

Arsitektur Komputer

Rangkaian Logika Kombinasiional & Sekuensial

Rangkaian Logika

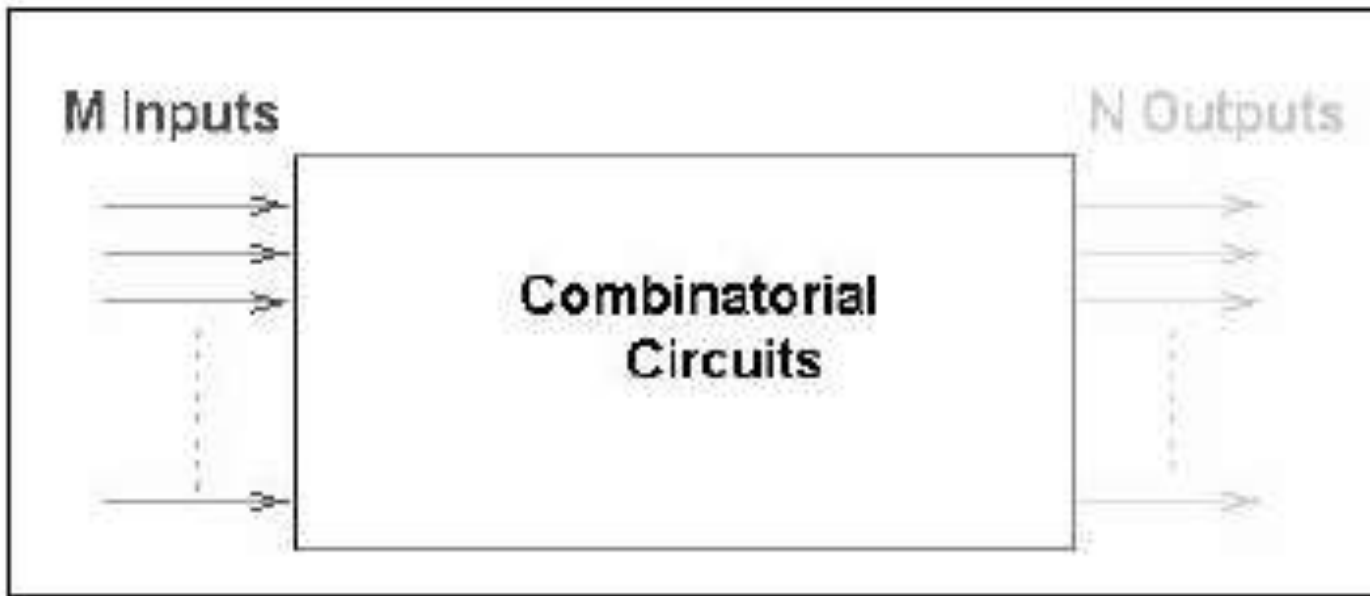
⌘ Rangkaian Logika secara garis besar dibagi menjadi dua, yaitu :

☑ **Rangkaian Kombinasional** adalah rangkaian yang kondisi keluarannya (output) dipengaruhi oleh kondisi masukan (input) pada saat itu saja

☑ **Rangkaian Sekuensial** adalah rangkaian yang kondisi keluarannya dipengaruhi oleh kondisi masukan dan keadaan keluaran sebelumnya atau dapat juga dikatakan rangkaian yang bekerja berdasarkan urutan waktu. Ciri rangkaian logika sekuensial yang utama adalah adanya jalur umpan balik (*feedback*) di dalam rangkaianannya.

Rangkaian Kombinasional

- ⌘ Rangkaian yang termasuk rangkaian logika kombinasional yaitu Decoder, Encoder, Multiplexer, Demultiplexer, Adder, Subtractor



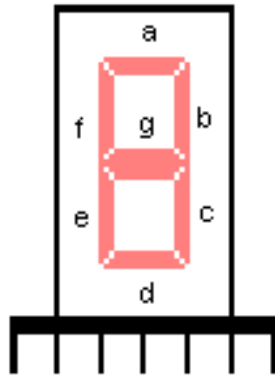
Rangkaian Kombinasional (lanj.)

