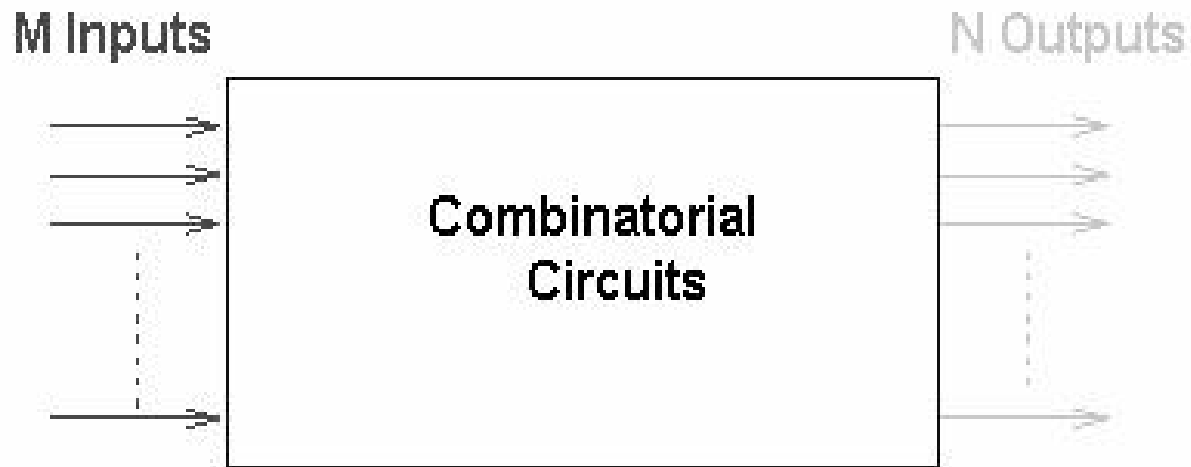




RANGKAIAN LOGIKA KOMBINASIONAL DAN SEQUENSIAL

Rangkaian Logika secara garis besar dibagi menjadi dua, yaitu Rangkaian logika Kombinasional dan rangkaian logika Sequensial. Rangkaian logika Kombinasional adalah rangkaian yang kondisi keluarannya (*output*) dipengaruhi oleh kondisi masukan (*input*). Struktur rangkaian kombinasional secara fisik adalah seperti gambar berikut:



3.1 RANGKAIAN LOGIKA KOMBISIONAL

3.1.1 ENKODER

Enkoder adalah rangkaian logika kombinasional yang berfungsi untuk mengkodekan suatu sinyal masukan diskrit menjadi kode biner. Hasil Keluarannya dinyatakan dengan aljabar boole. Sebuah Enkoder harus memenuhi syarat perancangan $m < 2^n$ (m : kombinasi masukan ; n : jumlah bit). Satu kombinasi masukan hanya dapat mewakili satu kombinasi keluaran. Perhatikan contoh tabel fungsi keluaran Enkoder berikut :

Input								Output		
i0	i1	i2	i3	i4	i5	i6	i7	Y2	Y1	Y0
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

