

P9

UNTAI SEQUENTIAL

Hindiantoro

hindiantoro.wordpress.com

hindiantoro.uksw@gmail.com

085854777772

Pengenalan Untai Sequential

- ◎ Untai Sequential adalah untai nalar yang keluarannya ditentukan tidak hanya oleh nilai masukan saat itu, tetapi juga oleh nilai masukan pada saat-saat sebelumnya.
- ◎ Hal ini menunjukkan bahwa untai nalar sekuensial mengandung unsur ingatan (memori).

Multivibrator

- ◉ Multivibrator adalah untai nalar dengan dua keluaran yang selalu berbeda, dimana salah satu atau keduanya dalam keadaan stabil.
- ◉ State/Keadaan Multivibrator
 - Set (1) : $Q = 1$ dan $\bar{Q} = 0$
 - Reset (0): $Q = 0$ dan $\bar{Q} = 1$



Jenis Multivibrator

1. Mono-Stable Multivibrator (MSMV)
Rangkaian ini mempunyai satu keadaan stabil dan satu keadaan tak stabil.
2. A-Stable Multivibrator (ASMV)
Dua keadaan (Set & Reset) dari rangkaian ini adalah tak stabil, sehingga tanpa sinyal picu dari luar, rangkaian secara periodik akan berubah dari keadaan SET ke RESET dan sebaliknya.
3. Bi-Stable Multivibrator (BSMV)/Flip-Flop
Rangkaian ini dapat stabil baik pada keadaan SET maupun RESET. Untuk merubah dari keadaan yang satu ke keadaan yang lain diperlukan sinyal Picu dari luar.
Dapat berfungsi sebagai memory 1 bit
Disebut Latch karena informasi terkunci di dalamnya.

Jenis Flip-Flop

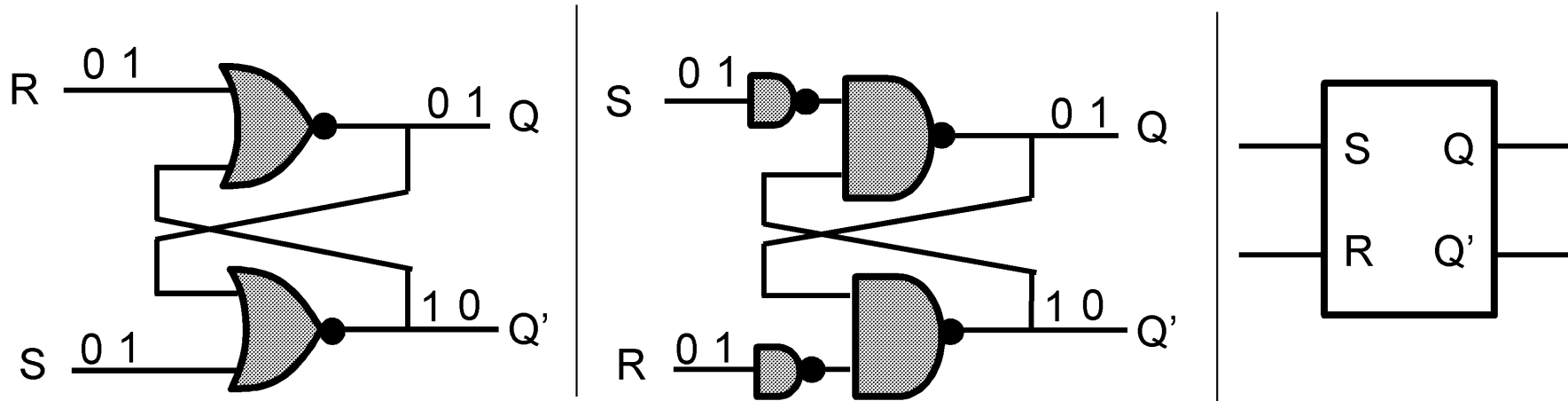
- ◎ SR Flip Flop (Set & Reset Flip Flop)
- ◎ JK Flip Flop
- ◎ D Flip Flop/ D-Latch (Data/Delay Flip Flop)
- ◎ T Flip Flop (Toggle Flip Flop)

SR Flip-Flop

- ◎ SR FF adalah flip-flop paling sederhana dan merupakan dasar bagi FF yang lain. Dapat disusun dari “2 NAND/NOR Gate” yang feedbacknya dihubungkan saling menyilang.
- ◎ Sesuai dengan namanya input S (Set) digunakan untuk mengubah FF ke keadaan $Q = 1$, sedangkan input R (Reset) digunakan untuk mengubah FF ke keadaan $Q = 0$.
- ◎ Kalau $S = R = 0$, dapat dibuktikan FF dapat dalam keadaan set ($Q = 1$) atau dalam keadaan reset ($Q = 0$)
- ◎ Operasinya disebut *Transparent Latch* karena output akan merespon input dengan cara mengunci nilai input yang diberikan atau mengingat input tersebut.

