

F10: Minneselement

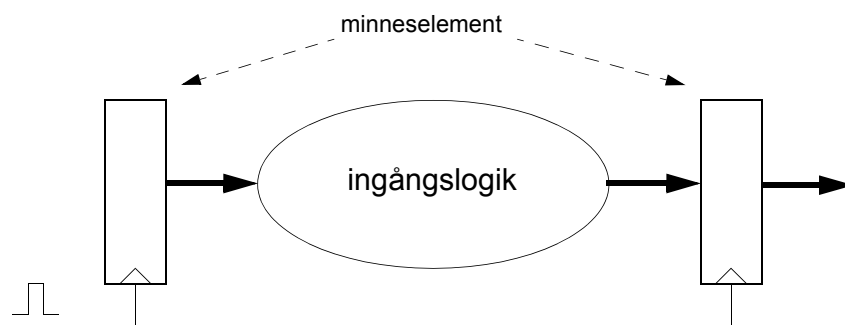
- **Målsättning**
 - Ge en bakgrund till de olika klockstrategier som finns
 - Visa hur latchar och register implementeras i CMOS
 - Visa vad som ingår i cyckeltiden i ett klockat system
- **Innehåll**
 - Klockade system
 - Latchar och register
 - System timing
 - Setup och hålltider

1 (17)

Klockat system

- **Antaganden för klockade system:**

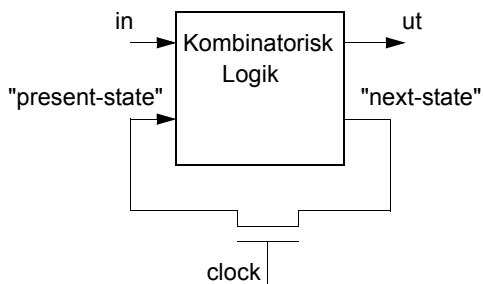
Minneselementen i en klockad sekventiell krets får endast ändra tillstånd en gång per klockperiod och dessa tillståndsförändringar måste ske samtidigt. Följt av en tillståndsförändring måste varje grind i ingångslogiken till minneselementet nått ett stabilt värde innan nästa tillståndsförändring triggas.



2 (17)

Klockning med dubbelsidigt timingkrav

• Tillståndsmaskin med "narrow pulse clocking"



1. Klockpulsen måste vara tillräckligt kort för i jämförelse med den snabbaste vägen genom kombinatoriska logiken.

2. "present-state" måste ändras snabbare än klockpulsens bredd. Fördröjningen i uppdateringen är i transistorn ($R_n \cdot C_{in}$)

Två krav på klockpulsens finns (dubbelsidigt villkor)

Fördel: enkelt (dynamiskt) minneselement, endast en transistor.

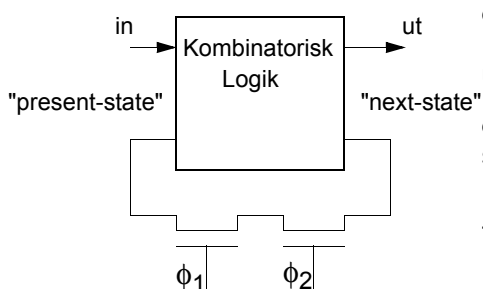
Nackdel: Ej rimligt att kunna uppfylla den dubbelsidiga relationen i en konstruktion under alla tankbara variationer i process, temperatur och matningsspänning.

Alltså: den här tekniken är i praktiken inte användbar.

3 (17)

Två-fas klockning med icke-överlappande klockfaser

• För att undvika den dubbelsidiga relation krävs (minst) två klockfaser



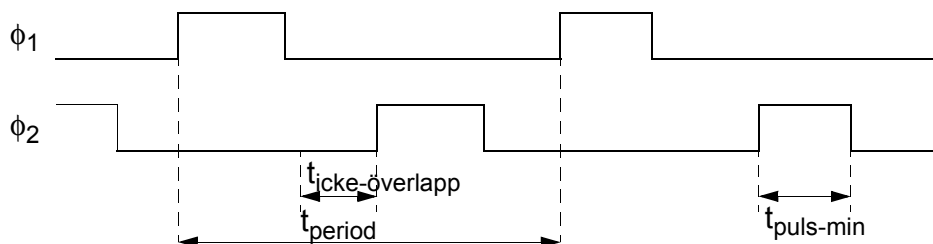
ϕ_1 aktiv: "present-state" uppdateras.