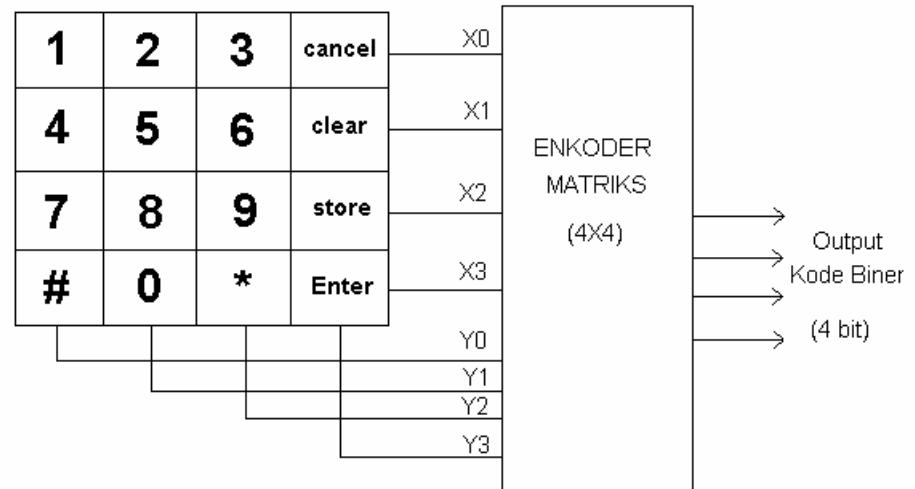
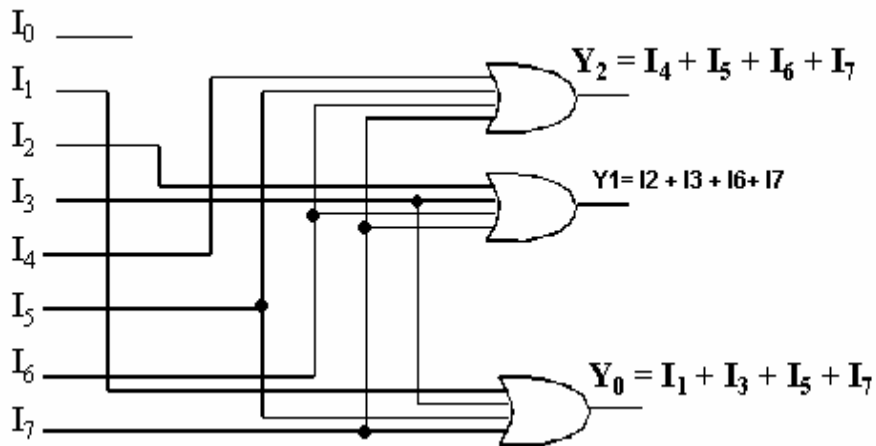
Dari tabel diatas, dapat dibuat fungsi keluaran sebagai berikut :

$$Y_0 = I_1 + I_3 + I_5 + I_7$$

$$Y_1 = I_2 + I_3 + I_6 + I_7$$

$$Y_2 = I_4 + I_5 + I_6 + I_7$$

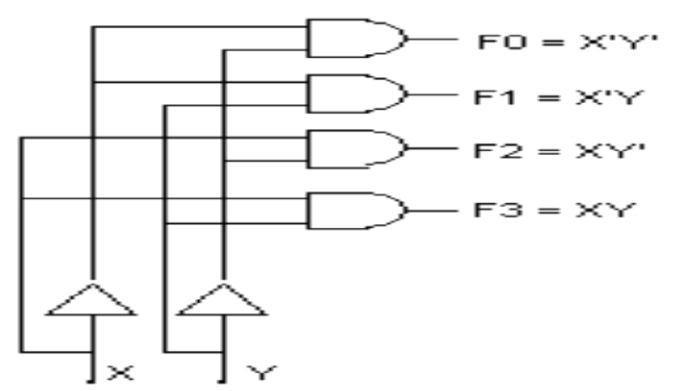
Dari persamaan tersebut, maka rangkaian gerbangnya dapat dibuat seperti pada gambar berikut :



3.1.2 DEKODER

Rangkaian Dekoder mempunyai sifat yang berkebalikan dengan Enkoder yaitu merubah kode biner menjadi sinyal diskrit. Sebuah dekoder harus memenuhi syarat perancangan $m \leq 2^n$ (m : kombinasi keluaran ; n : jumlah bit masukan). Satu kombinasi masukan hanya dapat mewakili satu kombinasi keluaran. Membuat suatu Enkoder dan dekoder dapat dilakukan dengan dua cara yaitu pertama, menggunakan gerbang-gerbang dasar yang disusun membentuk fungsi Enkoder atau dekoder, kedua, menggunakan IC Enkoder atau dekoder yang banyak terdapat dipasaran. Perhatikan contoh tabel fungsi keluaran dekoder berikut serta rangkaian gerbangnya :

X	Y	F0	F1	F2	F3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



3.2 RANGKAIAN LOGIKA SEQUENSIAL

Flip-flop adalah rangkaian utama dalam logika sequensial.

Counter, Register, Memory, serta rangkaian sequensial lainnya disusun

Dengan menggunakan flipflop sebagai komponen utama.