⌘ Decoder

- ☑ berfungsi merubah kode biner menjadi sinyal diskrit
- ☑ m = kombinasi keluaran, n = jumlah bit masukan dengan hubungan $m = 2^n$
- ☑ Contoh decoder yaitu decoder dot matrix dan decoder 7 segmen

Rangkaian Kombinasional (lanj.)

⌘ Decoder 7 segmen



INPUT				OUTPUT						
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	1	1	1	0	1	1	1
1	0	1	1	0	0	1	1	1	1	1
1	1	0	0	1	0	0	1	1	1	0
1	1	0	1	0	1	1	1	1	0	1
1	1	1	0	1	0	0	1	1	1	1
1	1	1	1	1	0	0	0	1	1	1

Rangkaian Kombinasional (lanj.)

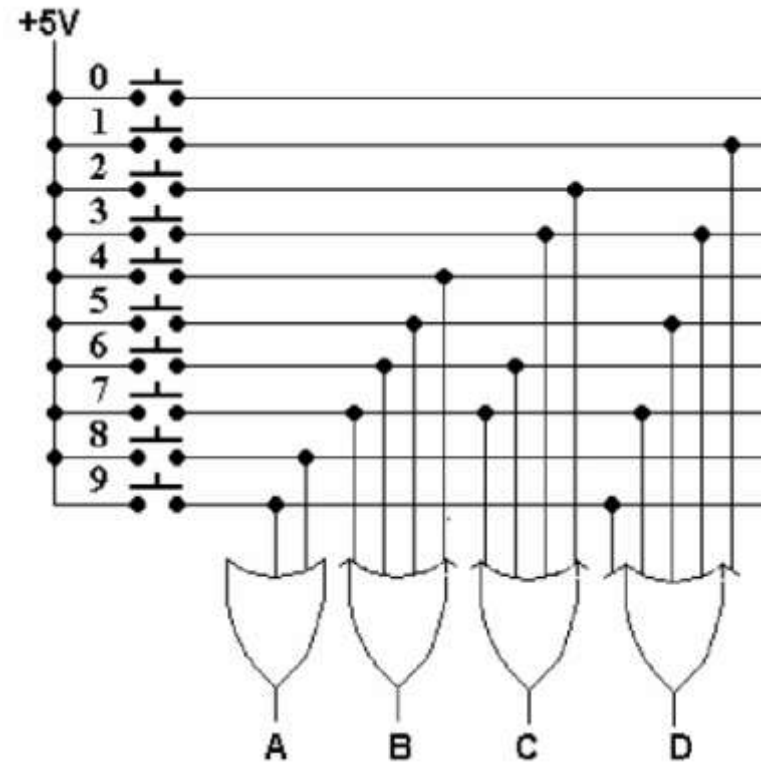
⌘ Encoder

- ☑ Kebalikan dari decoder, merubah sinyal diskrit menjadi kode biner
- ☑ Contohnya adalah decimal to BCD encoder, merubah kode desimal menjadi bentuk binernya (seperti pada kalkulator)

Rangkaian Kombinasional (lanj.)

⌘ Encoder Decimal to BCD

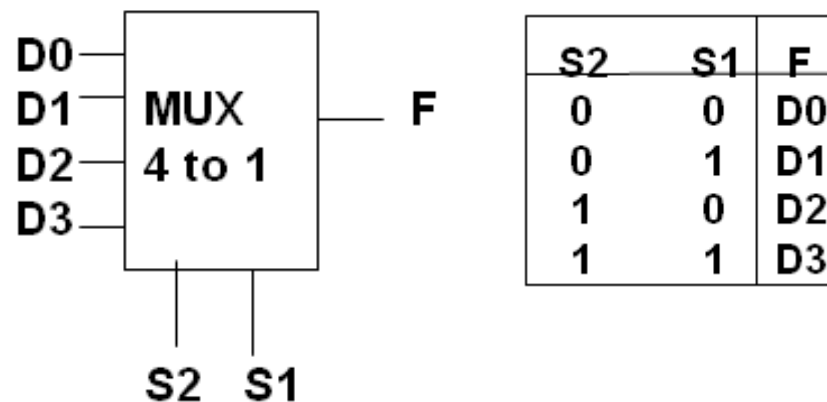
INPUT										OUTPUT			
0	1	2	3	4	5	6	7	8	9	A	B	C	D
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0	1	1	1
0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	1



Rangkaian Kombinasional (lanj.)