Kombinatoriska logiken sätter upp data på utgångarna oberoende när ϕ_1 stänger.

ϕ_2 aktiv: "next-state" samplas och måste vara stabilt en tid före ϕ_2 stänger.

Enkelsidig relation:

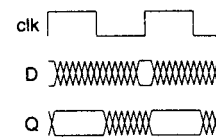
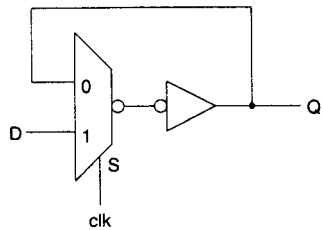
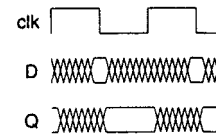
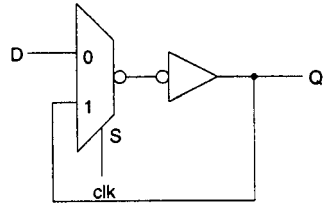
$$T_{\text{period}} > t_{\text{logik}} + t_{\text{clock-output}} + t_{\text{setup}} + t_{\text{icke-överlapp}}$$



4 (17)

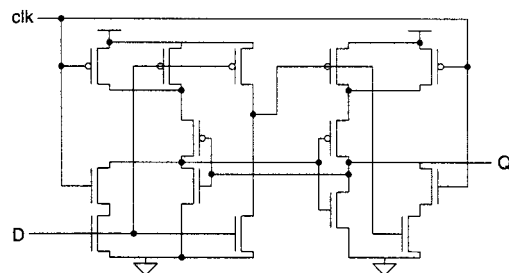
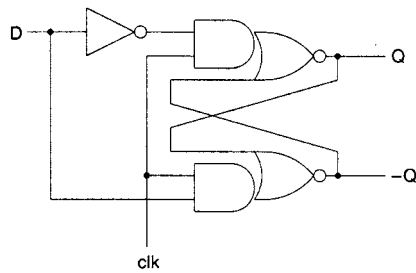
D-latchar - kombinatoriska

- Multiplexer-baserad



5 (17)

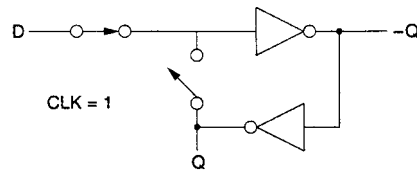
- SR-latch baserad



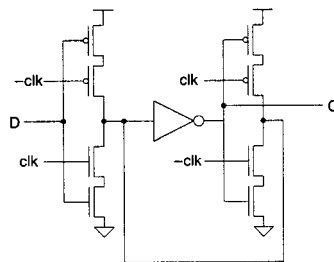
6 (17)

Latchar - TG baserade

- Logikschema

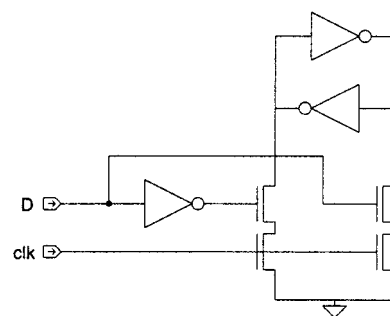


- Transistorschema



7 (17)

Latchar RAM-cells baserade



- **Fördel**

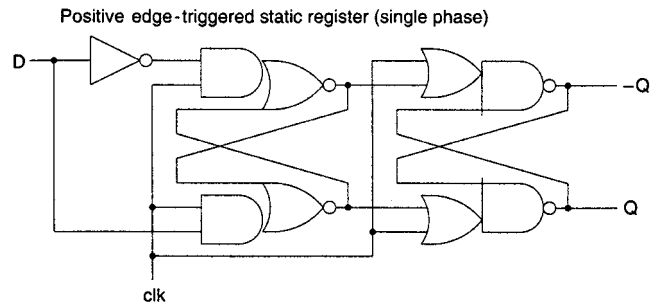
snabb och kompakt

- **Nackdel**

Kräver noggrann dimensionering av transistorerna för att få en robust konstruktion