Flip-flop adalah rangkaian yang mempunyai fungsi pengingat (*memory*).

Artinya rangkaian ini mampu melakukan penyimpanan data sesuai dengan kombinasi masukan yang diberikan kepadanya.

Ada beberapa macam flip-flop yang akan dibahas

yaitu R-S flipflop, J-K flipflop, D flipflop, dan T flipflop.

Ciri utama dari flip-flop adalah keluaran Q dan \bar{Q} adalah selalu

Berlawanan / stabil (jika $Q = 0$ maka $\bar{Q} = 1$, jika $Q = 1$ maka $\bar{Q} = 0$).

Karena kondisi dua keadaan stabil ini rangkaian flipflop dinamakan juga dengan rangkaian bistabil.

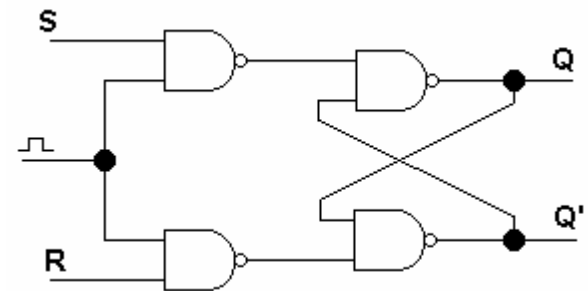
3.2.1 R-S FLIP-FLOP

Flipflop ini terdiri dari dua masukan, yaitu S (set) dan R (reset). Serta dua keluarannya yaitu Q dan \bar{Q} . Kondisi Set adalah kondisi ketika Q berlogika 1. Sedangkan kondisi Reset adalah kondisi ketika Q berlogika 0. Flipflop R-S tidak diperbolehkan / dilarang saat S = 1 dan R = 1.

Perhatikan tabel dibawah :

Input		Output	
S	R	Q_{n+1}	\bar{Q}_{n+1}
0	0	Q_n	\bar{Q}_n
0	1	0	1
1	0	1	0
1	1	Terlarang	

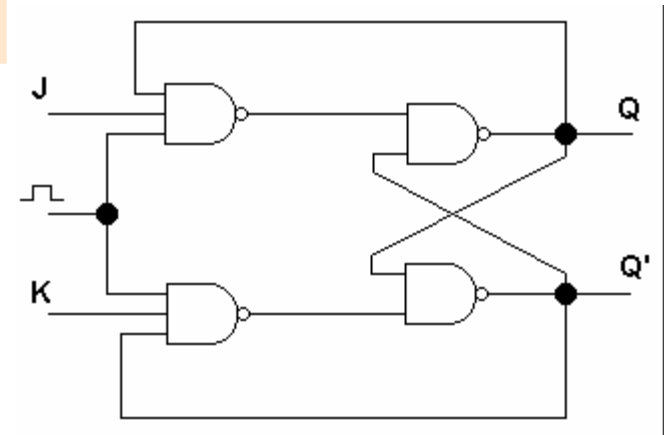
Flip-flop R-S dengan clock



3.2.2 J-K FLIP-FLOP

Flip-flop J-K merupakan penyempurnaan dari flip-flop R-S. Pada kondisi masukan $J = 1$ dan $K = 1$ akan membuat kondisi keluaran berlawanan dengan kondisi keluaran sebelumnya. Sementara untuk keluaran berdasarkan kondisi masukan yang lain semua sama dengan

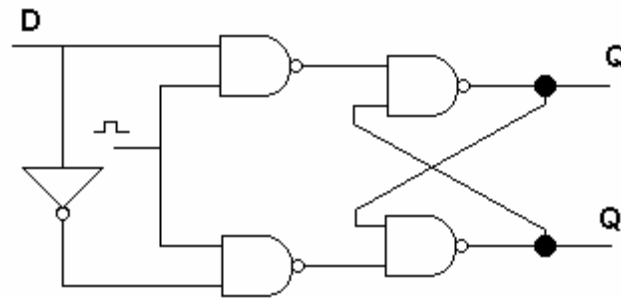
Input		Output	
J	K	Q_{n+1}	\bar{Q}_{n+1}
0	0	Q_n	\bar{Q}_n
0	1	0	1
1	0	1	0
1	1	\bar{Q}_n	Q_n



