

BAB 6

SISTEM DIGITAL

Sistem Digital adalah ilmu yang mempelajari tentang peralatan yang memproses informasi secara digital (bukan analog).

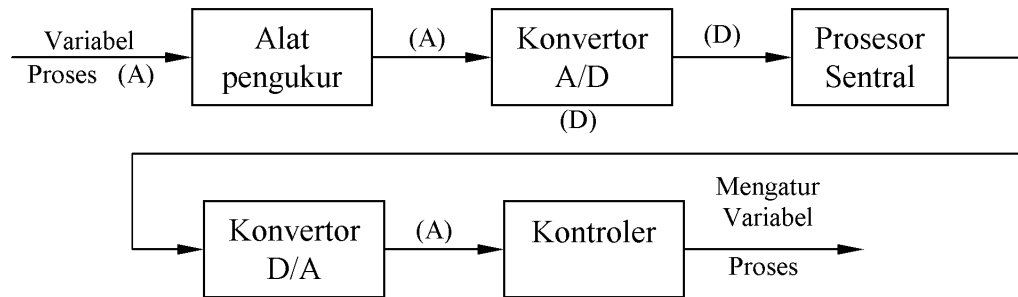
6.1. Representasi Bilangan

Pada dasarnya ada dua cara dalam menyatakan nilai bilangan dari suatu kuantitas, yaitu secara analog dan digital.

- Representasi Analog
Dalam hal ini suatu kuantitas dinyatakan dengan kuantitas lain yang berbanding lurus dengan kuantitas pertama tersebut. Jadi mereka dapat berubah secara bertingkat pada suatu rentang harga kontinue.
- Representasi Digital
Kuantitas-kuantitas tidak dinyatakan dengan kuantitas-kuantitas sebanding, tetapi dengan simbol-simbol yang disebut digit yang berubah secara diskrit (Step demi Step)

6.2. Sistem Digital, Analog Dan Hybrid

- Sistem Analog
Suatu kombinasi peralatan (listrik, mekanik, fotolistrik, dsbnya) yang memproses informasi yang masuk secara analog.
- Sistem Digital
Suatu kombinasi peralatan yang memproses informasi yang masuk secara digital.
Beberapa keuntungan penggunaan sistem digital :
 1. Kecepatan dan kecermatan yang lebih besar
 2. Kemampuan memory
 3. Tidak terpengaruh oleh perubahan karakteristik komponen dari sistem tersebut.
 4. Dapat digunakan pada rentang pemakaian yang lebih luas.
- System Hybrid
Merupakan gabungan sistem analog dan sistem digital dalam suatu peralatan.



6.3. Sequential Circuits

- Komponen digital :
 - Logika Kombinasi (*Combinational Logics*)
 - Logika Sekuensial (*Sequential Logics*)
- Flip - Flop :
 - Flip-Flop (FF) merupakan komponen logika sekuensial dan menjadi komponen utama dalam rangkaian sekuensial (*sequential Circuit*), misalnya : Register, counter, transfer data, dan sebagainya.
 - FF banyak digunakan sebagai rangkaian memori dalam rangkaian sekuensial.
- Memori :
 - Rangkaian untuk “mengingat “ atau “memori” adalah salah satu bagian yang penting dari sebuah komputer.
 - Memori dapat menyimpan (*storage*) suatu informasi (data maupun perintah) untuk selama waktu yang diperlukan, dan disimpan, suatu saat informasi tersebut dapat diambil kembali sewaktu-waktu.
 - Ada 2 (dua) macam “ Memori” yaitu :
 1. Memori yang tidak mudah hilang/menguap (*non volatile*)
 2. Memori yang mudah hilang/menguap (*volatile*)
- Memori tipe “ Non Volatile“

Dapat menyimpan atau mengingat suatu informasi untuk waktu yang lama dan bahkan bila sumber listriknya sudah diputuskan, jenis ini masih dapat menyimpan informasi tersebut dengan baik.

Contoh : Magnetic tapes, disc dan cores.

– Memori tipe “Volatile”

Hanya dapat menyimpan informasi selama sumber listriknya masih belum diputuskan dan bila sumber listrik diputuskan maka informasi yang disimpan akan hilang atau terhapus lagi.

Ada dua macam memori tipe Volatile, yaitu :

1. *Static memory*, yang bekerja atas dasar arus balik (*feedback*) dari beberapa *gate* yang saling dihubungkan menyilang, sehingga akan memberikan suatu keadaan yang tetap (stabil).
Contoh : Flip-flop yang dapat menyimpan informasi dalam bentuk digit-digit bilangan biner (“0” dan “1”)
2. *Dinamic Memory*, yang bekerja atas dasar penyimpan listrik pada kondensator dan informasi yang disimpan makin lama makin cacat atau rusak, sehingga pada periode waktu tertentu perlu diperbaiki lagi.

6.4. Untai Nalar Sekuensial

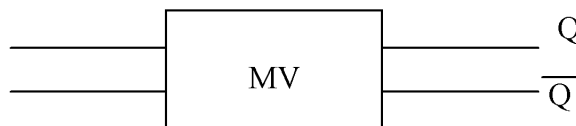
Untai nalar sekuensial adalah suatu untai nalar yang keluarannya ditentukan tidak hanya masukan saat itu, tetapi juga oleh harga masukan pada saat-saat sebelumnya. Hal ini menunjukkan bahwa untai nalar sekuensial mengandung unsur ingatan (memori).

