

BAB 7 REGISTER

7.1. Register

Sebuah flip flop dapat digunakan untuk menyimpan data 1 bit, sehingga jika ada sederetan dari n buah FF, maka dapat dipergunakan untuk menyimpan data sebanyak n buah bit pula. Kumpulan dari FF yang seperti ini disebut : REGISTER

Register adalah alat untuk menyimpan data yang berupa satu atau beberapa FF yang digabungkan menjadi satu. Register yang paling sederhana hanya terdiri dari satu FF saja, yang berarti hanya dapat menyimpan data yang terdiri dari satu bit bilangan biner saja yaitu "0" atau "1".

Kegunaan Register, selain untuk menyimpan data dalam waktu yang terbatas (*Temporary Storage*), juga berguna untuk :

- Sistem Pencacah (*Counter*)
- Operasi Bilangan
- Sistem Penundaan Sinyal (*Time Delay*)
- Sistem Pembangkit Sandi (*Code Generator*)

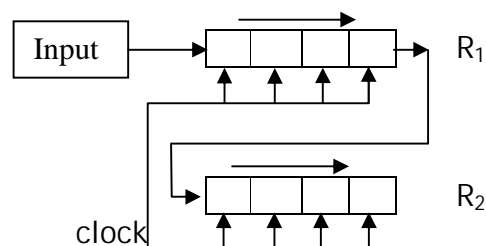
Keluar dan masuknya data ke dalam register dapat dilakukan dengan 2 cara:

1. Serial

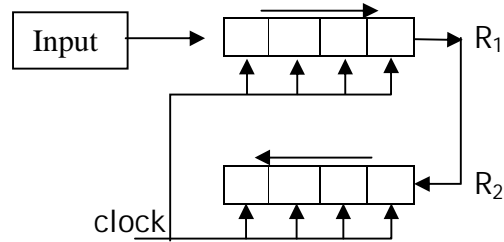
Data dimasukkan/dikeluarkan bit demi bit bergantian lewat satu saluran (biasanya LSB dulu = bit paling kanan). Pelaksanaannya diatur oleh sinyal "Clock" yang bekerja bersama untuk kedua register tersebut.

Sinyal clock dihubungkan menjadi satu untuk kedua register tersebut.

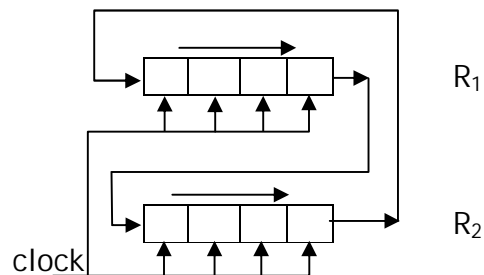
- Data yang berasal dari register ke kanan dipindahkan ke register yang bergeser ke kanan juga.



- Data yang berasal dari register yang bergeser ke kanan dipindahkan ke register yang bergeser ke kiri.

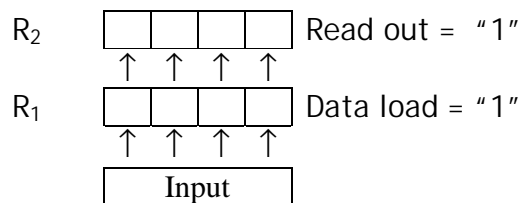


Serial output "R1" dihubungkan dengan serial input "R2" dan serial output "R2" dihubungkan dengan serial input "R1". Dalam kondisi semacam ini data akan bergeser secara berputar dari "R1" ke "R2" dan kembali lagi ke "R1" dan seterusnya.



2. Paralel

Data dimasukkan/dikeluarkan ke register yang lain pada saat yang sama (serempak), sehingga N buah bit akan dimasukkan/dikeluarkan secara bersamaan.



Data yang disimpan di register no 1 (R1) dipindah ke register no 2 (R2) dengan cara jajar (serempak)

Dari cara masuk dan keluarnya data maka ada 4 macam register :

- Serial in — Paralel Out / Serial to Paralel Register
- Serial in — Serial Out
- Paralel in — Serial out / Paralel to Serial Register
- Paralel in — Paralel Out