⌘ Multiplexer

- ☑️ Sirkuit logika yang memilih informasi boner dari salah satu 2^n baris input dan memindahkannya ke baris output tunggal
- ☑️ Memiliki n baris seleksi (pengendali)



Rangkaian Kombinasional (lanj.)

⌘ Demultiplexer (data distributor)

- ☑ Mengambil sebuah input kemudian menyebarkannya ke beberapa output

Rangkaian Sekuensial

⌘ Selain input data, rangkaian sekuensial juga menerima *trigger* (pemicu) berupa clock. Output rangkaian pada satu kondisi merupakan *feedback* dari clock

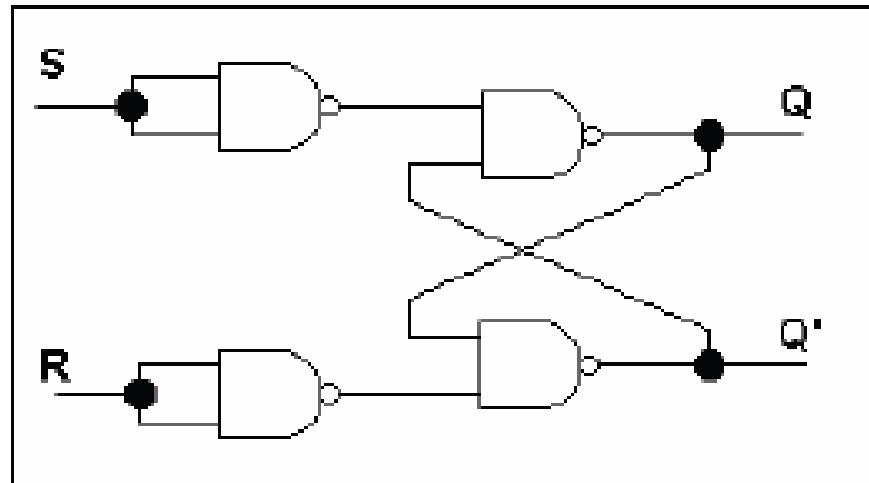
☐ Contoh: Rangkaian flip-flop

Rangkaian Flip Flop (FF)

- ⌘ Flipflop adalah rangkaian utama dalam logika sequensial. Counter, Register, Memory, serta rangkaian sequensial lainnya disusun dengan menggunakan flipflop sebagai komponen utama.
- ⌘ Flipflop adalah rangkaian yang mempunyai fungsi pengingat (*memory*). Artinya rangkaian ini mampu melakukan penyimpanan data sesuai dengan kombinasi masukan yang diberikan kepadanya.

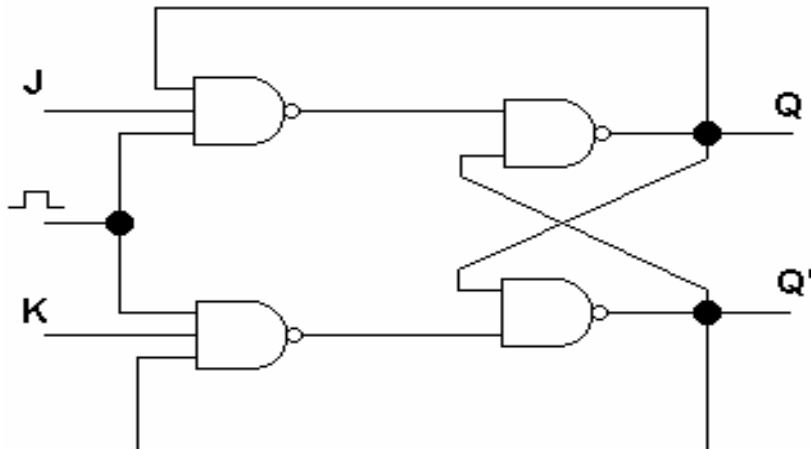
Flip Flop RS

- ⌘ Flipflop ini terdiri dari dua masukan, yaitu S (set) dan R (reset). Serta dua keluarannya yaitu Q dan Q' .
- ⌘ Kondisi Set adalah kondisi ketika Q berlogika 1. Sedangkan kondisi Reset adalah kondisi ketika Q berlogika 0.