Cont

⦿ Rangkaian, simbol dan tabel kebenaran adalah sebagai berikut :



S	R	Q_n	Q_{n+1}	Q'_{n+1}
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	0/1	0/1
1	1	1	0/1	0/1



S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	*

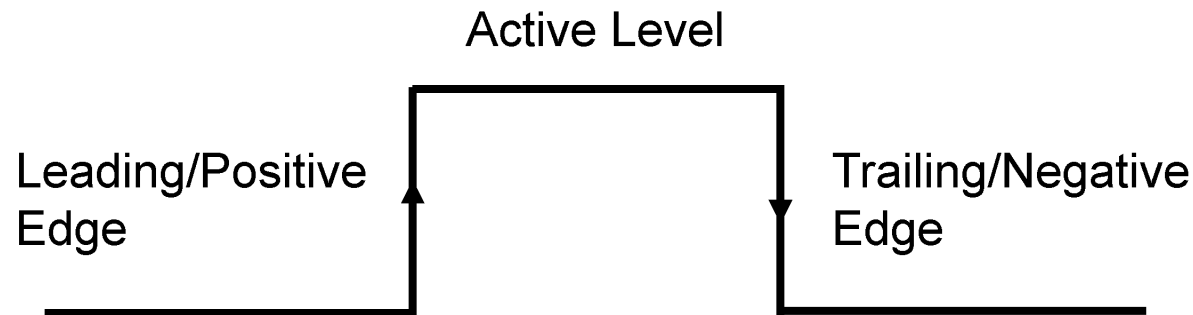
→ Tetap
 → Reset
 → Set
 → Tidak dipakai

S = R = 1, Tidak boleh
 Q_n = Output Sekarang
 Q_{n+1} = Output YAD

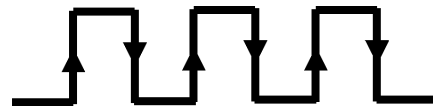
Tidak Boleh

Timing Diagram & Clock

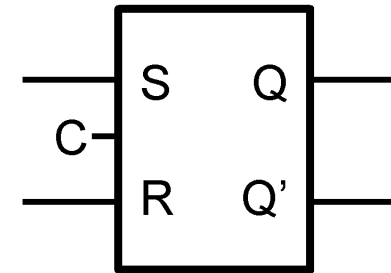
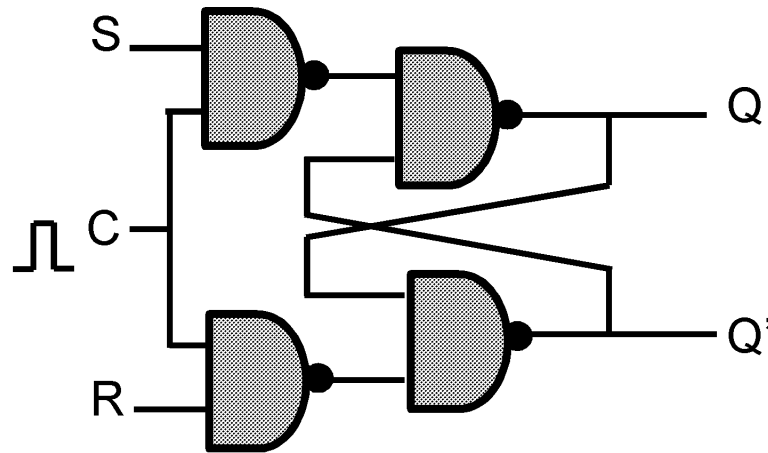
- Timing Diagram adalah diagram yang menggambarkan bentuk sinyal/pulsa input dan output suatu rangkaian logika bersesuaian dengan waktu.