3.2.3 D FLIP-FLOP

Flip-flop D merupakan Flip-flop R-S yang memaksa untuk memiliki satu Masukan dengan R selalu berlawanan dengan S, sehingga kondisi masukan SR sama tidak akan pernah terjadi. Perhatikan gambar flipflop D berikut :

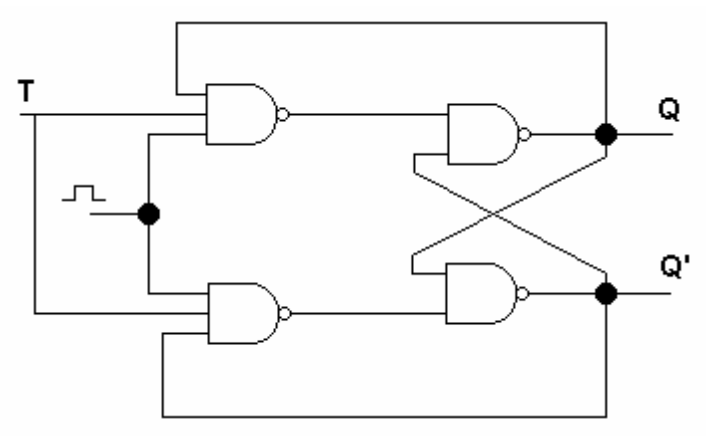
Input	Output
D	Q_{n+1}
0	0
1	1



3.2.4 T FLIP-FLOP

Flip-flop T atau flip-flop *toggle* adalah flip-flop JK yang kedua masukannya (J dan K) digabungkan menjadi satu sehingga hanya ada satu jalan masuk. Karakteristik dari flipflop ini adalah kondisi keluaran akan selalu *toggle* atau *berlawanan* dengan kondisi sebelumnya apabila diberikan masukan logika 1. Sementara itu kondisi keluaran akan tetap atau sama dengan kondisi keluaran sebelumnya bila diberi masukan logika 0.

Input	Output
T	Q_{n+1}
0	Q_n
1	\bar{Q}_n



3.2.5 REGISTER

Register adalah rangkaian yang tersusun dari satu atau beberapa flipflop yang digabungkan menjadi satu untuk menyimpan sebuah data. Untuk menyimpan data pada register, dapat dilakukan dengan dua cara :

- ❑ Disimpan secara sejajar (*Parallel In*) :

Pada cara ini semua bagian register atau masing-masing flipflop diisi (dipicu) pada saat yang bersamaan.

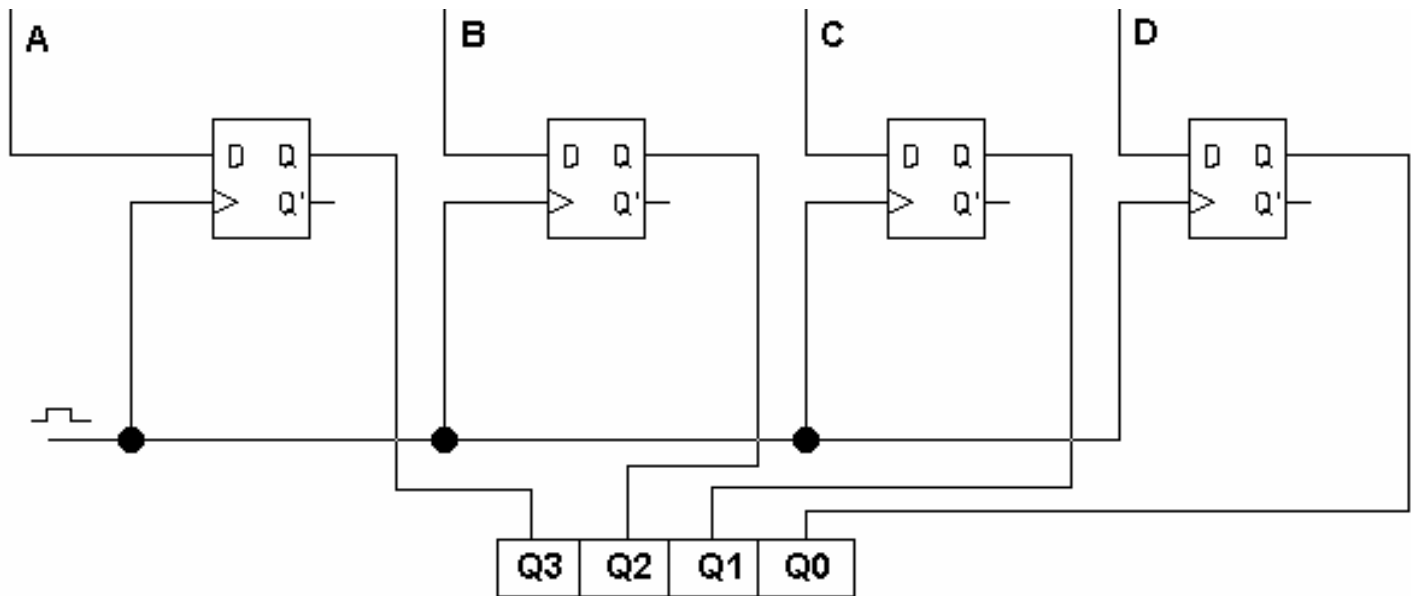
- ❑ Disimpan secara seri (*Serial In*) :