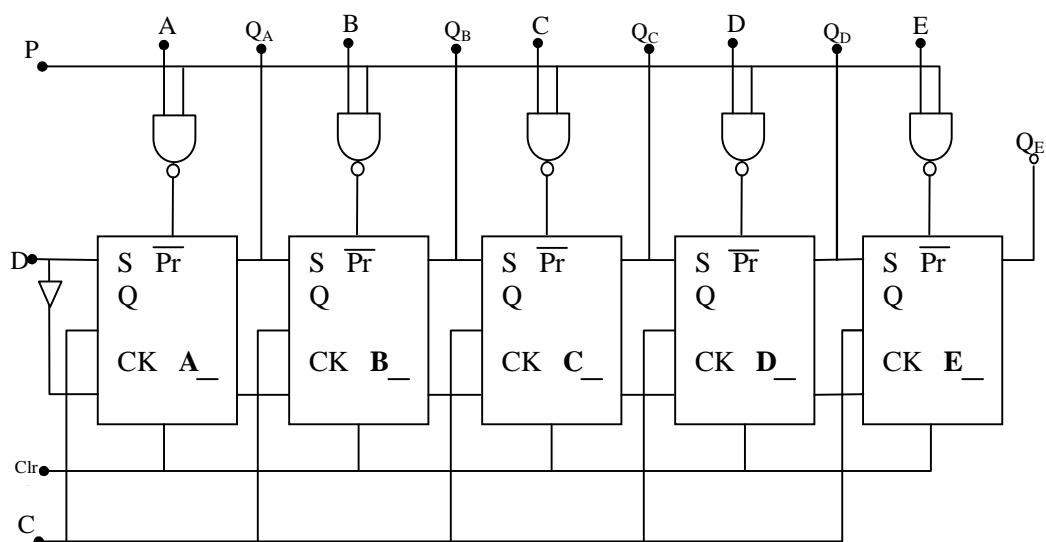
Ada dua cara untuk mengisi Register (*data loaded*) :

- Dimasukkan secara Jajar (Paralel In)
- Dimasukkan secara Deret (Serial In), dengan cara :
 1. Data digeser dari kanan ke kiri disebut : "Register bergeser ke kiri" (*Shift - Left Register*)
 2. Data di geser dari kiri ke kanan disebut : "Register bergeser kanan" (*Shift - Right Register*)
 3. Register yang dapat bergerak ke kiri dan ke kanan, disebut "*Reversible Shift Register*".

Untuk mengeluarkan data (*Read Data*), dapat dilakukan dengan cara :

- Dikeluarkan secara Jajar (*Paralel - Out*)
- Dikeluarkan secara Deret (*Serial - Out*)

Perhatikan gambar berikut ini untuk memahami cara kerja dari setiap macam register :



gambar (a)

- Gambar tersebut merupakan register yang terbuat dari 5 buah SRFF yang dilengkapi dengan input preset dan clear, dengan demikian dapat disimpan data sebanyak 5 buah bit.
- Diketahui bahwa apabila harga S dan R saling berlawanan, maka SRFF akan bekerja sebagai DFF, yang berarti outputnya setelah pulsa klok akan sama dengan harga S-nya.
- Dari gambar dapat dilihat bahwa dengan adanya gerbang NOT sebelum masuk ke input R dari FFA dan dengan dihubungkannya R dari FF berikutnya dengan output Q' dari FF sebelumnya, dan S dengan output Q, maka setiap FF yang ada akan berharga "1" sebagaimana layak DFF.
- Oleh karena SRFF yang digunakan dilengkapi dengan preset dan clear, maka agar FF dapat bekerja sebagaimana mestinya, setiap kali perlu diatur $PE = 0$ dan $Cr = 1$, karena dengan demikian $Pr = Cr = 1$.
- Cara me-reset FF dapat dilakukan dengan mengubah sebentar clear ke 0 dan kemudian segera dikembalikan lagi ke 1. Pada saat clear ke 0 yang berarti $Cr = 0$ dan $Pr = 1$, maka $Q = 0$. Dan setelah Clock kembali ke 1 harga Q masih 0 (tetap). Kalau setelah direset kemudian harga PE diubah sebentar ke 1 dan kemudian dikembalikan ke 0, maka $Q_a Q_b Q_c Q_d Q_e = ABCDE$, dengan demikian seolah-olah data ditransfer dari ABCDE ke $Q_a Q_b Q_c Q_d Q_e$.

Contoh :

- Jika $A = 1$, maka pada saat $PE = 1$, $Pr = 0$, sehingga $Q_a = 1$, sedang kalau pada saat $PE = 1$, $A = 0$, maka $Pr = 1$, sehingga $Q_a = 0$.
- Jadi untuk memasukkan data 5 bit secara paralel dapat dikerjakan lewat input A,B,C,D,E dengan merubah sebentar PE ke 1 (sebelum memasukkan data secara paralel, register harus diset dahulu).
- Gambar (a) dapat digunakan untuk memasukkan data baik secara serial maupun paralel.