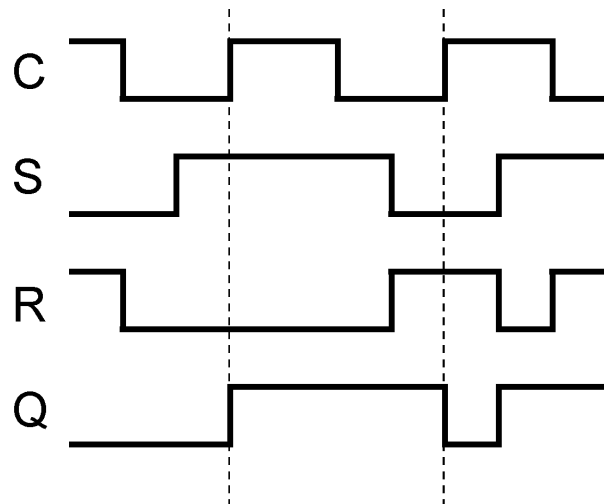
- Clock adalah sinyal pulsa periodik yang biasanya digunakan untuk mengontrol agar perubahan output terjadi secara serentak
- Perubahan terjadi pada saat Leading edge atau trailing edge atau saat aktif, sesuai dengan karakteristik device.



Clocked SR FF

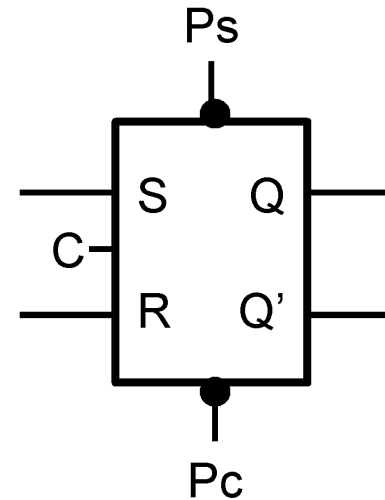
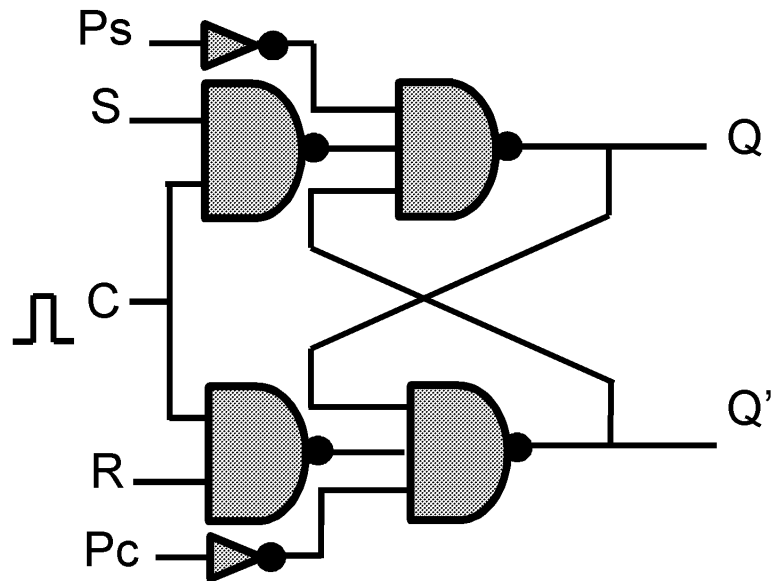


Timing Diagram (Perubahan saat C aktif)



