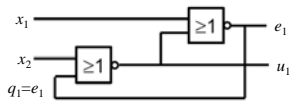
$x_1$	$x_2$	$q_1$	$e_1$	$u_1$
0	0	0	$q_1$	$q_1'$
0	0	1	$q_1$	$q_1'$
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0

$x_1$	$x_2$	$e_1$	$u_1$
0	0	$q_1$	$q_1'$
0	1	1	0
1	0	0	1
1	1	0	0

Vid återgång till 00 uppstår "kapplöpning" förbjudet tillstånd

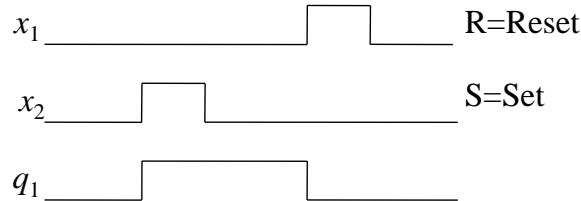
(kan inte finnas...)

### Latch (låskrets)



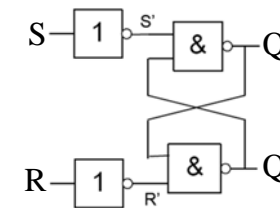
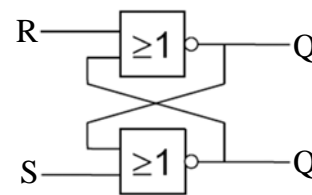
$q_1$  kan ettställas eller nollställas för att därefter behålla värdet. Kopplingen är ett "minneselement" och kallas SR-latch.

$x_1$	$x_2$	$e_1$	$u_1$
0	0	$q_1$	$q_1'$
0	1	1	0
1	0	0	1
1	1	-	-



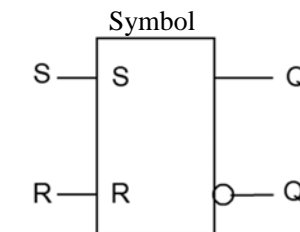
### SR-latch

Alternativa realiseringar



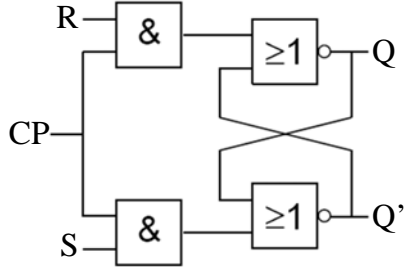
Funktionstabell

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



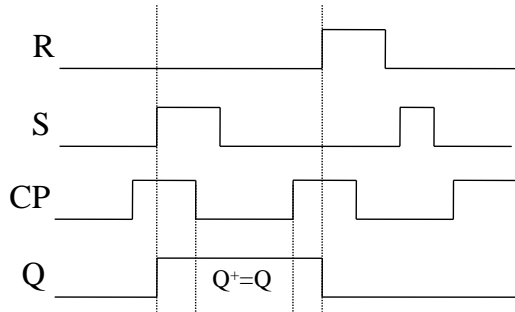
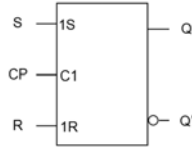
Asynkront minneselement

# Grindad SR-latch

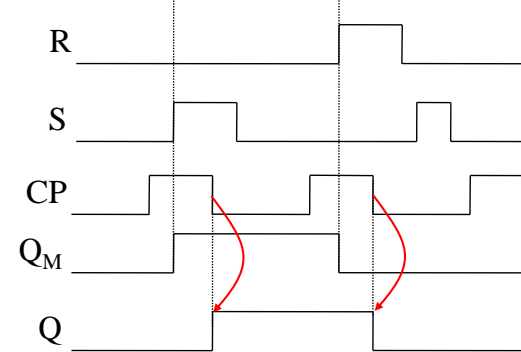
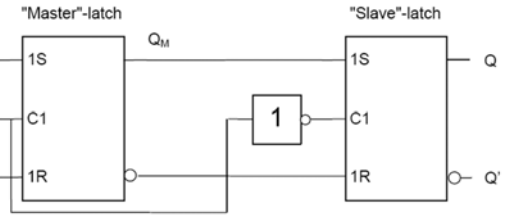
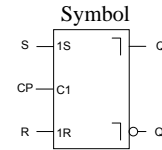