6.4.1. Multivibrator

Suatu rangkaian nalar yang mempunyai/memiliki dua keluaran yang keadaannya senantiasa berbeda dimana salah satu atau keduanya dapat dalam keadaan stabil.

Terdapat 2 (dua) macam keadaan Multivibrator :

1. Keadaan Set (1) $\Rightarrow Q = 1$ dan $\overline{Q} = 0$
2. Keadaan reset (0) $\Rightarrow Q = 0$ dan $\overline{Q} = 1$



Yang termasuk rangkaian multivibrator adalah:

1. Mono-Stable Multivibrator (MSMV)

Rangkaian ini mempunyai satu keadaan stabil dan satu keadaan tak stabil. Untuk mengubah dari keadaan stabil ke keadaan tak stabil diperlukan sinyal picu, yang selanjutnya setelah beberapa saat pada keadaan tak stabil ini rangkaian akan kembali dengan sendirinya ke keadaan stabil, keadaan ini (stabil) pada umumnya adalah keadaan RESET.

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.

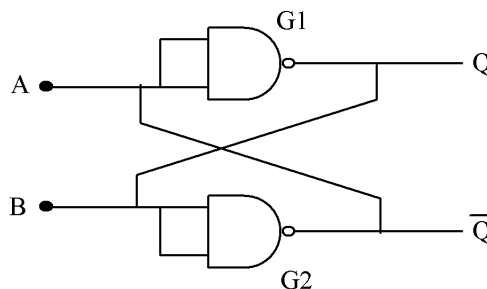
3. Bi-Stable Multivibrator (BSMV)

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.

6.4.2. BI STABLE MULTIVIBRATOR (BSMV) / Flip - Flop

Flip - Flop (FF) dapat berfungsi sebagai penyimpan data 1 bit, sehingga rangkaian ini memegang peranan penting di dalam untai logika sekuensial (Memori 1 bit / latch).

Rangkaian Dasar memori 1 bit.



Latch terdiri dari dua buah gerbang NOT (gerbang NAND masukan tunggal) G1 dan G2, keluaran dari suatu gerbang di umpan balikkan ke masukan pada gerbang yang lain. Kombinasi umpan balik ini disebut FLIP - FLOP.

Sifat penting yang dimiliki FF adalah FF tersebut hanya mempunyai 2 keadaan stabil, yaitu :

$$(Q = 1, \bar{Q} = 0) \Rightarrow \text{Keadaan Set}$$

$$(Q = 0, \bar{Q} = 1) \Rightarrow \text{Keadaan Reset}$$

Contoh :

- Jika keluaran dari G1 adalah $Q = '1'$, maka B, masukan ke G2 juga logika '1'.
- Gerbang G2, yang berfungsi sebagai pembalik akan menghasilkan keluaran \bar{Q} pada logika = '0', karena Q dihubungkan ke A, maka masukan ke G1 juga '0' dan keluaran Q pada logika = '1'.
- Sehingga, $Q = 1, \bar{Q} = 0$ merupakan salah satu keadaan stabil.
- Dapat dibuktikan, bahwa $Q = 0, \bar{Q} = 1$ adalah juga keadaan stabil.
- Tetapi Q dan \bar{Q} tidak mungkin mempunyai keadaan sama (keduanya logika 1 atau 0)
- Itulah sebabnya, Flip - Flop disebut sebagai untai biner atau untai bi-stabil (bistable Circuit). Disebut memori 1 bit karena menyimpan satu bit informasi. Disebut Latch karena informasi tersebut terkunci di dalamnya.

6.5. FLIP - FLOP

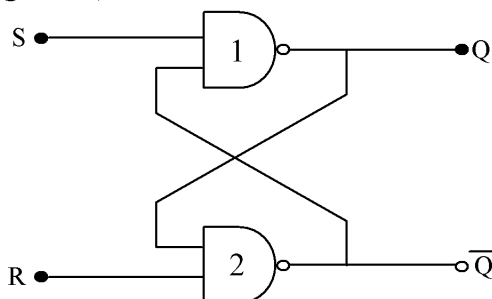
Ada empat macam FF yang paling banyak dikenal :

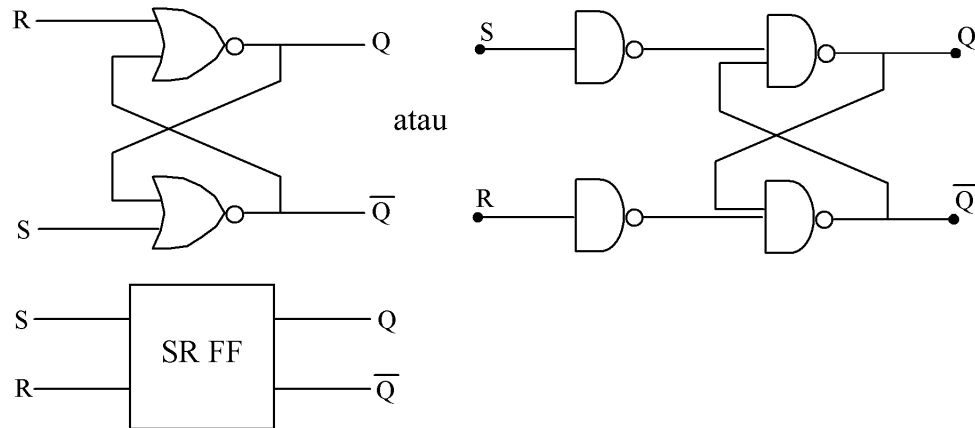
1. S - R FF (Set - Reset)
2. J - K FF
3. T - FF (Trigger atau Toggle)
4. D - FF (Delay atau Latch)

6.5.1. S-R Flip-Flop

SR FF adalah flip-flop yang paling sederhana dan merupakan dasar bagi FF yang lain. Dapat disusun / dibangun dari "2 Nand Gate" atau "2 Nor Gate" yang *feedback*nya dihubungkan saling menyilang.