SR FF dengan Ps dan Pc

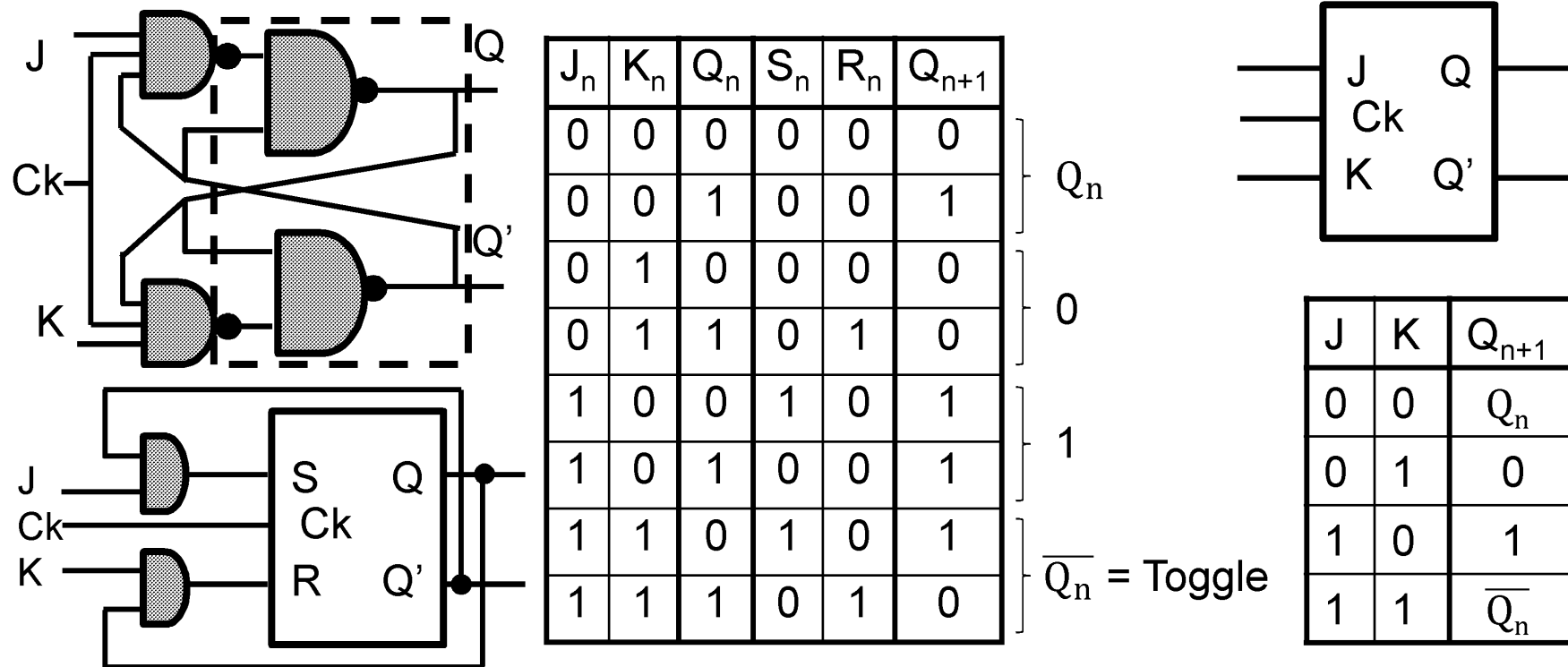
- ⦿ Preset untuk men set $Q = 1$ dan Pc untuk memberikan nilai $Q = 0$ tanpa melihat kondis input saat itu.



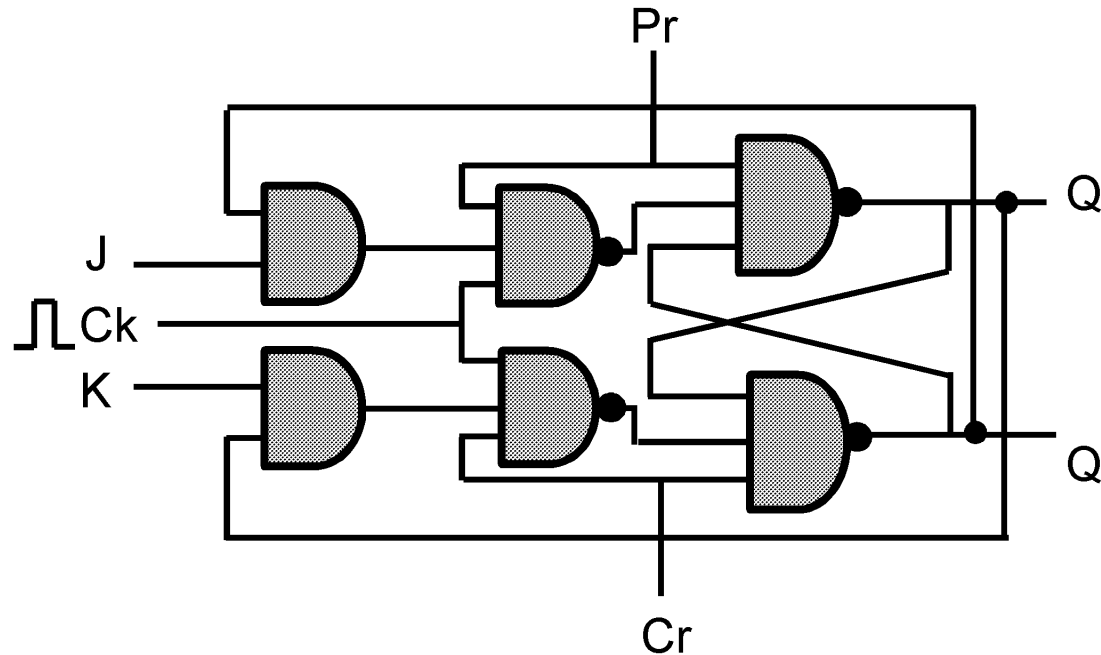
- ⦿ $P_s = 1 \rightarrow Q = 1$
- ⦿ $P_c = 1 \rightarrow Q = 0$
- ⦿ Kondisi kerja : $P_s = 0, P_c = 0$