S	R	Q'
0	0	Q
0	1	0
1	0	1
1	1	*

Symbol



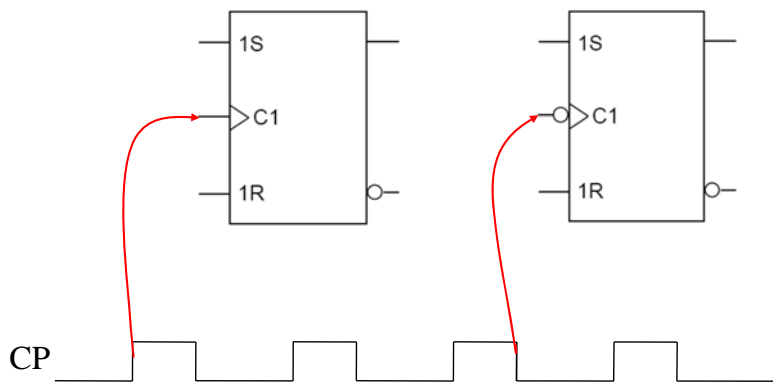
# Master-slave SR-latch



Ändringen i Q kommer alltid vid en negativ flank hos CP.

Signalen fördröjs (halv klockcykel)

# Flanktriggade SR-vippor



Funktions-  
tabell

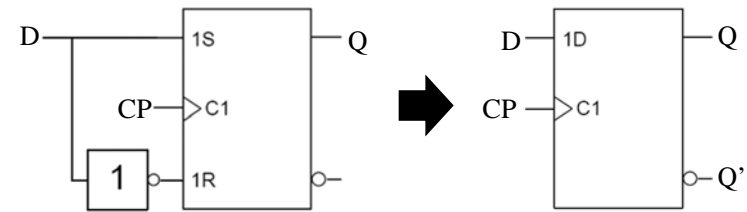
S	R	Q <sup>+</sup>
0	0	Q
0	1	0
1	0	1
1	1	*

Exitations-  
tabell

Q	Q <sup>+</sup>	S	R
0	0	-	-
0	1	1	0
1	0	0	1
1	1	-	0

*Synkront minneselement*

# Flanktriggad D-vippa

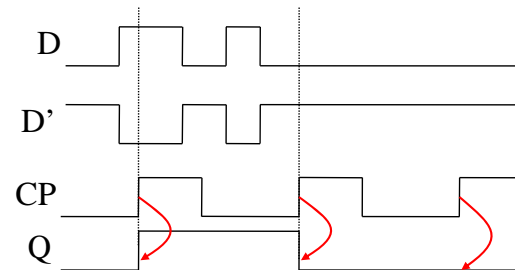


Funktions-  
tabell

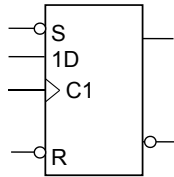
D	Q <sup>+</sup>
0	0
1	1

Exitations-  
tabell

Q	Q <sup>+</sup>	D
0	0	0
0	1	1
1	0	0
1	1	1



## D-vippa med asynkron Set/Reset

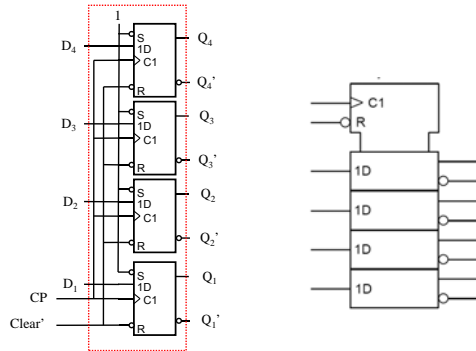


Då SR=11 ( $Q^+ = Q$ ) fungerar vippan som en ordinär D-vippa  
 Om SR=01, ettställs Q asynkront (väntar EJ på flank...)  
 Om SR=10, nollställs Q asynkront

