

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Pencacah.

Pencacah adalah rangkaian berurutan (sequensial) yang beroperasi melalui urutan kondisi tertentu berdasarkan aplikasi dari pulsa inputnya. Pulsa-pulsa input tersebut dinamakan pulsa penghitung. Pulsa penghitung ini bisa berupa clock atau pulsa-pulsa yang berasal dari external source dan mungkin terjadi pada interval waktu tertentu ataupun random.

Dalam suatu pencacah urutan kondisi tersebut mungkin mengikuti hitungan biner atau urutan kondisi yang lain. Pencacah tersebut digunakan untuk menghitung jumlah peristiwa dari suatu kejadian dan berguna untuk menimbulkan urutan waktu dalam mengontrol operasi-operasi dalam sistem digital.

Menurut cara kerja masukan ke dalam flip-flop maka pencacah dapat dibagi menjadi 2 macam yaitu pencacah Sinkron dan pencacah tak Sinkron. Cara penyusunan sejumlah flip-flop agar memenuhi urutan perubahan yang ditentukan tergantung pada macam dari pencacah dan juga flip-flop yang digunakan.

### 2.1. PENCACAH SINKRON

Pada pencacah sinkron saat perubahan dari keluaran flip-flop terjadi secara serempak sesaat setelah terjadi pulsa input dimasukkan pada input klok dari setiap flip-flop. Dengan cara ini hal yang perlu dikerjakan hanyalah mengatur harga input data dari flip-flop yang digunakan agar perubahan outputnya sesuai dengan tabel pencacahan.

Untuk J-K flip-flop tabel kebenarannya dapat ditulis sebagai berikut :

J	K	$Q_n$	$Q_{n+1}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Tabel 2.1 : Tabel kebenaran dari flip-flop J-K

Sedang pengaturan dari harga J dan K untuk setiap macam perubahan terlihat pada tabel 2.2. Dari tabel (2.2) nampak bahwa agar keluaran flip-flop dari 0  $\longrightarrow$  1, maka harga J harus = 0, sedang harga K boleh 0 atau

1. Hal ini dituliskan dalam tabel (2.2) baris 2 (X = boleh 0 atau 1). Tabel (2.2) disebut tabel transisi dari flip-flop J-K.

Dengan demikian kalau urutan perubahan dari keluaran flip-flop telah ditentukan maka dapat ditentukan tabel harga J dan K dari setiap flip-flop untuk setiap harga kombinasi  $Q_A Q_B Q_C$ . Dari tabel seperti ini dapat diperoleh persamaan J dan K sebagai fungsi dari  $Q_A$ ,  $Q_B$  dan  $Q_C$ .

$Q_n$	$Q_{n+1}$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Tabel 2.2 : Tabel transisi dari flip-flop J-K.

$Q_n$  : keluaran sebelum pulsa klok.

$Q_{n+1}$  : keluaran setelah pulsa klok.

### 2.1.2. PENCACAH BINER SINKRON MODULO-8

Urutan pencacahan dari pencacah ini dapat dilihat pada tabel dibawah ini :

Pulsa	$Q_A$	$Q_B$	$Q_C$
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

Tabel 2.3 : Tabel pencacahan dari pencacah biner sinkron modulo 8

Dari tabel (2. 3), pencacah biner sinkron modulo 8 mula-mula keluaran flip-flop dari pencacah ini  $Q_A Q_B Q_C = 000$  dan setelah ada pulsa klok masuk, keluaran flip-flop harus berubah menjadi 001. Dengan melihat tabel (2. 3), ini berarti bahwa pada keadaan  $Q_A Q_B Q_C = 000$ , setelah ada pulsa klok masuk berlaku :

$$\begin{array}{lll}
 Q_A = 0 & \text{--->} 0 & \text{maka } J_A = 0 \text{ dan } K_A = X \\
 Q_B = 0 & \text{--->} 0 & \text{maka } J_B = 0 \text{ dan } K_B = X \\
 Q_C = 0 & \text{--->} 1 & \text{maka } J_C = 1 \text{ dan } K_C = X
 \end{array}$$

Demikian juga pada keadaan  $Q_A Q_B Q_C = 001$ , yang berubah menjadi 010 kalau ada pulsa masuk, karena :

$$\begin{array}{lll}
 Q_A = 0 & \text{--->} 0 & \text{maka } J_A = 0 \text{ dan } K_A = X \\
 Q_B = 0 & \text{--->} 1 & \text{maka } J_B = 1 \text{ dan } K_B = X \\
 Q_C = 1 & \text{--->} 0 & \text{maka } J_C = 1 \text{ dan } K_C = 1
 \end{array}$$

Demikian seterusnya kalau hal ini dikerjakan untuk setiap harga  $Q_A Q_B Q_C$  yang mungkin dari pecacah biner sinkron modulo-8 maka terbentuklah tabel (2.4) yang merupakan tabel kebenaran dari J dan K, setiap flip-flop sebagai fungsi dari  $Q_A, Q_B$  dan  $Q_C$ .