Flip Flop JK

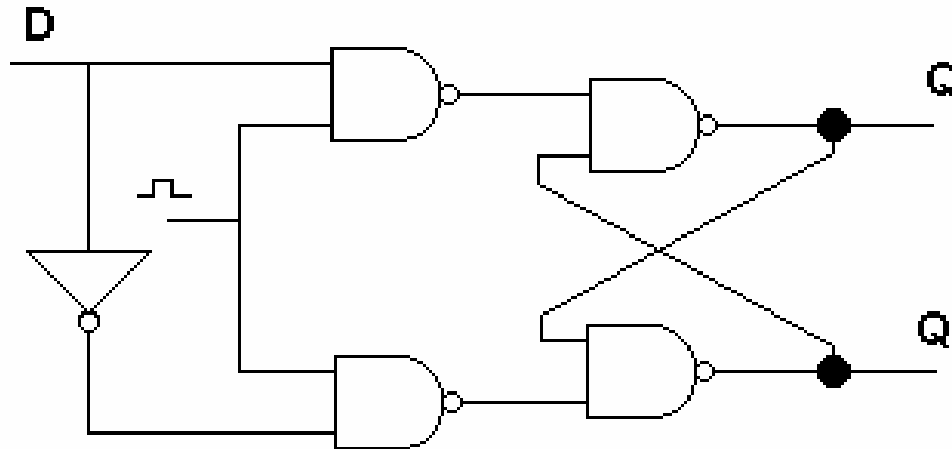
- ⌘ Flipflop JK merupakan penyempurnaan dari flipflop RS terutama untuk mengatasi kondisi bistabil
- ⌘ Pada kondisi masukan $J = 1$ dan $K = 1$ akan membuat kondisi keluaran berlawanan dengan kondisi keluaran sebelumnya.



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

Flip Flop D

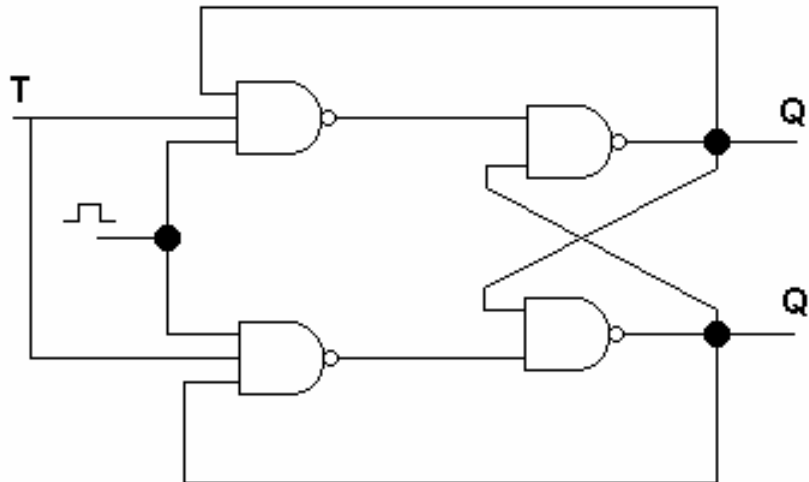
- ⌘ Flipflop D merupakan Flipflop RS yang dimodifikasi sehingga hanya memiliki satu masukan yang selalu berlawanan
- ⌘ Flip flop data merupakan rangkaian dasar dari register



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

Flip Flop T

- ⌘ Flipflop T atau flipflop toggle adalah flipflop JK yang kedua masukannya (J dan K) digabungkan menjadi satu sehingga hanya ada satu input.
- ⌘ Karakteristik dari flipflop ini adalah kondisi keluaran akan selalu *toggle* atau berlawanan dengan kondisi sebelumnya apabila diberikan input logika 1.