Pada cara ini, data dimasukkan bit demi bit mulai dari flipflop yang paling ujung , sampai semuanya terisi. Bila data digeser dari kanan ke kiri disebut “Register geser kiri” (*Shift Left Register*), sebaliknya bila data digeser dari kiri ke kanan disebut “Register geser kanan” (*Shift Right Register*).

Untuk mengeluarkan data juga dapat dilakukan dengan dua cara :

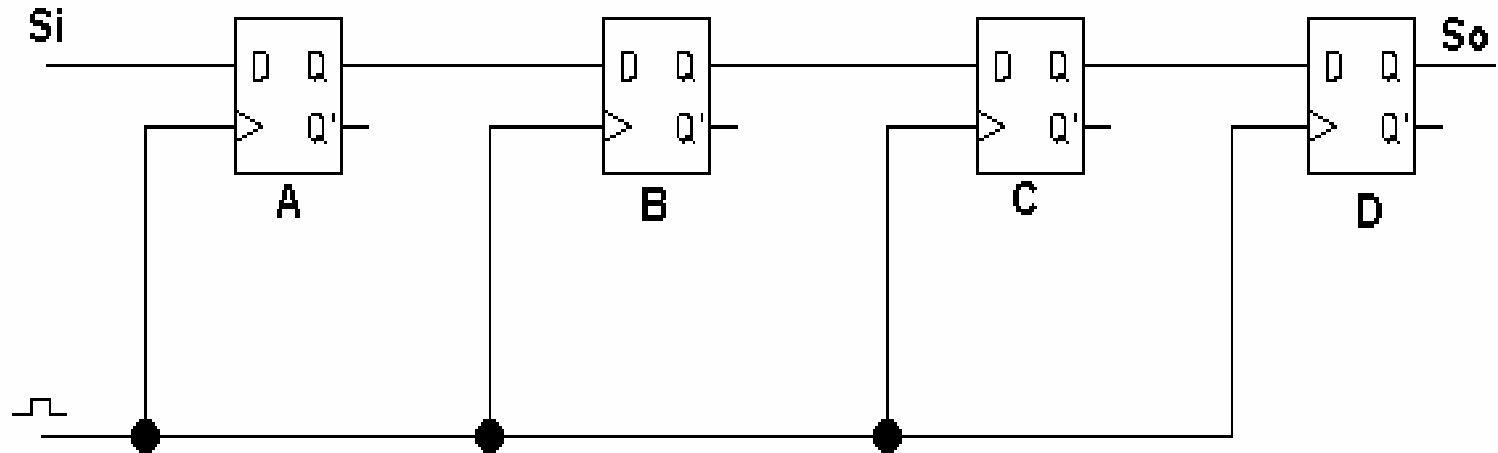
- ❑ Dikeluarkan secara sejajar (*Parallel Out*)
- ❑ Dikeluarkan secara seri (*Serial Out*)