Dari tabel (2.4), kalau dipilih semua  $J_A = X = 0$ , maka terlihat bahwa  $J_A = 1$  hanya pada keadaan  $Q_A Q_B Q_C = 011$ .

$Q_A Q_B Q_C$	$Q_A Q_B Q_C$	$J_A$	$K_A$	$J_B$	$K_B$	$J_C$	$K_C$
0 0 0	0 0 1	0	X	0	X	1	X
0 0 1	0 1 0	0	X	1	X	X	1
0 1 0	0 1 1	0	X	X	0	1	X
0 1 1	1 0 0	1	X	X	1	X	1
1 0 0	1 0 1	X	0	0	X	1	X
1 0 1	1 1 0	X	0	1	X	X	1
1 1 0	1 1 1	X	0	X	0	1	X
1 1 1	0 0 0	X	1	X	1	X	1

Tabel 2.4 : Tabel kebenaran dari J dan K setiap flip-flop sebagai fungsi dari  $Q_A, Q_B$  dan  $Q_C$  untuk pecacah biner sinkron modulo-8.

Secara persamaan Boole dapat ditulis sebagai berikut:

$$J_A = Q_A Q_B Q_C$$

Tetapi dengan memilih  $J_A = x = 1$ , untuk keadaan  $Q_A Q_B Q_C = 111$ , maka persamaannya menjadi lebih sederhana yaitu:

$$\begin{aligned} J_A &= Q_A Q_B Q_C + Q_A Q_B Q_C = Q_B Q_C (Q_A + Q_A) \\ &= Q_B Q_C \end{aligned}$$

Dengan cara yang sama diperoleh:

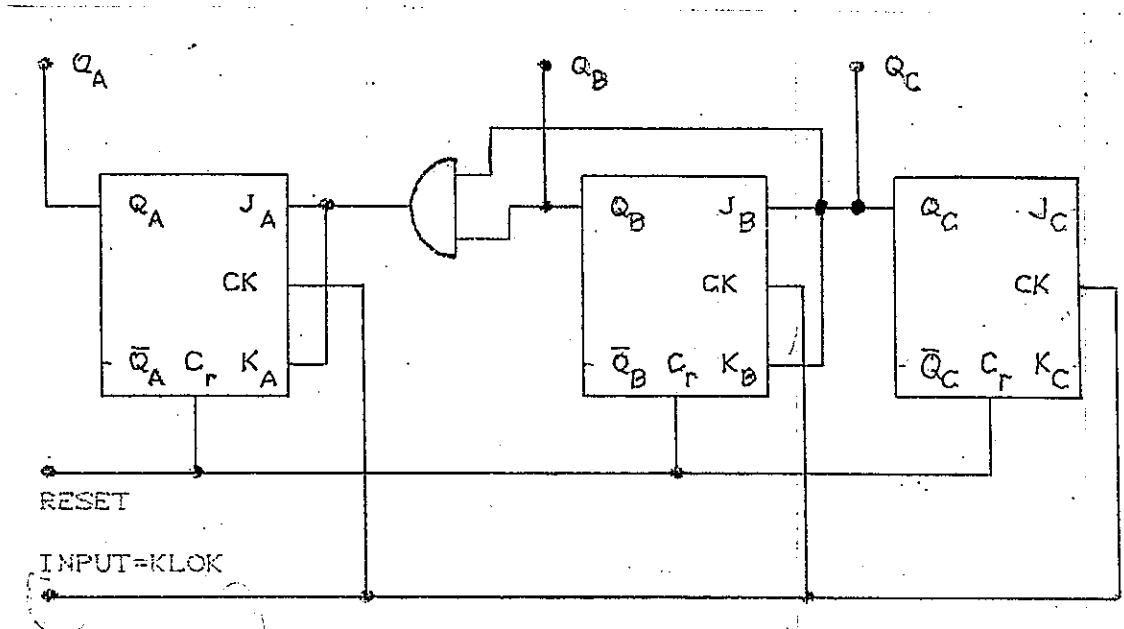
$$K_A = Q_B Q_C$$

$$K_B = Q_B$$

$$J_C = 1$$

$$K_C = 1$$

Atas dasar persamaan diatas, maka rangkaian pencacah biner sinkron modulo-8 berbentuk seperti pada gambar (2.1). Input RESET digunakan untuk membentuk (membuat)  $Q_A Q_B Q_C = 000$ .