Rangkaian, simbol dan tabel kebenaran adalah sebagai berikut :





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 ($Q = 0$)

Sehingga dapat dilihat bahwa output FF pada suatu saat tidak hanya tergantung pada harga inputnya saat itu, tetapi juga tergantung pada harga output sebelumnya.

- Tabel Kebenaran SR FF :

S	R	Q_n	Q_{n+1}	Q'_n
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	1	1
1	1	1	1	1

→ Tetap

→ Reset

→ Set

Tdk boleh

$S = R = 1$, Tidak boleh

Tidak diperkenan

Q_n = Output sebelum input diubah/Q mula-mula

Q_{n+1} = Output sesudah input diubah

Dari tabel kebenaran di atas, dapat disimpulkan bahwa kalau masukannya diubah menjadi :

1. $S = R = 0$ Maka $Q_{n+1} = Q_n$
2. $S = 0, R = 1$ Maka $Q_{n+1} = 0$
3. $S = 1, R = 0$ Maka $Q_{n+1} = 1$
4. $S = 1, R = 1$ Maka $Q_{n+1} = Q'_{n+1}$, Hal ini bertentangan dengan Q'_{n+1} , oleh karena itu masukan $S = R = 1$ perlu dihindari.

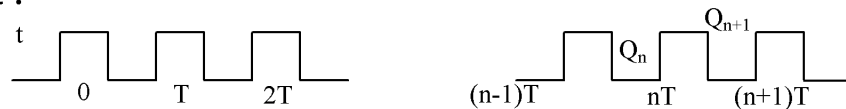
S	R	Q_{n+1}
0	0	don't care
0	1	0
1	0	1
1	1	Q_n

← don't care → diabaikan karena outputnya tidak dapat ditentukan

← no change – kondisi menyimpan (storage state)

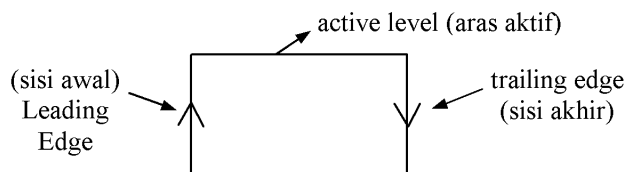
Untuk memberikan kemungkinan perubahan keluaran flip-flop yang lain, perlu ditambahkan masukan ketiga yang disebut masukan klok (“Clock Input”) sehingga perubahan keluaran FF hanya akan terjadi kalau pada masukan klok dimasukkan sinyal pulsa.

Pada umumnya sinyal pulsa terjadi secara periodik seperti pada gambar berikut :



- Diagram Waktu

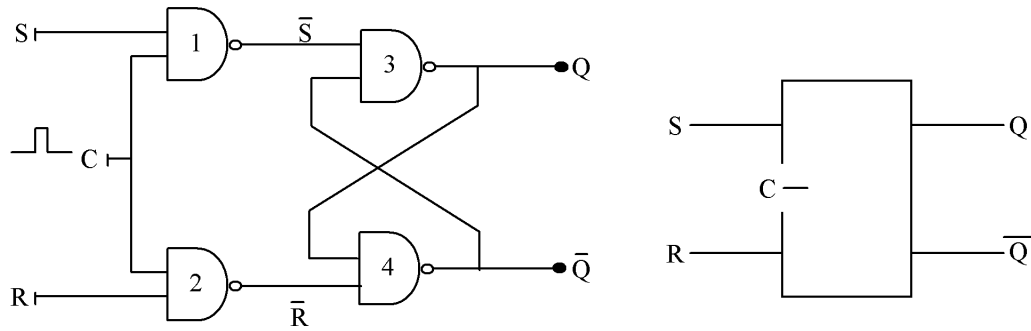
Diagram waktu (*Timing Diagram*) adalah diagram yang menggambarkan bentuk-bentuk sinyal/pulsa input dan output suatu rangkaian logika



- Clock S-R FF

Agar output S-R FF berubah secara bersamaan dan pada saat-saat tertentu yang dikehendaki, digunakan sinyal pengontrol yang disebut "Clock" yang akan mengubah output S-R FF pada saat $C = 1$ dan pada saat $C = 0$ tidak berubah.