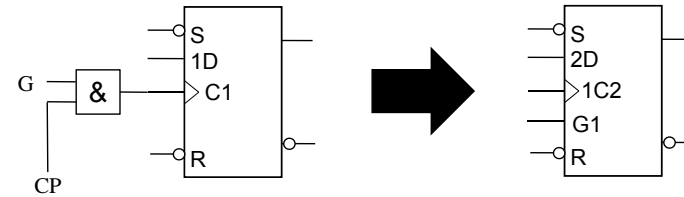
Vi säger att S och R ingångarna är "aktivt låga"

Exempel på användning:

4-bitars REGISTER med RESET-funktion

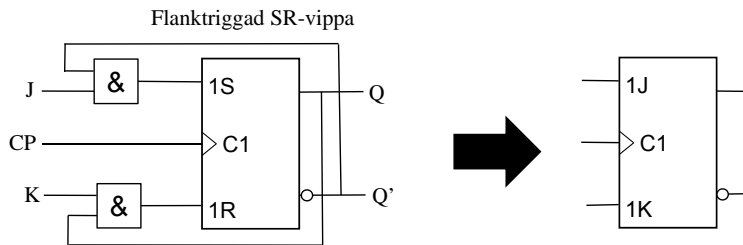


## D-vippa med "Load Enable"



Styrsignalen G ("Gate") kan användas för att "strypa" klockpuls.  
 Då  $G=0$  behåller vippan sitt värde oavsett vad som finns på D-ingången

## Flanktriggad JK-vippa



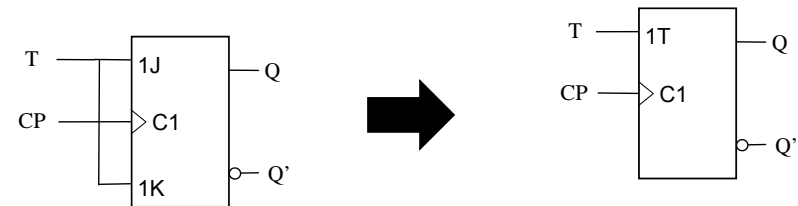
Funktions-

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

Exitations-

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

## Flanktriggad T-vippa ("toggle")



Funktions-

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

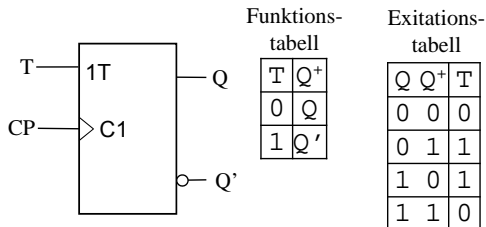
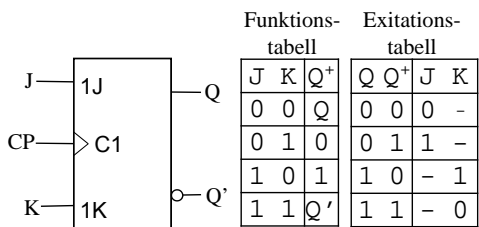
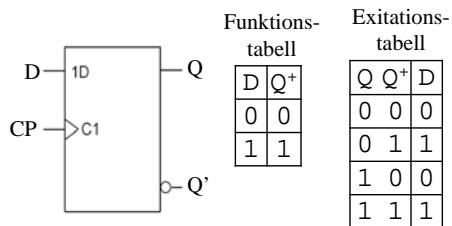
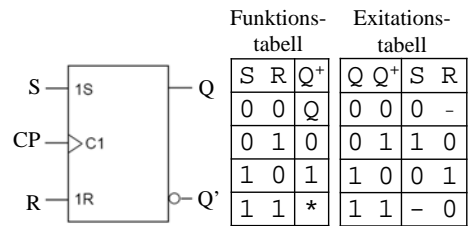
Funktions-