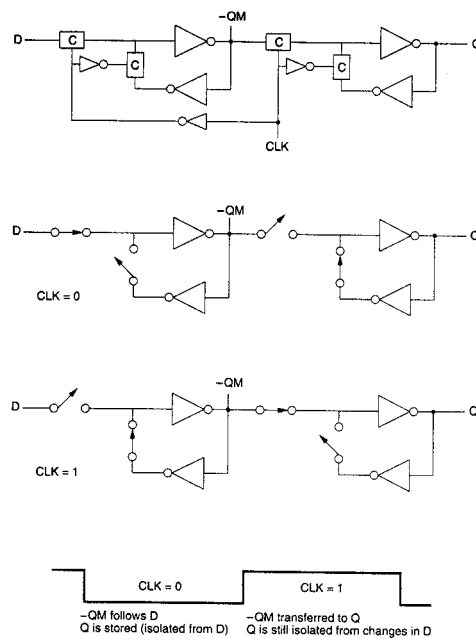
8 (17)

DFF (D Flip/flop) - kombinatoriska



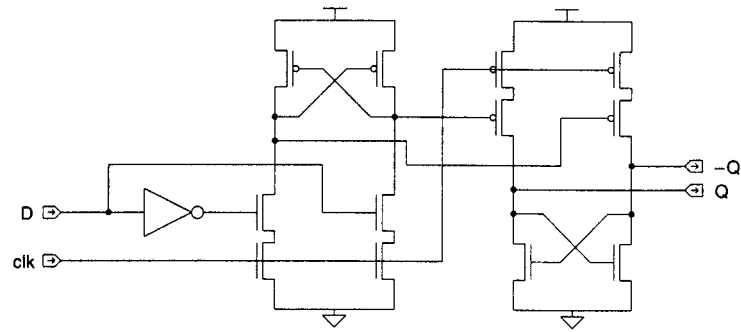
9 (17)

DFF - TG baserade



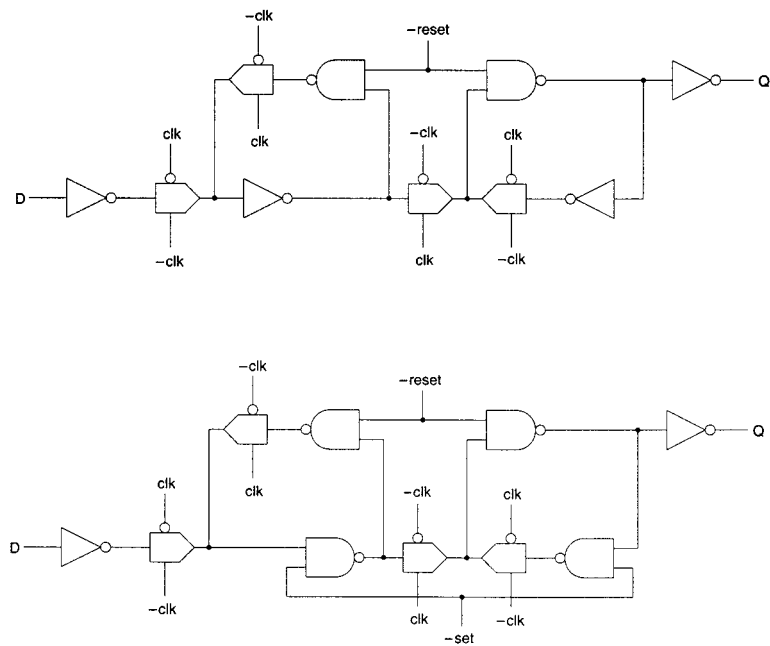
10 (17)