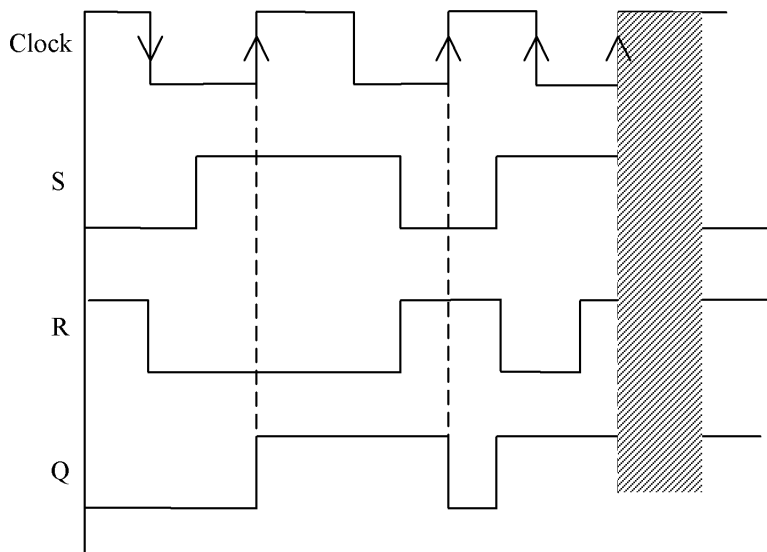
Rangkaian logika, simbol dan tabel kebenarannya adalah sebagai berikut :



S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	don't care

Q_n = Adalah " Q " sebelum clock
 Q_{n+1} = Adalah " Q " sesudah clock

- Diagram waktu " S-R FF "



Tabel kebenaran dari clocked SR FF sama dengan tabel kebenaran dari SR FF. Q_n dan Q_{n+1} berturut-turut menyatakan keluaran sebelum dan sesudah pulsa klok pada saat $t = nT$ terjadi.

Kalau klok = 0, maka keluaran dari gerbang 1 dan 2 selalu “1” tidak tergantung pada harga S dan R. Pada keadaan ini keluaran Q dapat “0” atau “1”.

Andaikan: Mula-mula pada saat klok = 0, $Q = 0$; selama klok tetap “0” keluaran tidak berubah, harga S dan R berubah.

Andaikan: Pada saat klok berubah menjadi 1, harga $S = 1$, $R = 0$, maka keluaran gerbang 1 = 0, keluaran gerbang 2 = 1.

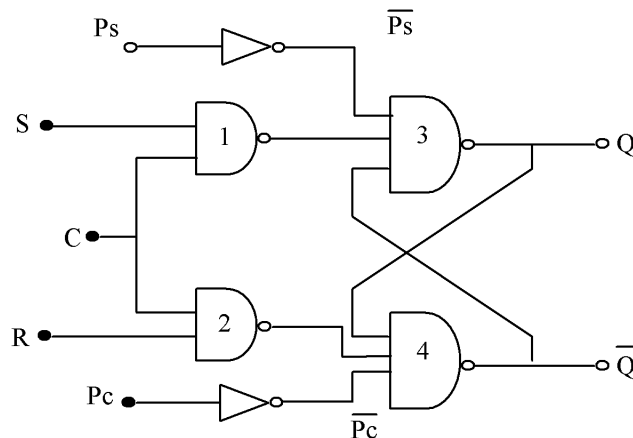
Oleh karena masukan dari gerbang 3 salah satunya “0”, maka Q berubah menjadi “1”, dengan demikian masukan dari gerbang 4 keduanya “1” sehingga Q berubah menjadi “0”.

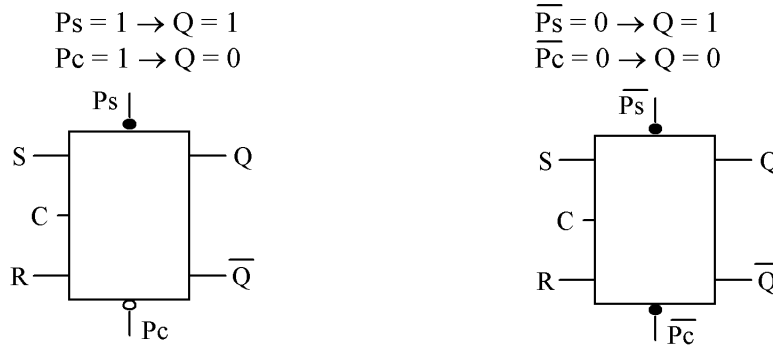
Selanjutnya keluaran FF tidak berubah kalau klok berubah menjadi “0” lagi.

Dari diagram waktu clock S-R FF tersebut dapat diambil beberapa kesimpulan sebagai berikut :

- Output “Q” akan segera berubah pada saat “Leading Edge” sinyal clock.
- Pada saat clock masih “1”, bila “S” dan “R” di ubah, maka “Q” juga berubah sampai “Trailing Edge” sinyal clock.
- Pada saat “C = 0”, maka “ $Q^{n+1} = Q^n$ ”

S-R FF ini dapat dilengkapi dengan “Ps” (Preset) dan “Pc” (PreClear), bila menginginkan men-set ($Q = 1$) atau me-reset ($Q = 0$) output lebih awal tanpa melihat kondisi input FF pada saat itu.