7.2. Menyimpan Data

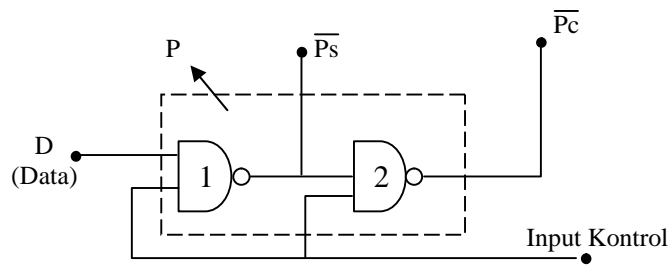
7.2.1. Menyimpan Data Secara Paralel (Paralel - In)

Prinsip Dasar Paralel-In adalah sebagai berikut :

1	1	0	1
\hat{a}	\hat{a}	\hat{a}	\hat{a}

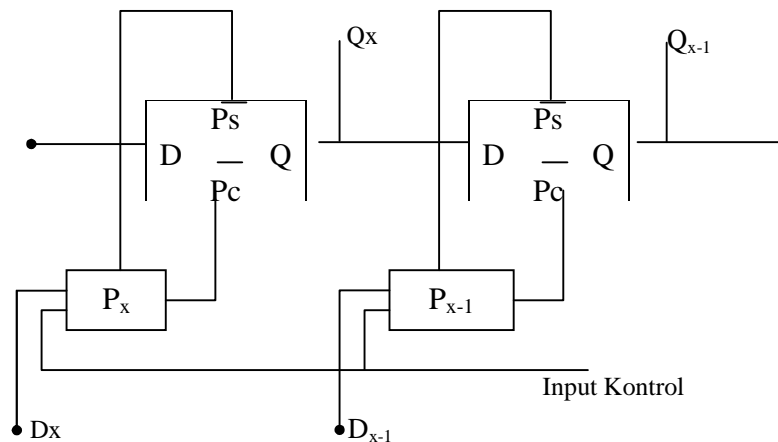
Register atau masing-masing FF akan dimuati dalam waktu yang bersamaan (serempak) yang diatur oleh sinyal pengontrol yang disebut "Input Kontrol". Bila input kontrol = "0", maka semua output FF = Q_n tetapi bila input kontrol = 1, maka semua output FF = Q^{n+1} . Akan dimuati data secara serempak (output FF = Q^{n+1}) Register terisi secara serentak setelah input control diberi perintah "Data loaded"

Rangkaian input kontrol adalah sebagai berikut :



Bila input kontrol = 0, maka $\bar{P}_s = 1$ dan $\bar{P}_c = 1$ sehingga input ("D") tidak mempengaruhi output \bar{P}_s dan \bar{P}_c maka Q_x tak berubah.

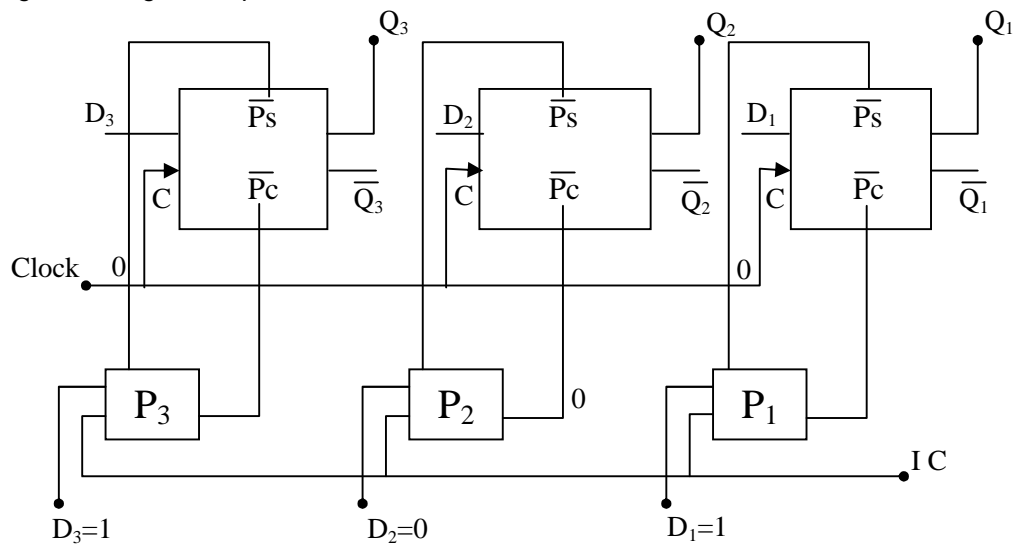
Rangkaian Logikanya sebagai berikut :



Bila input kontrol = 1 maka $\overline{P_s}$ dan $\overline{P_c} = f(D)$,
jadi jika $D = 0 \Rightarrow \overline{P_s} = 1$ dan $\overline{P_c} = 0$ sehingga $Q_x = 0 \Rightarrow Q_x = f(D)$.