T	$Q^+$
0	Q
1	$Q'$

Exitations-

Q	$Q^+$	T
0	0	0
0	1	1
1	0	1
1	1	0

## Sammanfattning av vippor



## Användning av vippor

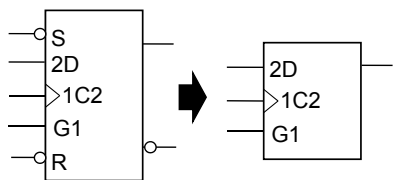
Vi har klarat av:

- Hur fungerar en SR-latch?
- Hur konstrueras JK-, D- och T-vippor?
- Funktionstabeller och excitationstabeller

..och ska nu titta på viktiga tillämpningar

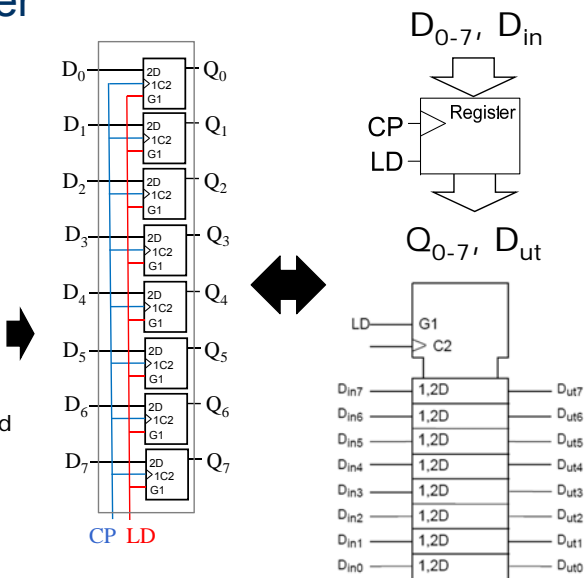
- Register som minneselement
- Bussar

## 8-bitars minnesregister

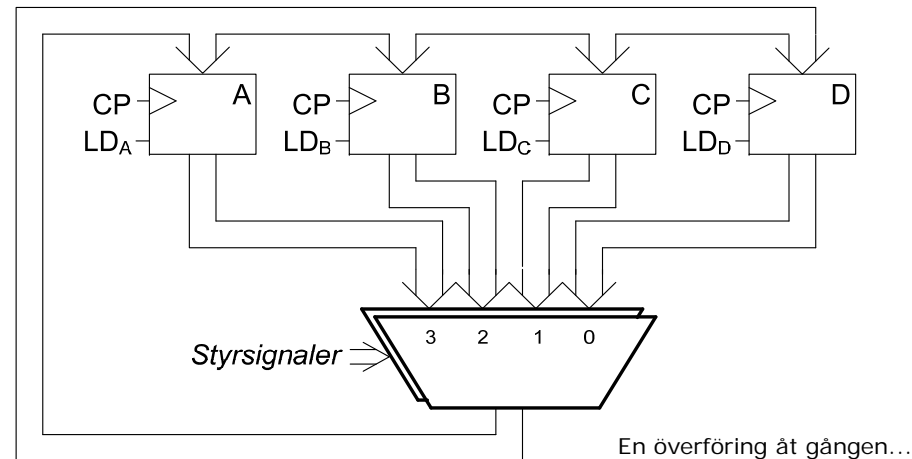


Utgå från känd D-vippa, ta bort asynkrona ingångar och Q'...

Packa ihop 8 likadana med gemensam CP resp G...