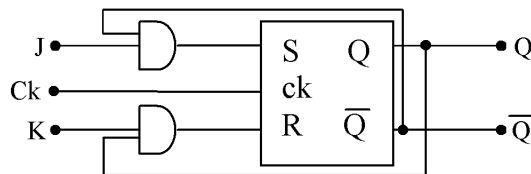
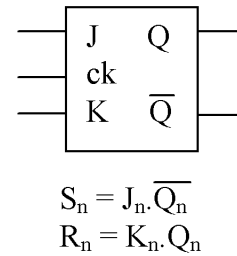


6.5.2. J-K FLIP - FLOP

Pada SR FF, kedua masukan tidak boleh berharga “1”, agar kedua masukan dapat berharga “1” sedang harga Q nya masih tetap dapat berlawanan dengan harga \overline{Q} , maka pada masing-masing masukan SR FF perlu ditambahkan gerbang AND. Flip - flop yang merupakan pengembangan dari SR FF ini disebut JK FF

Tabel Kebenaran JK FF :

J_n	K_n	Q_n	S_n	R_n	Q_{n+1}	
0	0	0	0	0	0	} Q_n
0	0	1	0	0	1	
0	1	0	0	0	0	} 0
0	1	1	0	1	0	
1	0	0	1	0	1	} 1
1	0	1	0	0	1	
1	1	0	1	0	1	} $\overline{Q_n}$
1	1	1	0	1	0	



Dari tabel di atas, dapat dilihat bahwa S_n dan Q_n tidak pernah keduanya berharga “1”, sehingga $Q = Q'$ dapat dihindari.

Tabel Kebenaran JK FF sama dengan SR FF kecuali untuk $J = K = 1$, dimana harga Q_{n+1} berlawanan dengan Q_n .

Pada JK FF, perubahan keluarannya terjadi kalau pulsa klok = 1 (High Level Pulsa), sehingga kalau inputnya berubah-ubah selama klok = 1, maka outputnya juga berubah-ubah, dan harga Q_{n+1} akan ditentukan oleh harga J_n , K_n dan Q_n . Sesaat sebelum pulsa klok turun menjadi "0". Jadi seharusnya J dan K dipertahankan tetap selama klok = 1, hanya saja meskipun masukan dipertahankan tetap, kesulitan terjadi jika pulsa klok terlalu lebar (Terlalu lama pada harga "1")

Contoh :

Untuk $J = K = 1$ dan $Q = 0$ (baris 7 dalam tabel), Q akan berubah menjadi "1" kalau klok = 1, dan karena kemudian, sekarang $Q = 1$ sedang J dan K tetap 1 (baris 8), maka kalau tetap klok = 1, Q akan berubah kembali ke "0". Demikian seterusnya selama $J = K = 1$ dan klok = 1, Q akan berubah dari 0 ke 1 dan sebaliknya secara terus menerus; Kemudian kalau klok berubah menjadi "0", harga Q menjadi tidak menentu. Kejadian ini disebut gejala : Race Around.

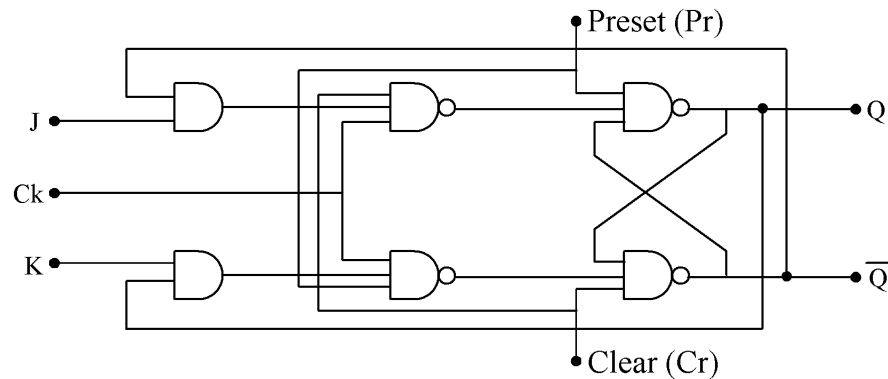
Gejala *Race Around* tidak akan terjadi jika lebar pulsa (t_p) lebih kecil dari waktu yang diperlukan untuk berubahnya keluaran (t_d), tetapi hal ini sukar dicapai karena orde besar dari t_d adalah puluhan nano (10^{-9}) detik. Untuk menghindari kesulitan ini rangkaian terintegrasi ini dari JK FF pada umumnya menggunakan rangkaian *Master Slave*.

PRESET Dan CLEAR

Pada umumnya FF dalam bentuk terintegrasi (IC) dilengkapi dengan masukan preset dan clear atau hanya clear saja. Dengan masukan ini , keluaran FF dapat diubah tanpa menunggu terjadinya pulsa klok.

Kegunaannya adalah untuk menentukan keadaan awal dari FF atau untuk mereset FF sewaktu-waktu.

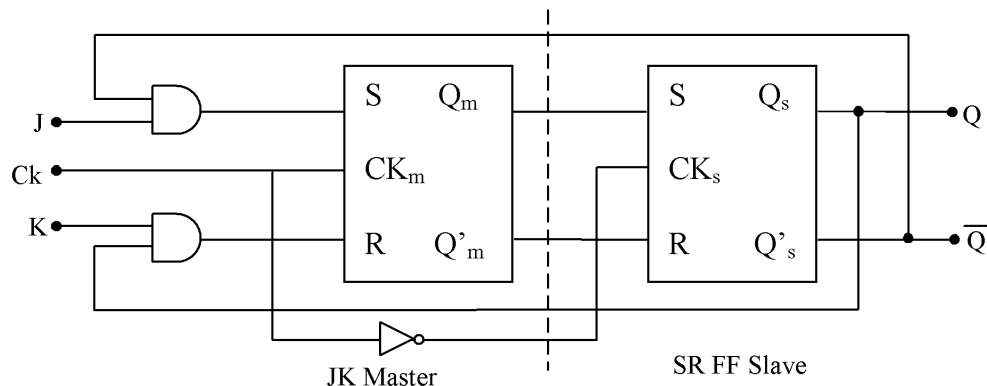
- Rangkaian JK FF dengan Preset dan Clear :