DFF - CVSL baserad



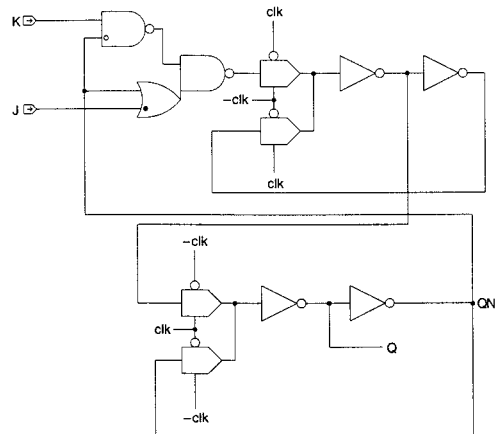
11 (17)

DFF med set och reset



12 (17)

JK-vippa

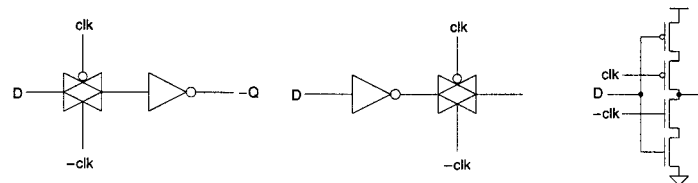


J	K	clk	Q	QN
0	0	↑	Q	QN
0	1	↑	0	1
1	0	↑	1	0
1	1	↑	QN	Q

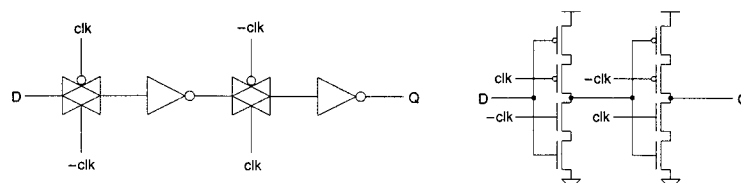
13 (17)

Dynamiska latchar och DFF baserade på C²MOS

Dynamisk latch

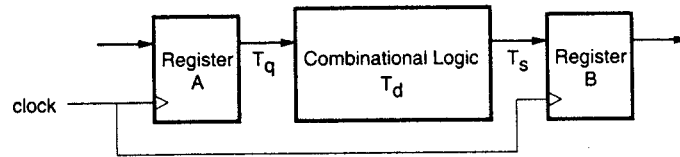


Dynamisk DFF



14 (17)

Registerbaserad pipelining



maximal fördröjning i logiken ges av $t_{logic} < T_{c,low} - T_q - T_s$

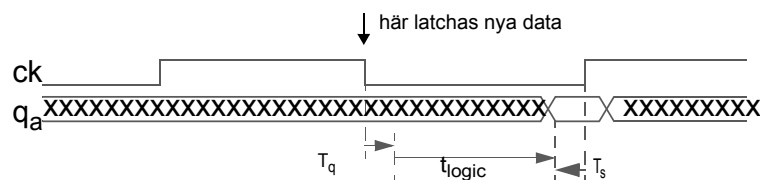
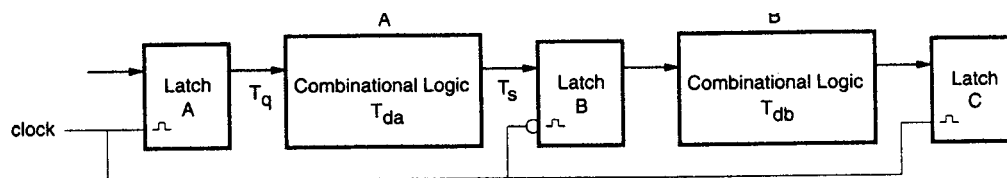
Cykeltiden ges av: $T_c = T_q + T_d + T_s$

där

T_q är *clock-to-output* fördröjning, T_d = fördröjning i logiken och T_s är setup-tiden.

15 (17)

Latch-baserad pipelining



Cykeltiden ges av: $T_c = T_{da} + T_{db} + 2 \cdot (T_q + T_s)$

där

T_q är *clock-to-output* fördröjning, T_d = fördröjning i logiken och T_s är setup-tiden.

Observera att en latch är hälften så stor som en DFF.

16 (17)

Setup och hålltider

- **Setup tid**

beror på fördröjningar i datavägen hos latchen eller DFF

- **Hålltid**

beror på fördröjningar i klockvägen hos latchen eller DFF