JK FF

- JK FF merupakan pengembangan SR FF
- Mengizinkan kedua masukan berharga "1" dengan menambahkan gerbang "AND", sehingga nilai SR tidak pernah keduanya berharga "1"



JK FF dengan Preset dan Clear



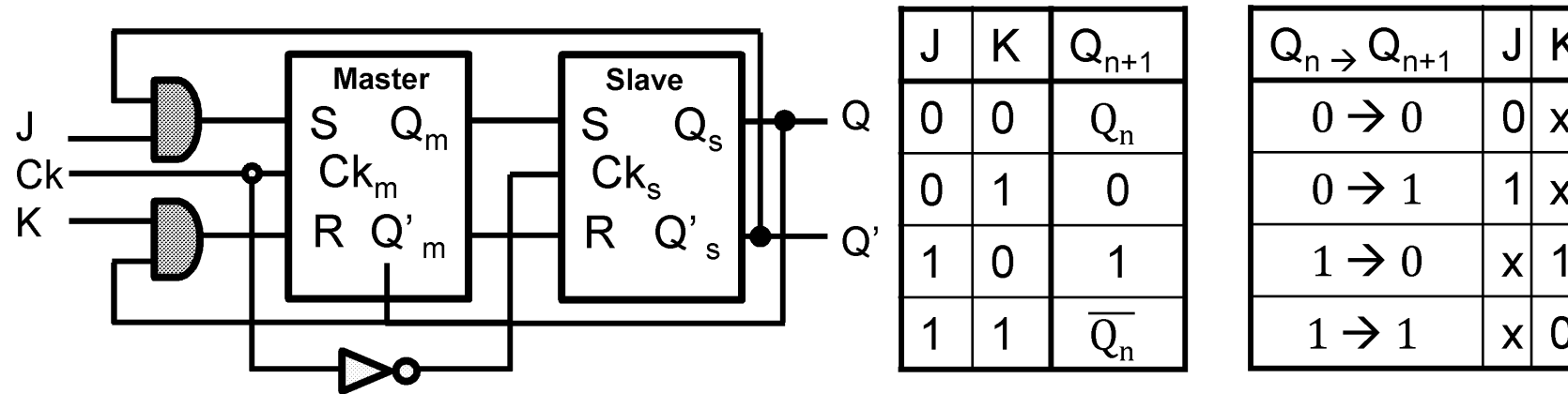
- ⦿ $Pr = 0, Cr = 1 \rightarrow Q = 1$
- ⦿ $Pr = 1, Cr = 0 \rightarrow Q = 0$
- ⦿ $Pr = 0, Cr = 0 \rightarrow Q = 1, Q' = 1$ ← Harus dihindari
- ⦿ $Pr = 1, Cr = 1 \rightarrow$ Tidak merubah Q, JK FF bekerja Normal

Master Slave JK FF

- ◉ *Race Round* yaitu keadaan saat $J = K = 1$, dimana Q akan berubah terus nilainya, dan ini terjadi jika lebar pulsa clock (t_p) lebih besar dari waktu untuk merubah output (t_d).
- ◉ Sebenarnya t_d cukup besar dan keadaan ini sulit terjadi, namun demikian untuk menghindari keadaan ini dibuatlah Master Slave JK FF

Cont.

⦿ Rangkaian :



⦿ Saat $Ck = 1$:

- Master ON, Input enable, Q_m berubah sesuai input JK
- Slave OFF, Output Disable, Q tetap/Hold