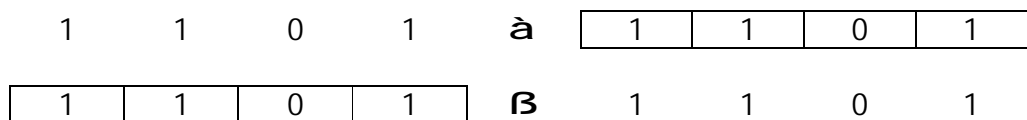
$D = 1 \Rightarrow \overline{P_s} = 0$ dan $\overline{P_c} = 1$ sehingga $Q_x = 1 \Rightarrow Q_x = f(D)$

Diagram rangkaian paralel - In :



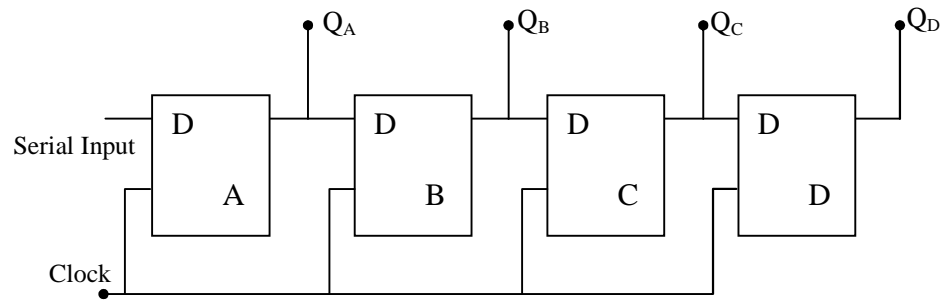
7.2.2. Menyimpan Data Secara Deret (Serial In)

Prinsip dasar Serial - In adalah sebagai berikut :

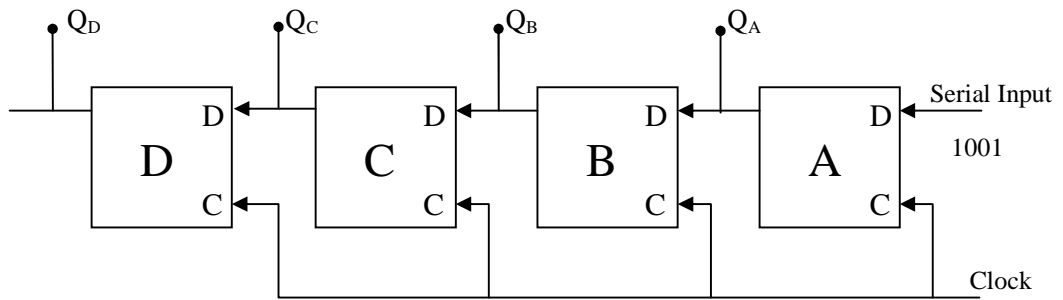


- Menyimpan data secara deret dilakukan dengan memasukkan data bit demi bit mulai FF yang paling ujung, dan digeser satu per satu sampai FF berisi.
- Pergeseran data diatur oleh sinyal klock.

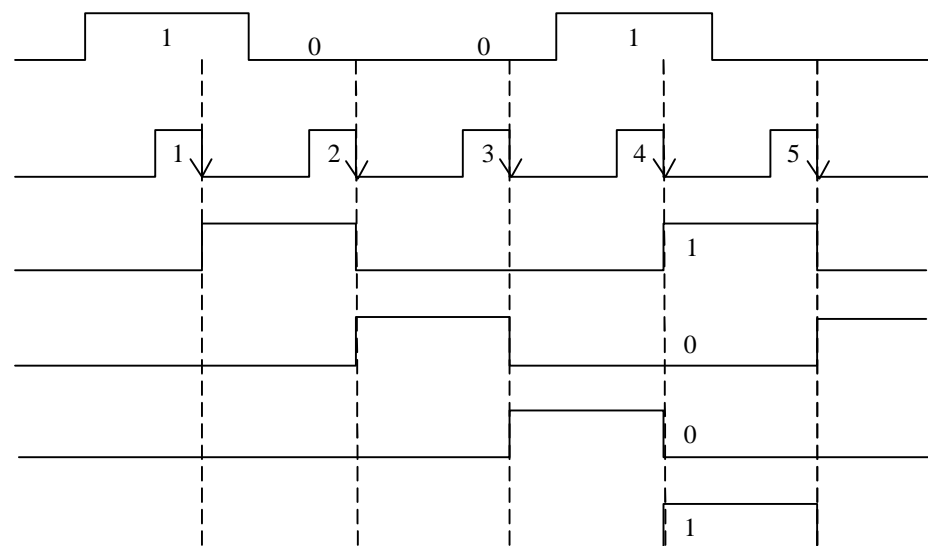
Rangkaian logikanya :