Gambar 2. 1 : Rangkaian pencacah biner sinkron modulo-8

2.1.3. PENCACAH BINER SINKRON MODULO-6

Untuk pencacah ini diperlukan sebanyak 3 (tiga) buah flip-flop.

Pulsa	QA	QB	QC
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	0	0	0

Tabel (2. 5) Tabel Pencacahan pencacah Biner Sinkron Modulo-6.

$(Q_A Q_B Q_C)_n$	$(Q_A Q_B Q_C)_{n+1}$	$J_A$	$K_A$	$J_B$	$K_B$	$J_C$	$K_C$
0 0 0	0 0 1	0	X	0	X	1	X
0 0 1	0 1 0	0	X	1	X	X	1
0 1 0	0 1 1	0	X	X	0	1	X
0 1 1	1 0 0	1	X	X	1	X	1
1 0 0	1 0 1	X	0	0	X	1	X
1 0 1	0 0 0	X	1	0	X	X	1
.....	.....	.....	.....	.....	.....	.....	.....
1 1 0	0 0 0	X	X	X	X	X	X
1 1 1	0 0 0	X	X	X	X	X	X

Tabel 2.6 : Tabel kebenaran dari J dan K Setiap flip-flop sebagai fungsi dari  $Q_A, Q_B$  dan  $Q_C$  untuk pencacah biner sinkron modula-6.

Dari tabel pencacahan (tabel 2.5) dan tabel transisi dari flip-flop J-K (tabel 2.2), pengaturan harga dari J dan K dari setiap flip-flop sebagai fungsi dari  $Q_A, Q_B$  dan  $Q_C$  terlihat pada tabel (2.6) karena harga  $Q_A, Q_B, Q_C$  dari pencacah ini tidak pernah 110 dan 111, maka harga J dan K untuk harga ini boleh diisi sembarang (= X).

Untuk memilih persamaan yang paling sederhana akan lebih mudah kalau tabel kebenaran untuk J dan K sebagai fungsi dari  $Q_A, Q_B$  dan  $Q_C$  dituliskan dalam bentuk map Karnaugh, yang akan menghasilkan persamaan sebagai berikut.