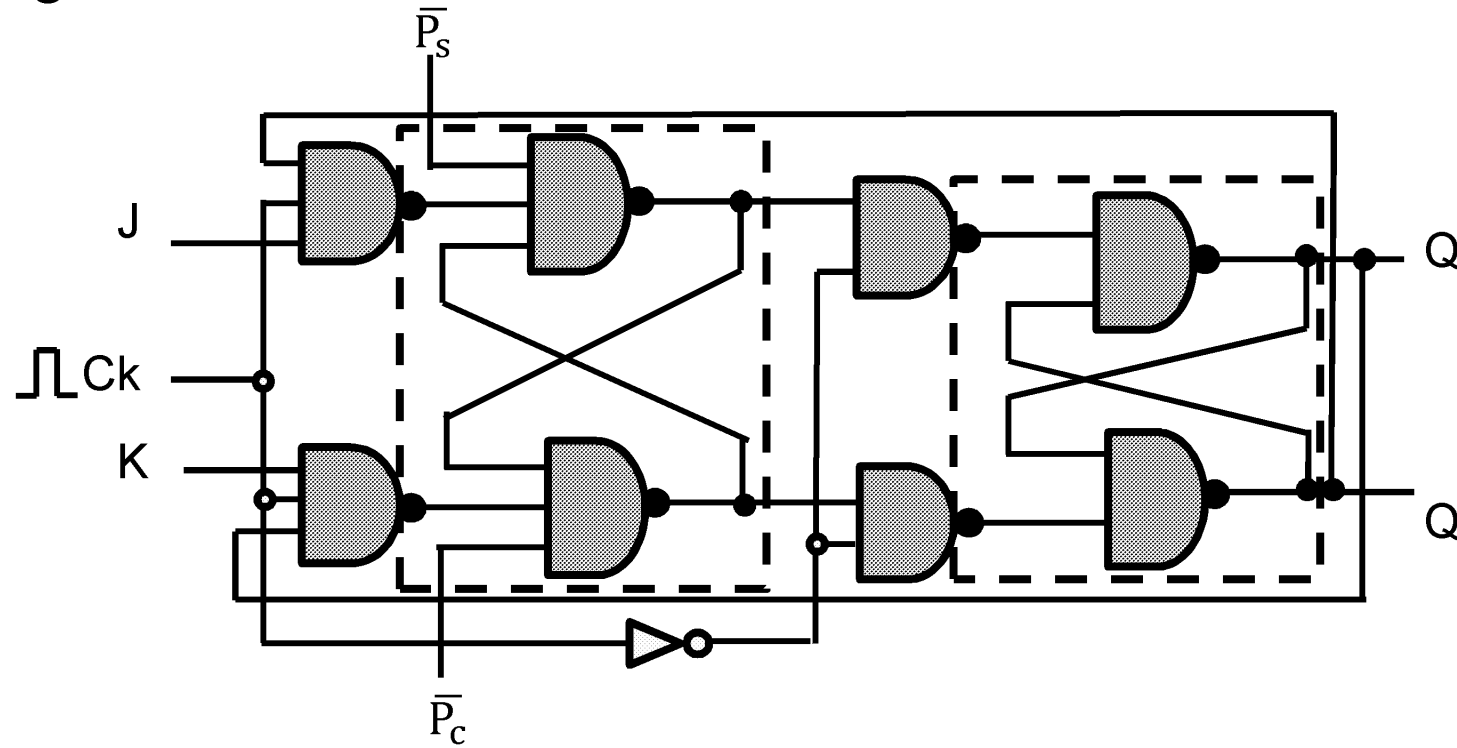
⦿ Saat $Ck = 0$:

- Master OFF, Input Disable, Q_m tetap sesuai keadaan terakhir
- Slave ON, Output Enable, Q sesuai nilai Q_m