Rangkaian dan Timing Diagramnya (Serial - In)



D-FF berasal dari JK-FF



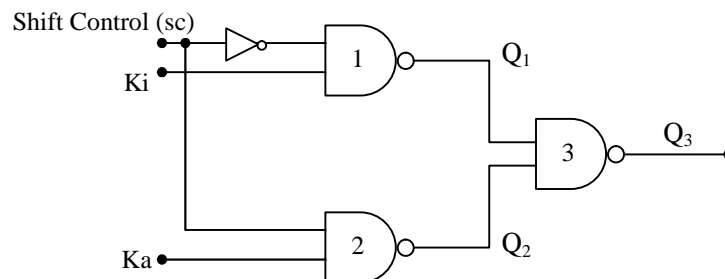
CLOCK	Pergeseran ke Kiri				INPUT DATA
	D	C	B	A	
1	0	0	0	1	1
2	0	0	1	0	0
3	0	1	0	0	0
4	1	0	0	1	1

7.3. Register Geser Kanan/Kiri (Right / Left Shift Register)

Register - register sebelumnya, bit demi bitnya akan bergeser ke kanan bila ada pulsa clock yang masuk maka register demikian disebut Register Geser Kanan (*Right Shift Register*).

Ada register yang bukan saja hanya bergeser ke kanan tetapi juga ke kiri. Register yang bekerja demikian disebut juga dengan *Reversible Shift Register*.

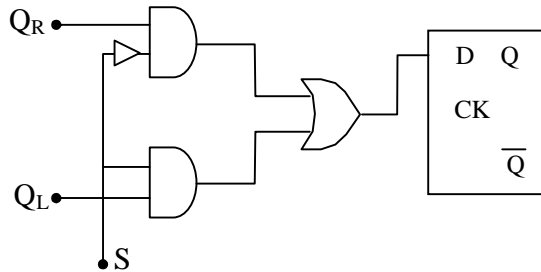
Untuk register geser kiri/kanan dibutuhkan suatu rangkaian pengatur tambahan yang dapat mengatur pergeseran bit demi bit baik ke kanan maupun ke kiri. Rangkaian pengatur ini dihubungkan dengan output FF di sebelah kirinya (Q1) kalau harus bergeser ke kanan; atau dengan output FF sebelah kanan (Q3) jika akan bergeser ke kiri.



Sebagai pengontrol pergeseran yang dikehendaki digunakan "Shift Control Circuit "

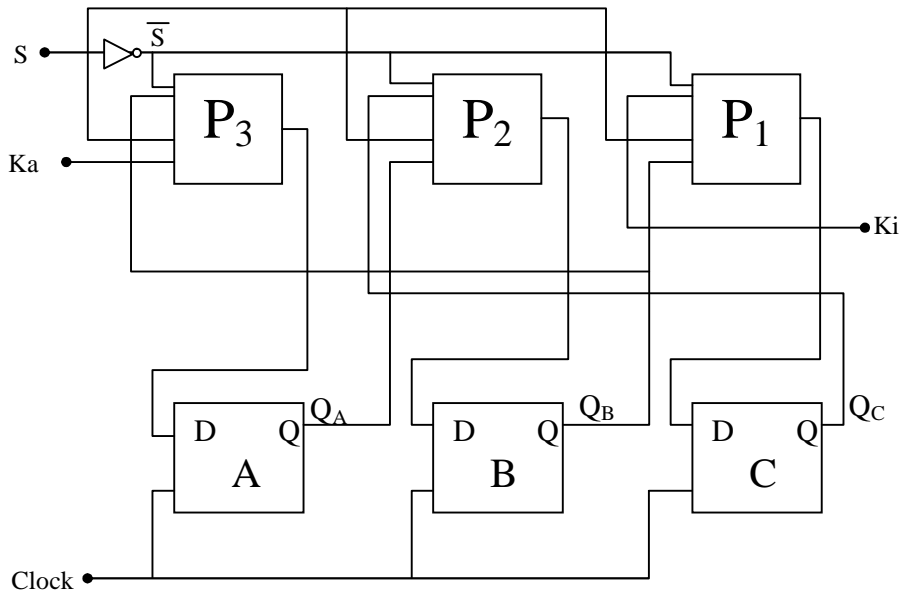
- Bila "SC = 1" → Q1 = 1 → Q1 # f (Ki) → Q3 = f (Ka)
- Bila "SC = 0" → Q2 = 1 → Q2 # f (Ka) → Q3 = f (Ki)

Rangkaian dengan DFF sebagai berikut :