$$J_A = Q_B Q_C$$

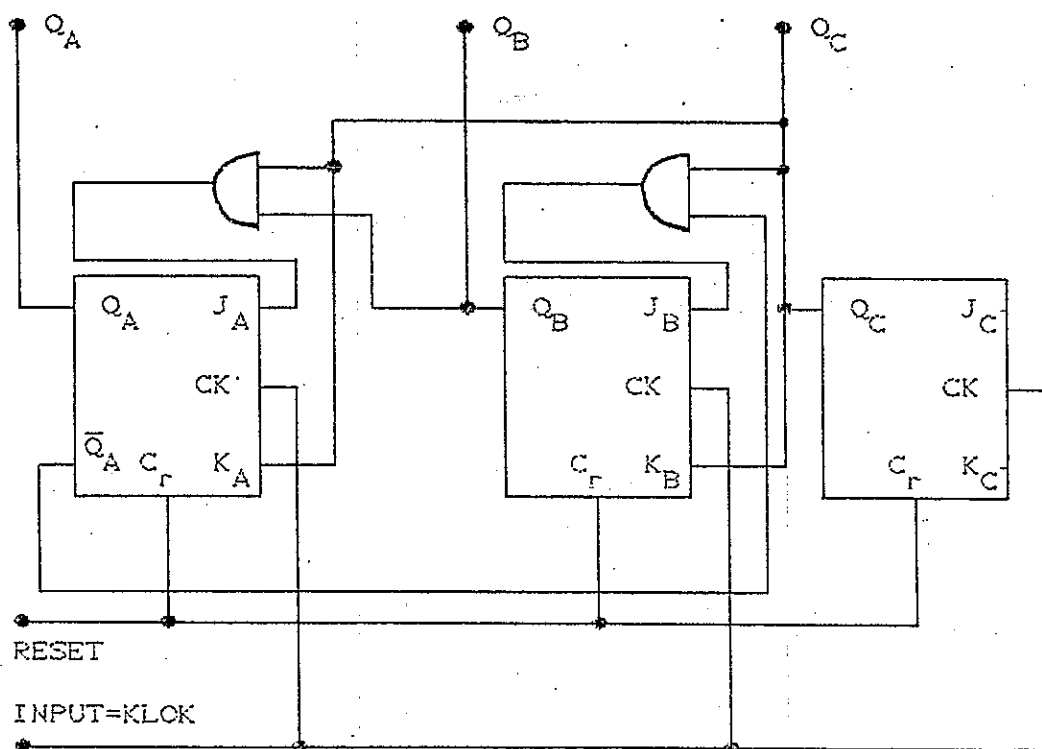
$$K_A = Q_C$$

$$J_B = Q_C \bar{Q}_A$$

$$K_B = Q_C$$

$$J_C = K_C = 1$$

Dengan demikian rangkaian pencacah biner sinkron modulo terlihat pada gambar (2.2).



Gambar 2.2 : Rangkaian pencacah biner sinkron modulo-6.

#### 2.1.4. PENCACAH B C D SINKRON

Tabel pencacahan dari pencacah BCD Sinkron ini dapat dilihat pada tabel dibawah ini :

Pulsa	$Q_A$	$Q_B$	$Q_C$	$Q_D$
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	0	0	0	0

Tabel 2. 7 Tabel Pencacahan untuk pencacah BCD Sinkron.

Pencacah ini menggunakan empat buah flip-flop. Dari tabel pencacahannya dan tabel transisi dari flip-flop J-K (tabel 2.2) dapat dibuat tabel kebenaran dari J dan K sebagai fungsi dari  $Q_A, Q_B, Q_C$  dan  $Q_D$ . Perlu diingat bahwa keluaran (output) dari pencacah ini yaitu  $Q_A Q_B Q_C Q_D$  tidak pernah mempunyai harga 1010, 1011, 1100, 1110 dan 1111, sehingga harga J dan K untuk keadaan tersebut diisi dengan X.

Berdasarkan pada tabel kebenarannya dan menggunakan map Karnough maka dapat dibuktikan bahwa persamaan paling sederhana yang diperoleh adalah :