⦿ Dengan rangkaian seperti ini akan menghindari Race Round

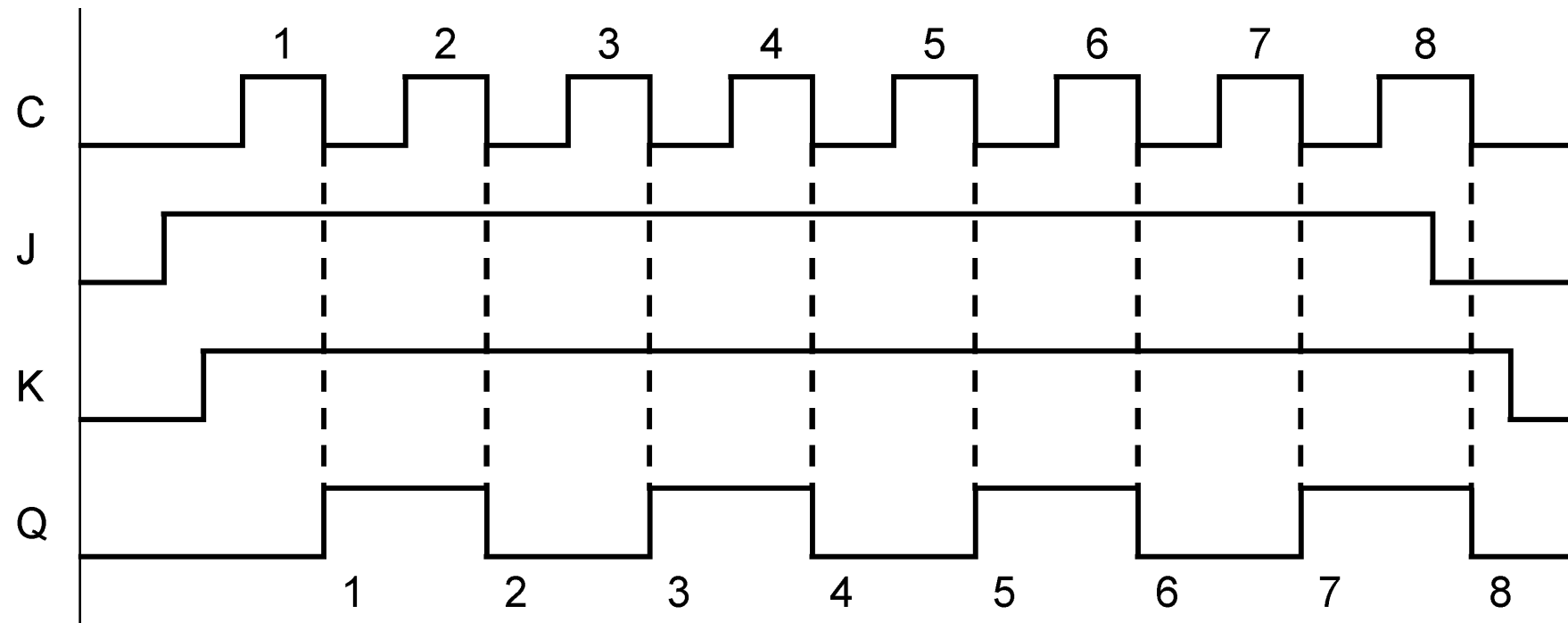
Master-Slave JK FF dengan P_s dan P_c

◎ Rangkaian Dasar



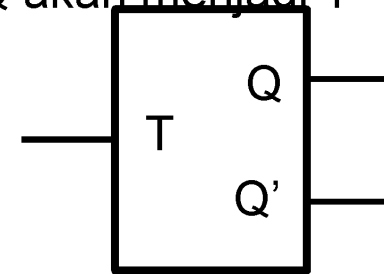
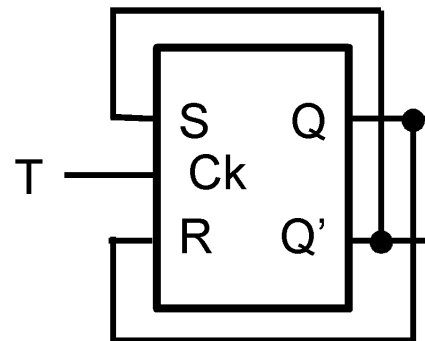
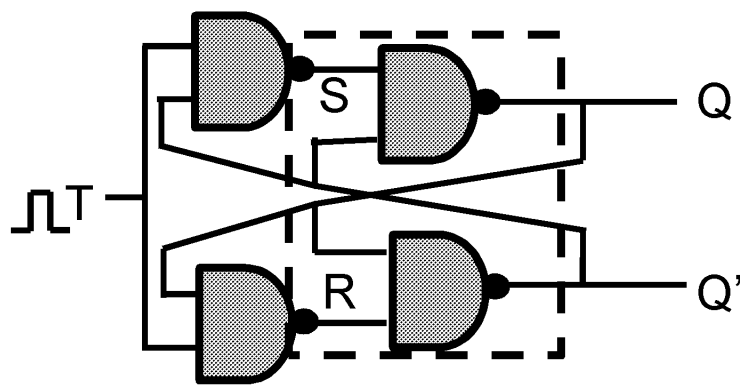
Timing Diagram JK FF

- ⦿ Aktif saat pulsa turun (trailing edge)