- Dari rangkaian di atas, dapat dibuktikan Kalau $Pr = 0$, $Cr = 1$ maka $Q = 1$, kalau $Pr = 1$ $Cr = 0$ maka $Q = 0$, kalau $Pr = Cr = 0$ maka $Q = Q' = 1$ (harus dihindari), Sedang kalau $Pr = Cr = 1$, keluarannya tidak berubah dan JK FF akan bekerja sebagai biasanya. Sehingga perlu diingat, sesudah mengadakan pengaturan dengan input preset dan clear, input harus dikembalikan ke keadaan $Pr = Cr = 1$

MASTER SLAVE JK FF

Master Slave JK FF dibuat untuk menghindari terjadinya gejala *race around*. Rangkaianannya terdiri dari JK FF sebagai master dan SR FF sebagai Slave.



Keluaran yang dikembalikan ke JK FF bukanlah keluarannya sendiri Q_m atau Q'_m , melainkan keluaran dari SR FF, yang juga merupakan keluaran dari master slave JK FF secara keseluruhan.

Dengan adanya gerbang 'not', maka $C_k = C_{km} \neq C_k's$. Dengan demikian kalau mula-mula $C_k = 0$ karena C_{km} juga 0 maka Q_m dan $Q'm$ tetap, meskipun ada perubahan pada masukan. Sedang pada keadaan ini karena $C_{ks}=1$ maka kalau $S = Q_m = 0$ dan $R = Q'm = 1$ maka $Q = Q_s = 0$. Jika $S = Q_m = 1$ dan $R = Q'm = 0$ maka $Q = Q_s = 1$. Atau pada saat $C_k = 0$ maka keluaran dari master tetap sedang keluaran dari slave mengikuti keluaran masternya. Kalau kemudian C_k berubah menjadi 1 maka Q_m berubah sebagai fungsi dari J_n . K_n dan Q_n (yang sama dengan Q_m) sesuai dengan tabel kebenaran JK FF. Pada saat ini, karena $C_{ks} = 0$ maka Q harganya masih tetap sehingga mencegah terjadinya gejala race around; baru setelah C_k berubah menjadi 0 kembali, maka Q akan berubah sama dengan Q_m . Jadi perubahan dari keluaran master slave JK FF sama dengan perubahan JK FF, hanya saja perubahan keluarannya terjadi bukan pada saat pulsa klok naik tapi sebaliknya pada saat turun kembali menjadi 0.

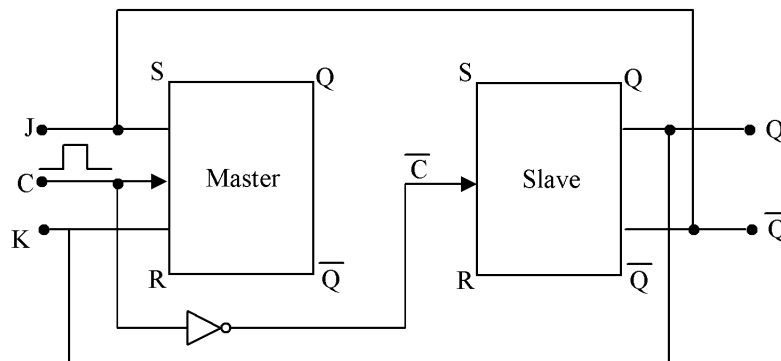
Perlu Diingat

Bahwa kalau $C_k = 1$, Q_m akan berubah-ubah kalau masukan (J,K) diubah-ubah, sehingga sebaiknya pada saat $C_k = 1$, harga dari input dibuat tetap.

J-K FF dibangun dari 2 - SR FF, di mana :

- S-R FF pertama disebut " Master "
- S-R FF Kedua disebut " Slave "