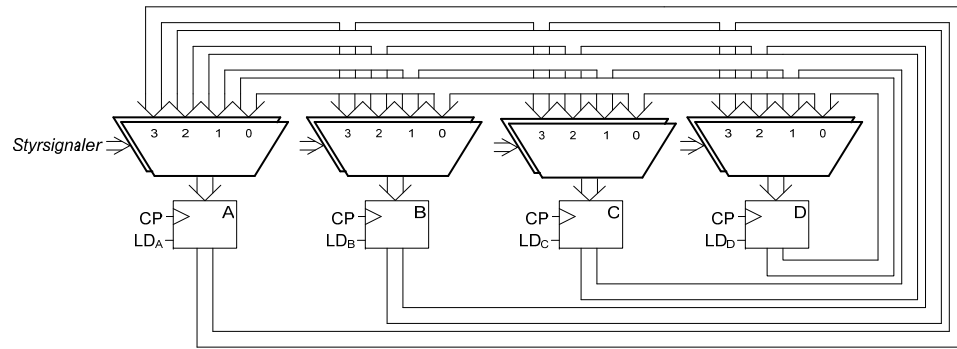


## Dataöverföring, register och bussar



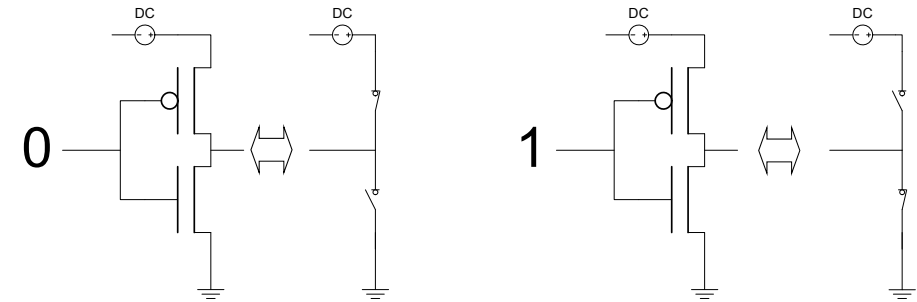
En överföring åt gången...

# Snabbare, dyrare...

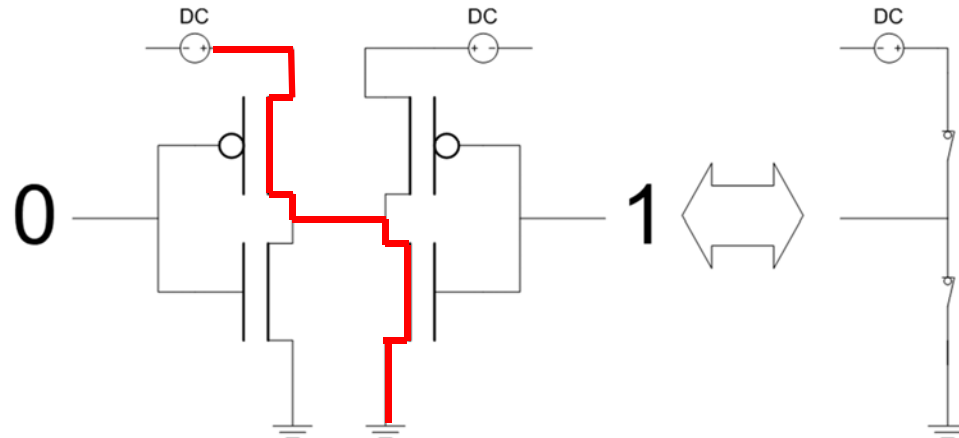


Samtidiga överföringar till samtliga register...

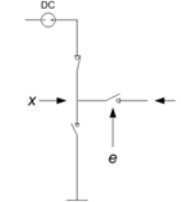
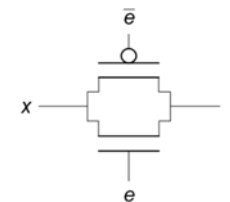
# Kan utgångarna kopplas samman?



# Kortslutning...



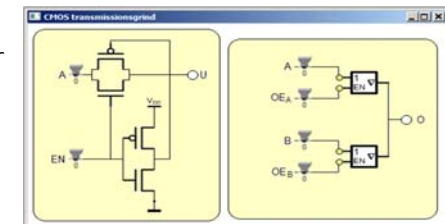
# Transmissionsgrind ("three-state")