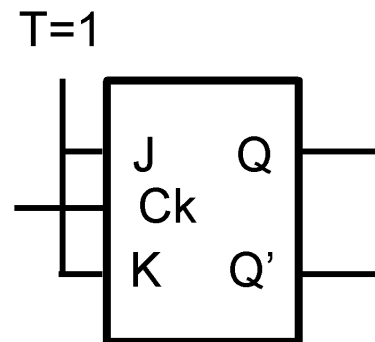
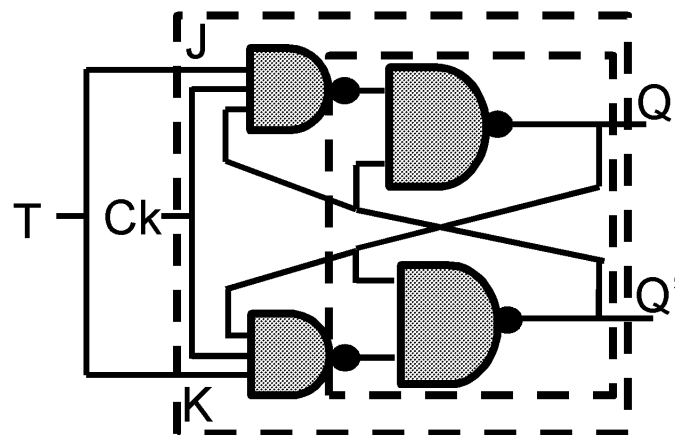


T FF

- T FF juga dapat dibentuk dengan SR FF dimana input S dihubungkan dengan Q'
- Saat T = 0 atau, keluaran tetap
- Saat $T = 1$ dan $S = 0$, R dapat tersambung ke clock, akan berubah
- Saat $T = 1$ dan $S = 1$, R = 0, jika terdapat pulsa clock, Q akan menjadi 1



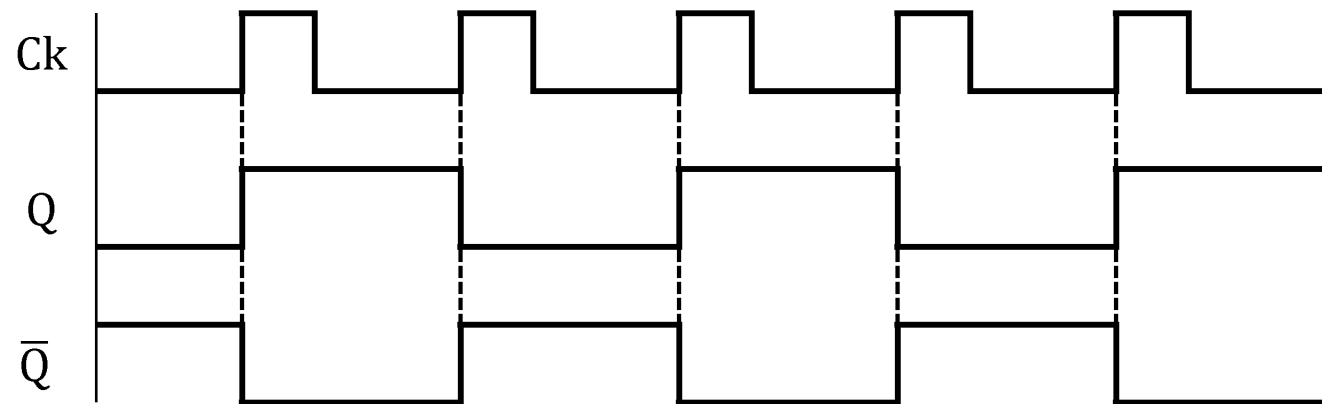
T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0



$Q_n \rightarrow Q_{n+1}$	T
$0 \rightarrow 0$	0
$0 \rightarrow 1$	1
$1 \rightarrow 0$	1
$1 \rightarrow 1$	0

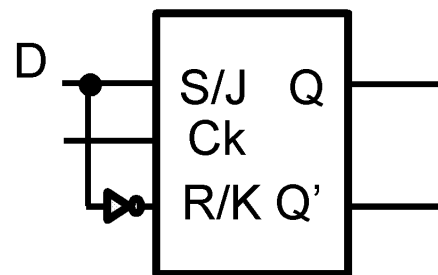
Cont.

Timing Diagram



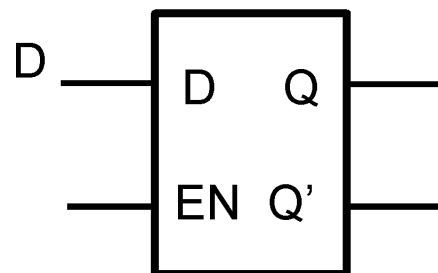
D FF

- Dari SR dan JK FF, terlihat jika nilai S (J) dan R (K) berlawanan, maka setelah clock, keluaran akan sama dengan nilai SR (JK).
- Dengan prinsip di atas dibuatlah D (Delay) FF dengan memberikan gerbang NOT antara S (J) dan R (K), dimana keluarannya akan sama dengan masukannya (D) setelah terjadi clock.



D_n	D_{n+1}
0	0
1	1

D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1



Active High