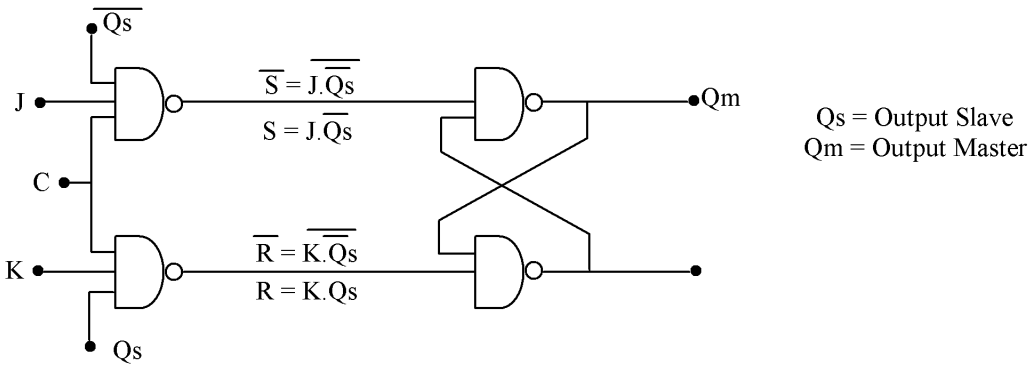
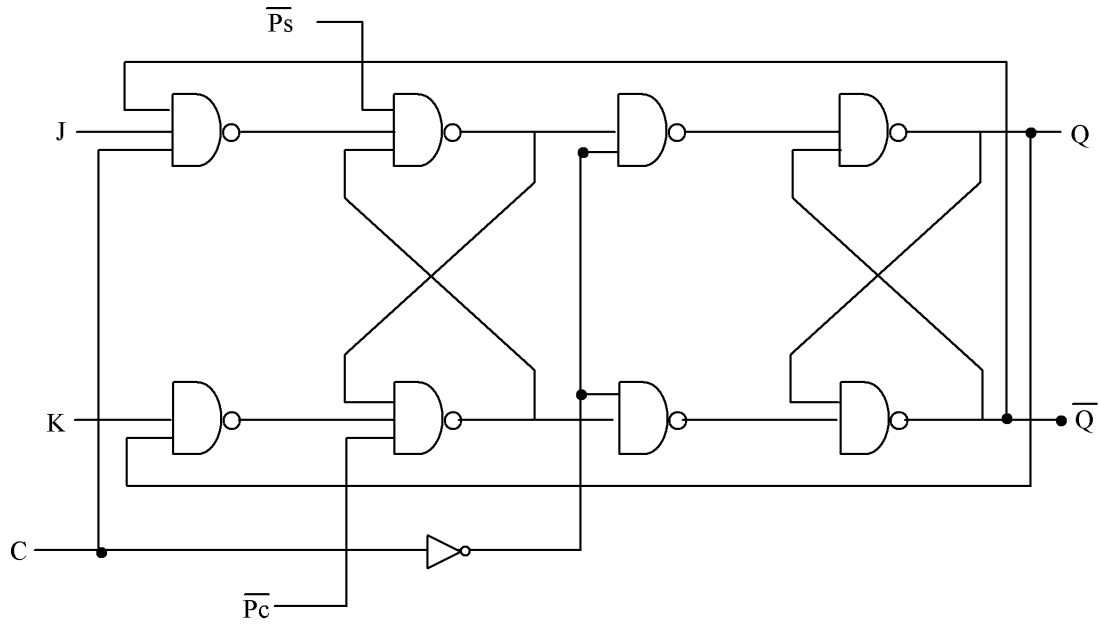
➤ Prinsip dasar J-K FF



J	K	Q^{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

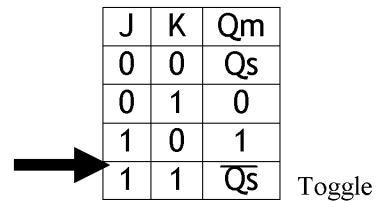
$\overline{Q_n} = \text{Toggle}$

- Rangkaian J-K FF



J	K	Q^{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	\bar{Q}_n

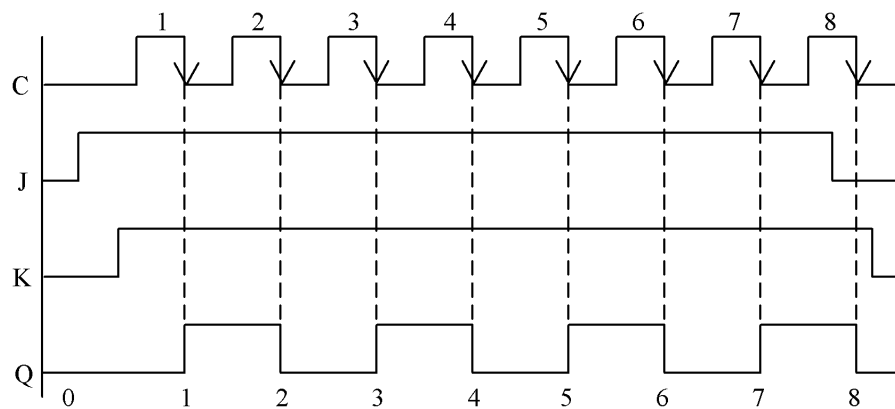
J	K	Q_s	S	R	Q_m
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	0	0
0	1	1	0	1	0
1	0	0	1	0	1
1	0	1	0	0	1
1	1	0	1	0	1
1	1	1	0	1	0



- Transition List J-K FF

$Q_n \rightarrow Q^{n+1}$	J	K
0 \rightarrow 0	0	x
0 \rightarrow 1	1	x
1 \rightarrow 0	x	1
1 \rightarrow 1	x	0

- Timing Diagram J-K FF

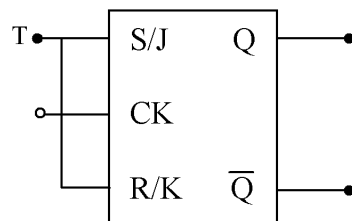


JK - FF dalam kondisi "TOGGLE" (J=1;K=1) sebagai "Pembagi - 2"

6.5.3. T FLIP - FLOP (TOGGLE FLIP - FLOP)

T-FF tidak lain adalah JK FF yang masukannya dihubungkan menjadi satu. Kalau $T = J = K = 0$, keluarannya tidak berubah setelah terjadi pulsa klok, sedang kalau $T = J = K = 1$, maka setelah terjadi pulsa klok keluarannya akan berubah berlawanan dengan keluarannya mula-mula.