❖ *Parallel In - Parallel Out (PIPO)*



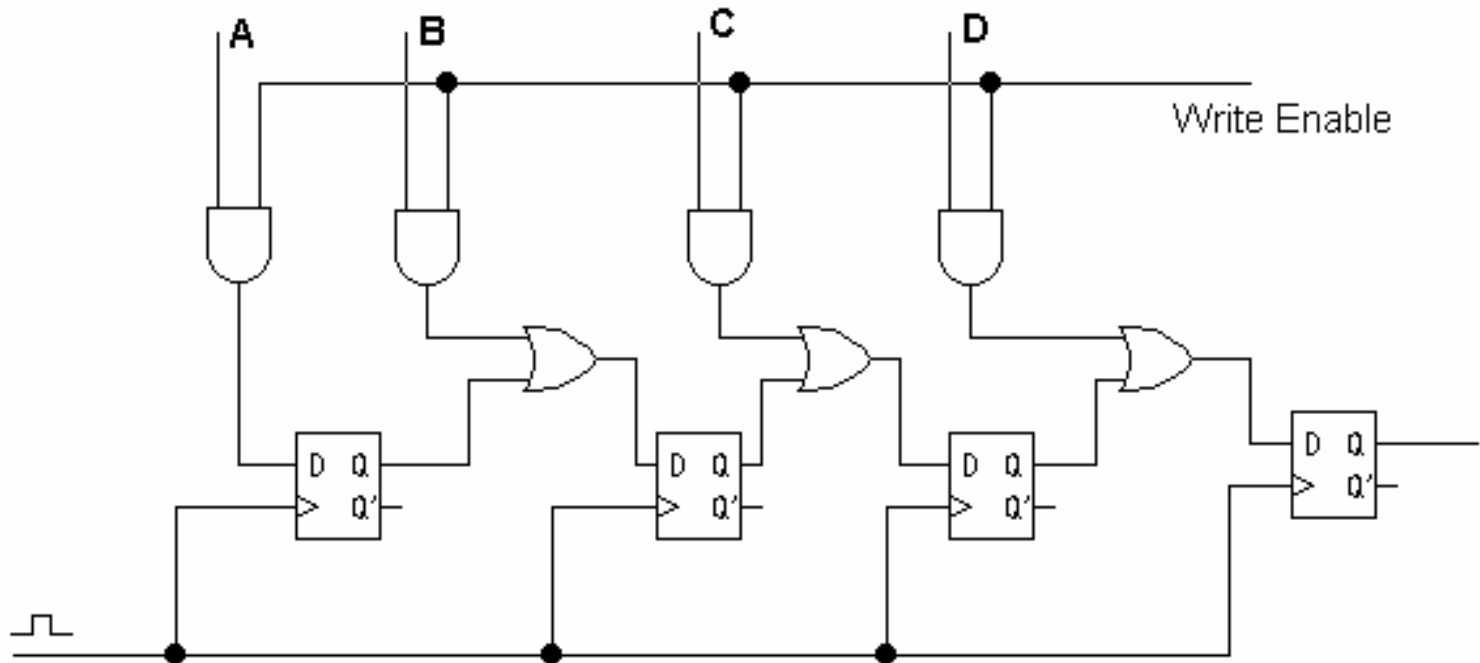
A, B, C, dan D adalah sinyal masukan. Saat *clock* (pemicu) diaktifkan (Logika 1), maka data yang ada akan dikeluarkan secara bersamaan ke Q3, Q2, Q1, dan Q0. Saat *clock* kembali tidak dipicu (Logika 0), maka apapun masukannya, keluaran Q akan tetap.

❖ *Serial In - Serial Out (SISO)*



Saat sinyal *clock* diberikan pertama kali, data dari Si masuk ke flipflop A, pada saat *clock* kedua, data dari flipflop A masuk ke flipflop B, demikian seterusnya, sampai keluar ke So. Jadi pada register SISO untuk membaca data pertama kali dibutuhkan jumlah *clock* yang sama banyak dengan jumlah flipflop yang ada pada register (dalam hal ini adalah empat).

❖ *Parallel In – Serial Out (PISO)*



❖ *Serial In – Parallel Out (SIPO)*