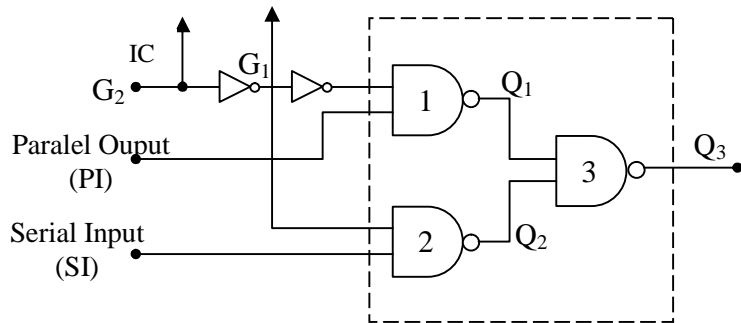


$D = S' \cdot Q_R + S \cdot Q_L$
 Kalau $S=1$, maka $D= Q_L$,
 jadi bit demi bit akan
 menggeser ke kanan.
 Kalau $S=0$, maka $D= Q_R$,
 jadi bit demi bit akan
 bergeser ke kiri.

Rangkaian Logika *Reversible Shift Register* adalah sebagai berikut :

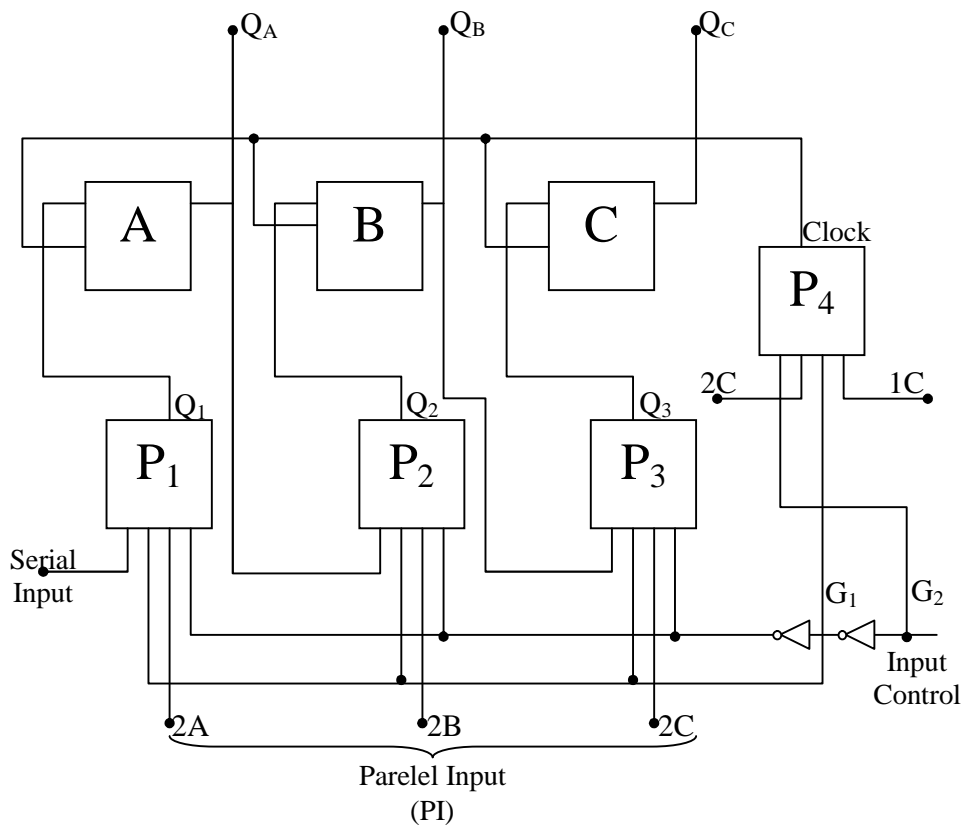


Register yang bekerja secara deret maupun secara jajar
 Membuka / menutup Clock



- Register/rangkaian ini merupakan pengontrol Register yang bekerja Parallel - In atau Serial In.
- Bila "IC = 0" \Rightarrow G2 = 0 \Rightarrow G1 = 1 \Rightarrow Q1 = 1 \Rightarrow Q1 # f (PI)
Pada saat G1 = 1 \Rightarrow Q2 = f (SI) ; Q3 = f (SI) ; dan G1 - pun akan membuka sinyal clock.
- Bila "IC = 1" \Rightarrow G2 = 1 \Rightarrow G1 = 0 \Rightarrow Q2 # f (SI) ; dan Q1 = f (PI),
Q3 = f (PI).
Pada saat "G2 = 1", berfungsi sebagai pembuka sinyal clock.