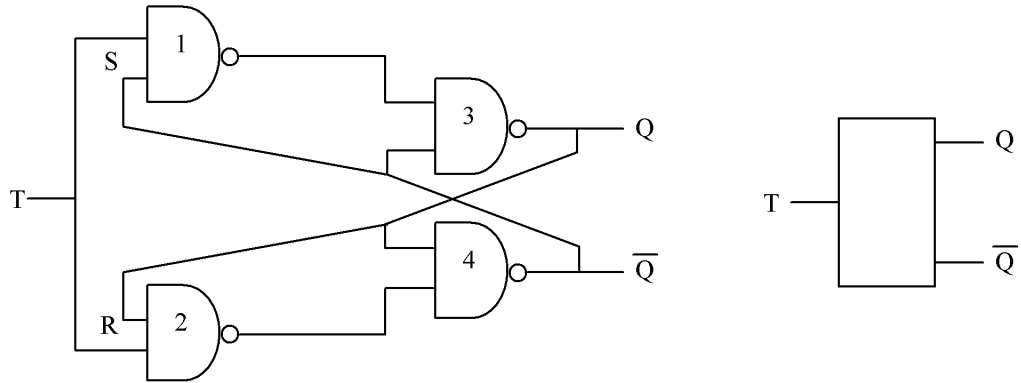
- Tabel kebenaran dan gambar dari TFF :



T_n	Q_{n+1}
0	Q_n
1	Q'_n

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

- Rangkaian T - FF

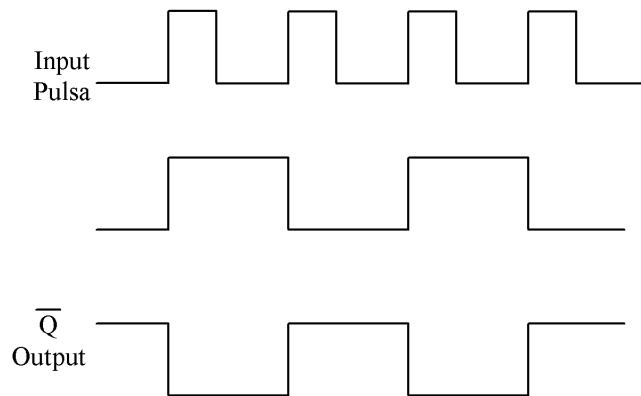


- Tabel Kebenaran

T	Q _n	Q _{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

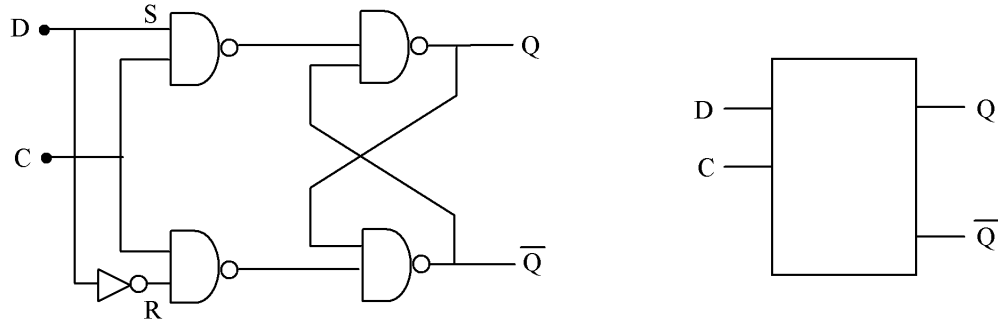
Q _n → Q _{n+1}	T
0 → 0	0
0 → 1	1
1 → 0	1
1 → 1	0

- Diagram Waktu

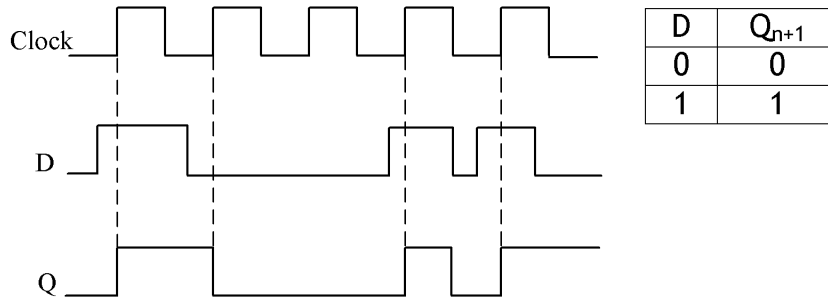


6.5.4. D - FF (DELAY FLIP-FLOP)

- Rangkaian D-FF



- Diagram Waktu



Dari tabel kebenaran SR - FF dan JK - FF dapat dilihat, jika harga S/J berlawanan dengan harga R/K, maka setelah pulsa klok harga outputnya akan sama dengan harga S/J.

Kalau masukan S/J dihubungkan dengan R/K lewat gerbang NOT seperti terlihat pada gambar dibawah ini, maka terbentuklah D (Delay) FF yang keluarannya setelah klok terjadi, sama dengan masukannya (D) atau dapat dikatakan keluarannya tertunda 1 waktu klok terhadap masukannya.

D - FF dapat dibentuk dari SR - FF atau JK - FF baik yang Master Slave ataupun bukan. Tabel kebenaran dari D - FF dan gambarnya sbb :