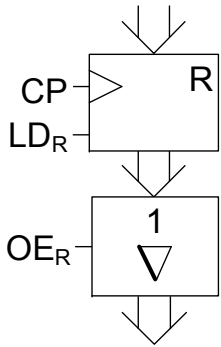
x	e	f
0	0	Z
1	0	Z
0	1	0
1	1	1

"högimpedanstillstånd"-  
varken 0 eller 1 kopplas  
till utgången

"three-state"- (0,1,Z) buffer



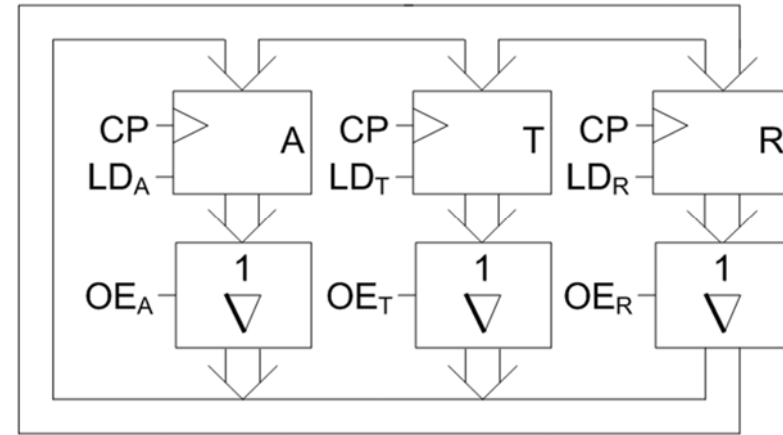
# Register med "three-state" utgång



Tre beteckningar:

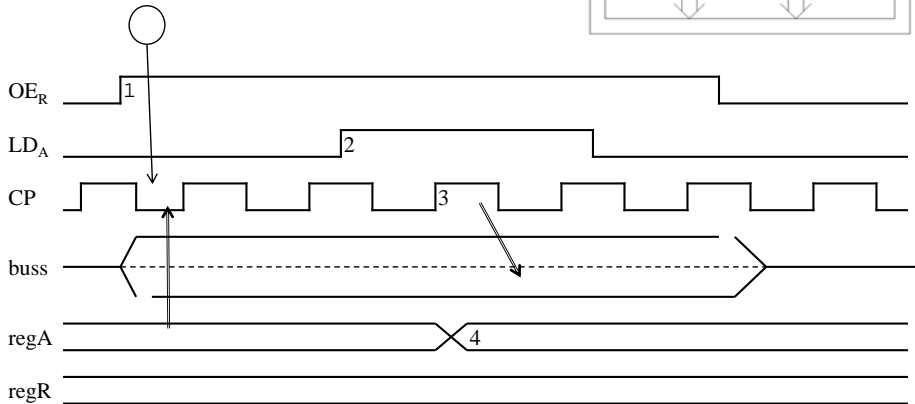
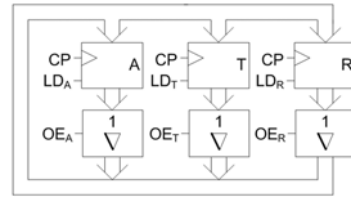
- CP: Clock Pulse, kretsens klockingång.
- LD<sub>R</sub>: Load Enable R, styrsignal för att kunna ladda register R
- OE<sub>R</sub>: Output Enable R, styrsignal för att överföra innehållet i R till utgången.

# Register med gemensam buss



Endast EN "three-state"-buffer får aktiveras (0 eller 1) åt gången. Övriga måste vara i Z-tillstånd.

# Exempel: (R)→A



# Dataväg med enkel styrenhet

- Omkopplarna ställs i lägen
- Ge klockpuls
- Upprepa...

### Uppgift 10.6

Du ska nu undersöka vad som händer om du aktiverar två (eller flera) OE-signalerna samtidigt.

Placera värdet 5C<sub>16</sub> i register A och värdet 21<sub>16</sub> i register T. Lägg ut båda dessa registerinnehåll till bussen och studera bussens värde i displaymodulen.

Vilket resultat får du?

Vad kan dra för slutsatser om bussens värde då flera moduler samtidigt driver bussen (skriver ut på bussen)?