Rangkaian Logika Register yang dapat bekerja paralel maupun serial adalah sebagai berikut :



- Untuk Blok "P4" berfungsi untuk mengatur jalannya sinyal clock, dan prinsip kerjanya sama.
- Bila "IC = 0" \Rightarrow G2 = 0, maka output Q3 (PI) sampai dengan Q3 (P3) adalah f (SI), dan data yang diteruskan dari "Serial Input" yang dikontrol oleh sinyal Clock IC.

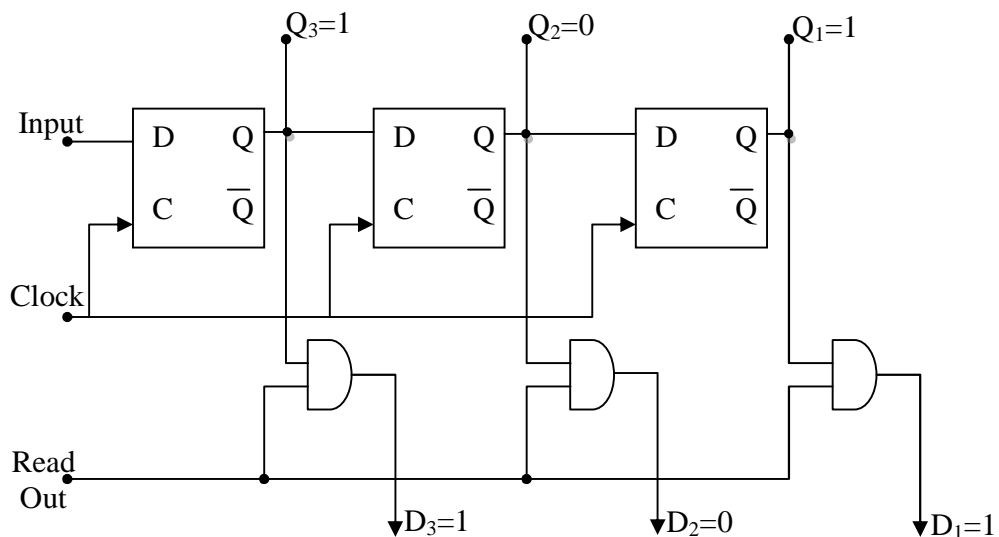
- Bila "IC = 1" dan $G_2 = 1$, maka output Q_3 (PI) sampai dengan Q_3 (P3) adalah f (PI), dan data yang diteruskan dari "Paralel Input" yang dikontrol oleh sinyal clock - 2C.

7.4. Mengeluarkan Data

7.4.1. Serial - Out

Mengeluarkan Data secara seri (serial - out). Pelaksanaannya sama seperti mengisi data secara seri (serial - in)

7.4.2. Parallel - Out



$$D = Q, \quad \text{bila "Read Out" = 1}$$

7.5. Kombinasi Serial Dan Paralel

7.5.1. Serial In - Paralel Out

Untuk memasukkan atau menulis data 5 bit secara serial, langkah pertama register harus direset dahulu sehingga $Q_a Q_b Q_c Q_d Q_e = 00000$. Misalkan data yang akan dimasukkan 10011 (dimulai bit yang paling kanan). Kemudian sinkron dengan pulsa klock pertama, kedua, ... dan sampai kelima, masukkan lewat input serial inisial A berturut-turut harga 1, 1, 0, 0, dan 1.

Dengan demikian :

Pulsa klok ke satu	, Qa Qb Qc Qd Qe = 10000
Pulsa klok ke dua	, Qa Qb Qc Qd Qe = 11000
Pulsa klok ke tiga	, Qa Qb Qc Qd Qe = 01100
Pulsa klok ke empat	, Qa Qb Qc Qd Qe = 00110
Pulsa klok ke lima	, Qa Qb Qc Qd Qe = 10011

Jadi setelah pulsa klok kelima data 5 bit tersebut dapat dibaca dan dikeluarkan lewat 5 buah output dari masing-masing FF secara paralel.

Data di atas dimasukkan secara serial dan dikeluarkan/dibaca secara paralel, maka register tersebut disebut Register Serial In - Paralel Out atau Serial Paralel Converter.

7.5.2. Serial In - Serial Out

Data yang dimasukkan secara serial seperti pada serial in - paralel out, kemudian Qe dijadikan sebagai tempat keluarnya data, dan sesudah itu dimasukkan pulsa clock sebanyak 5 kali, maka setiap data akan bergeser ke kanan 5 kali.

Dengan demikian :

Pada klok ke satu	, Qe = 1
Pada klok ke dua	, Qe = 1
Pada klok ke tiga	, Qe = 0
Pada klok ke empat	, Qe = 0
Pada klok ke lima	, Qe = 1

Data 5 bit akan keluar lewat Qe secara berurutan mulai dari bit LSB (Bit Paling kanan) sinkron dengan masuknya setiap pulsa clock.