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

Register

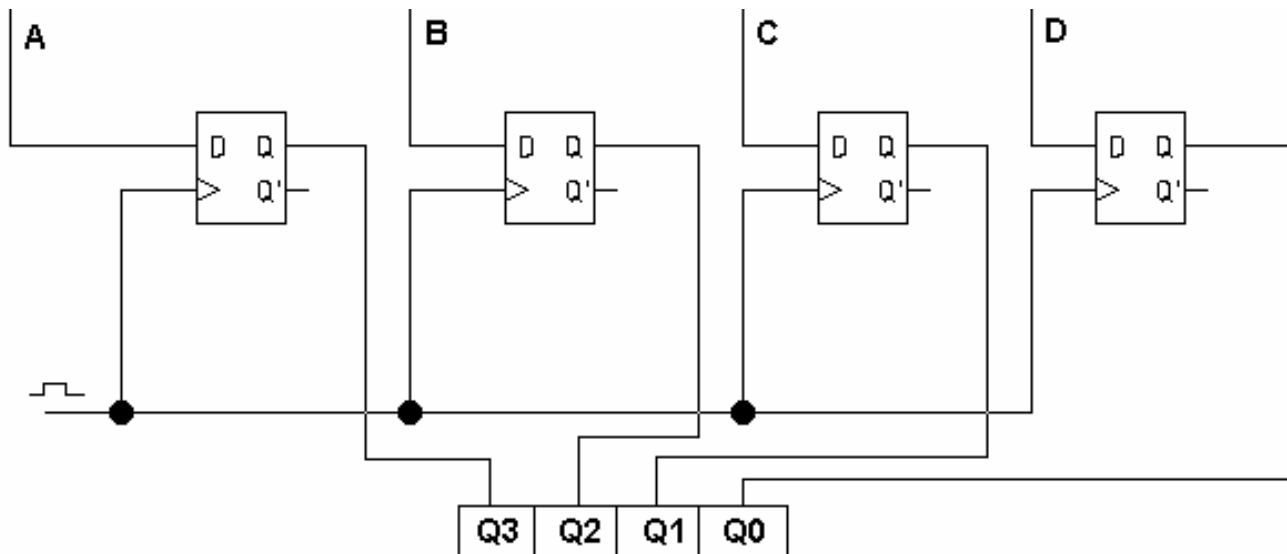
- ⌘ Register adalah rangkaian logika yang digunakan untuk menyimpan data.
- ⌘ Rangkaian register tersusun dari satu atau beberapa flip flop yang digabungkan menjadi satu.
- ⌘ Flipflop disebut juga sebagai register 1 bit. Jadi untuk menyimpan 4 bit data, register harus terdiri dari 4 buah flipflop.

Register (lanj.)

- ⌘ 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 (dapat dari kiri atau dari kanan), dan digeser (*shift*) sampai semuanya terisi.
- ⌘ Untuk mengeluarkan data juga dapat dilakukan dengan dua cara :
 - ☑ Dikeluarkan secara sejajar (Parallel Out)
 - ☑ Dikeluarkan secara seri (Serial Out)