7.5.3. Paralel In - Paralel Out

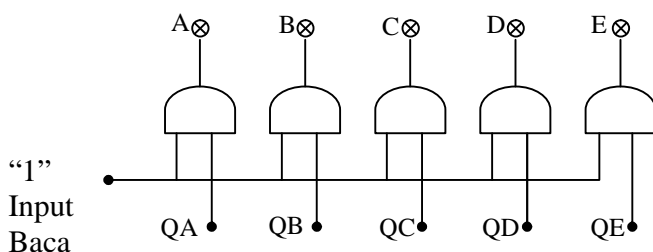
Data 5 bit dapat dimasukkan secara bersamaan (serempak) lewat input A, B, C, D, dan E (E sebagai LSB / bit paling kanan).

Andaikan harga ABCDE = 10011, maka setelah register direset terlebih dahulu dan kemudian PE diubah sebentar ke 1 maka data ditransfer ke Qa Qb Qc Qd Qe, sehingga harga Qa Qb Qc Qd Qe = 10011.

Data yang dimasukkan adalah secara paralel sedang pengeluarannya dilakukan secara paralel pula maka register ini disebut Paralel in - Paralel Out.

Untuk mengeluarkan data secara paralel, masing-masing output dari FF dihubungkan ke input sebuah gerbang AND, sedang input yang lain dari

gerbang AND dihubungkan ke input "baca". Input "Baca" harus diatur sama dengan 1. Gambarnya adalah sebagai berikut :



7.5.4. Paralel In - Serial Out

Cara kerjanya sama dengan "Paralel In - Paralel Out". Sebagai outputnya adalah Qe, jadi data outputnya dikeluarkan secara serial hanya dari output Qe. Untuk itu perlu dimasukkan pulsa clock sebanyak lima kali untuk mengeser data yang ada pada output-output sebelumnya.

Keluarnya output untuk setiap clock sama dengan serial in - paralel out, yaitu LSB akan keluar terlebih dahulu. Register ini dapat disebut sebagai register paralel in - serial out atau paralel to serial konverter.

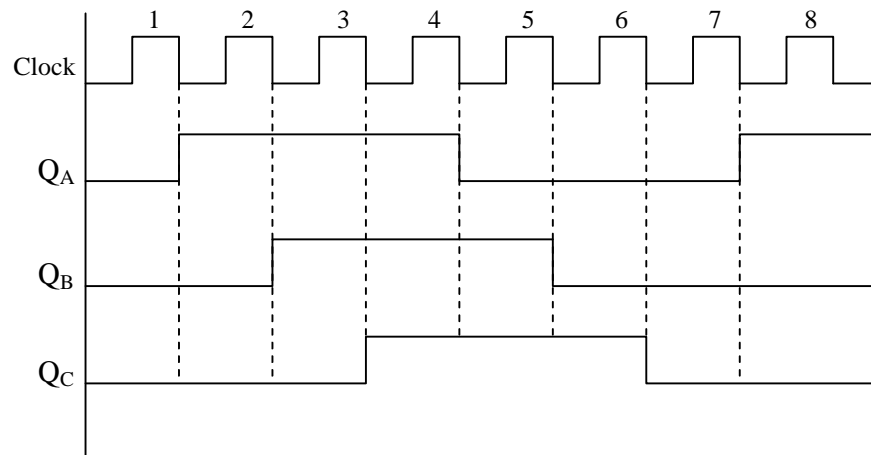
7.6. Feedback Shift Register (FSR)

Feedback Shift Register (FSR) sering juga disebut "*twisted ring counter*", yaitu semacam ring counter yang dihubungkan dengan bukan serial outputnya tetapi ke komplemen serial outputnya. "FSR" dengan "N-FF" akan menghasilkan "*2N - Sequence*" atau merupakan "pembagi - 2N"

Sinyal Clock	Kondisi FF		
	A	B	C
Kondisi awal	0	0	0
1	1	0	0
2	1	1	0
3	1	1	1
4	0	1	1
5	0	0	1
6	0	0	0
7	Kembali seperti Clock no.1, dst-nya		

Dari tabel di atas dapat dilihat bahwa digit "1" selalu bergerak/berjalan setiap pergantian hitungan, sehingga "FSR" untuk ini akan menghasilkan "Sandi Berjalan" atau "*Walking Coded*"

Diagram waktu dari "FSR Tiga Bit" tersebut di atas, dapat lihat pada gambar berikut ini:



LATIHAN

Rancanglah FSR - 4 bit dan FSR - 5 bit. Buat Tabel kebenaran dan Gambarkan diagram waktu dari masing-masing FSR tersebut.