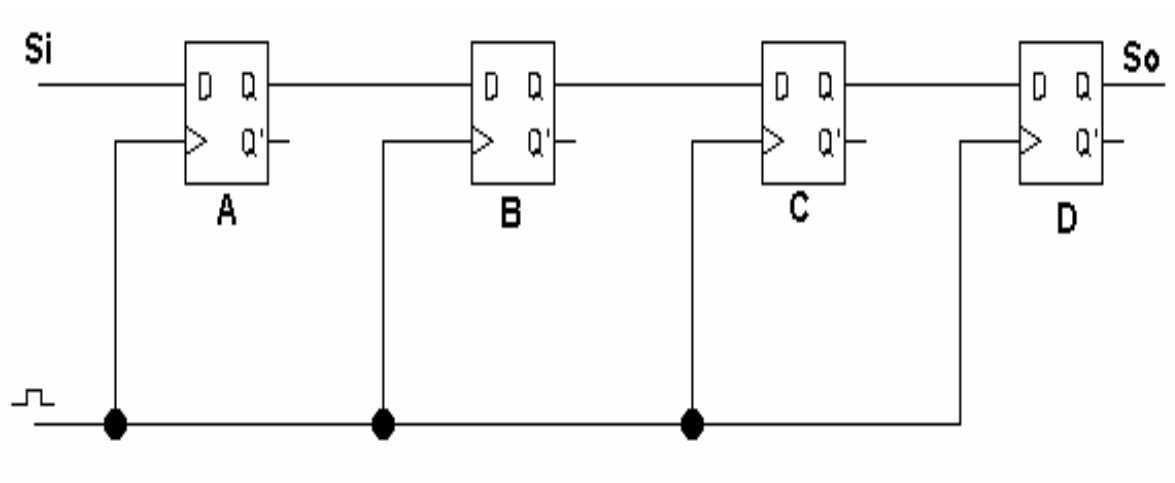
Register Parallel In-Parallel Output

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

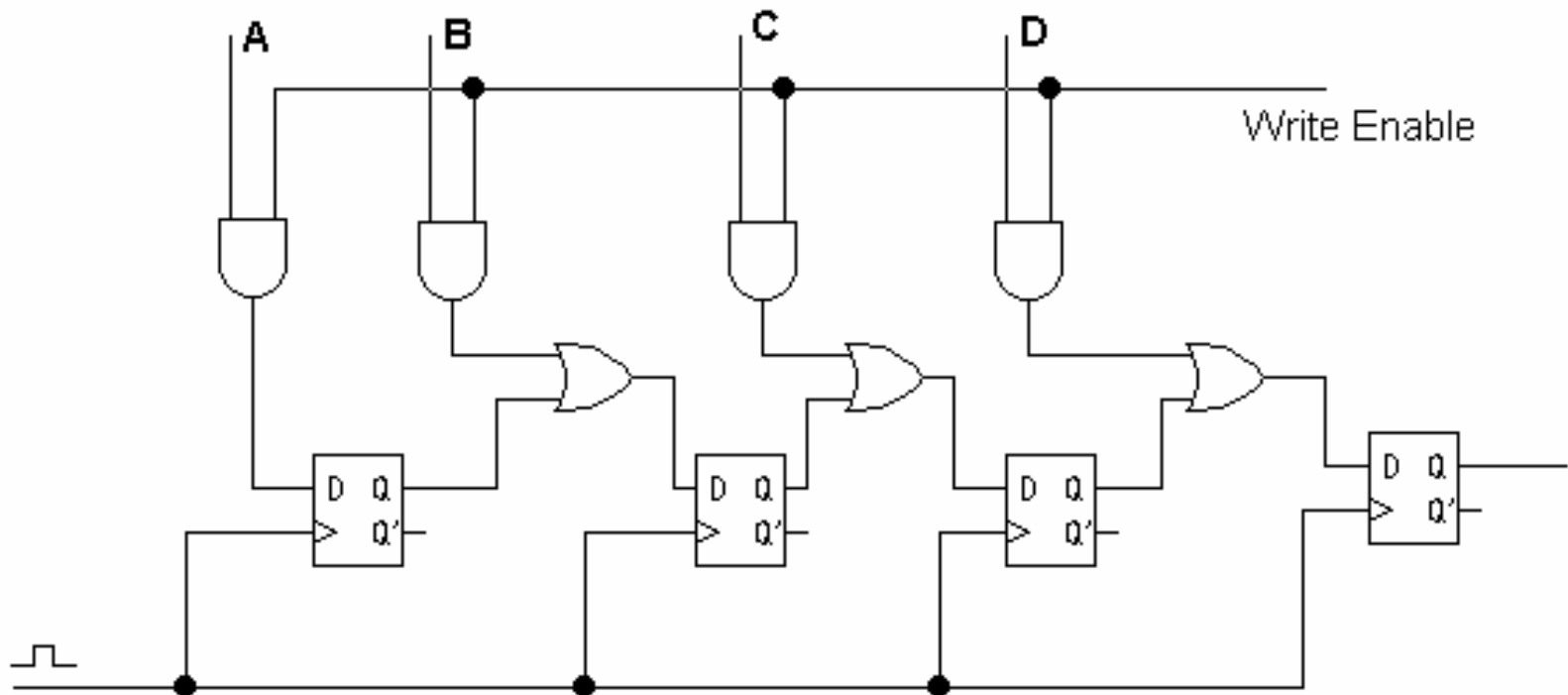


Register Serial In-Serial Output

- ⌘ 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 Output (PI-SO)



Serial In-Parallel Output (SI-PO)