$$J_A = Q_B Q_C Q_D$$

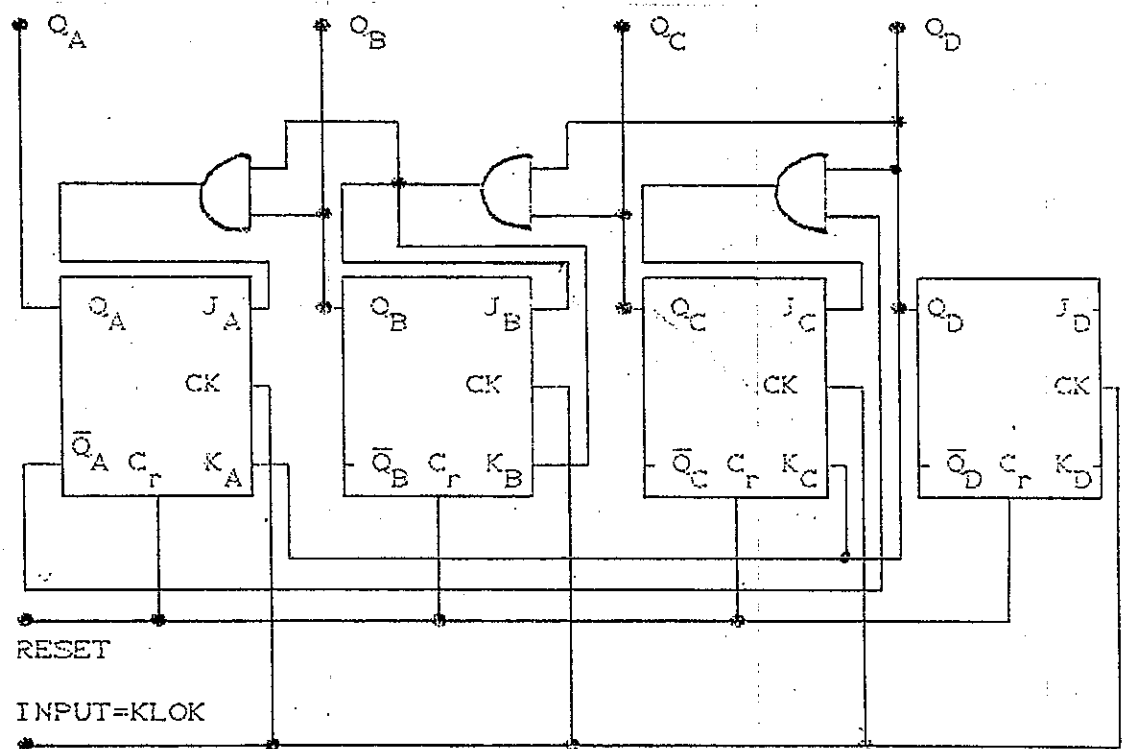
$$K_A = Q_D$$

$$J_B = K_B = Q_C Q_D$$

$$J_C = Q_A Q_D$$

$$K_C = Q_D$$

Sehingga rangkaian untuk pencacah BCD sinkron dapat digambarkan seperti pada gambar 2.3



Gambar 2.3 : Rangkaian pencacah BCD sinkron.

#### 2.1.5. PENCACAH TAK SINKRON

Pada pencacah tak sinkron perubahan dari keluaran flip-flop tidak terjadi secara serempak, karena pulsa masukan hanya dimasukkan pada flip-flop yang terdepan

(Least Significant Bit = LSB) sedang pulsa klok dari flip-flop yang lain diperoleh dari keluaran flip-flop di depannya. Dengan demikian perubahan dari keluaran flip-flop akan terjadi secara beruntun dari depan ke belakang, sehingga pencacah ini juga disebut dengan "pencacah simpal".

Kelebihan dari pencacah ini dibandingkan dengan pencacah sinkron adalah bahwa rangkaianya lebih sederhana, sehingga lebih murah. Hal ini disebabkan karena tidak setiap flip-flop menerima pulsa clock setiap terjadi pulsa masuk, sehingga pada keadaan-keadaan dimana suatu flip-flop tidak menerima pulsa klok pada saat terjadi pulsa masuk, harga dari J dan K nya boleh sembarang (X).

Dari pembicaraan sebelumnya diketahui bahwa semakin banyak harga X dalam tabel kebenaran J dan K semakin sederhana persamaan J dan K yang diperoleh, yang berarti bahwa rangkaian dari pencacah juga semakin sederhana. Hanya saja tidak setiap urutan pencacah dapat direalisasikan dengan pencacah tak sinkron.

#### 2.1.6. PENCACAH BINER TAK SINKRON MODULO-8

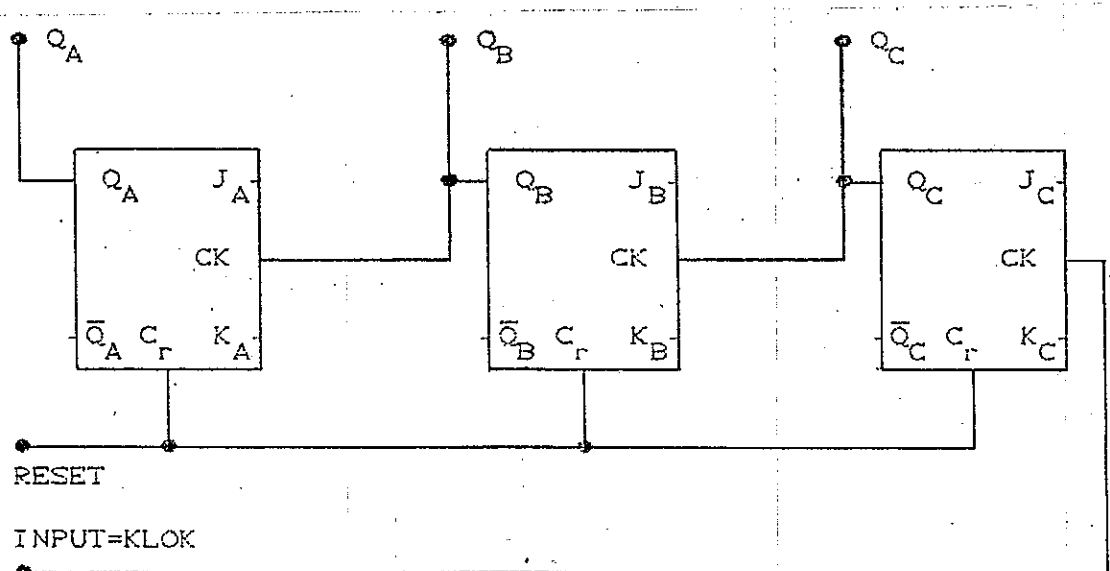
Urutan pencacahan dari pencacah ini terlihat pada tabel (2.8).

Dari tabel tersebut nampak bahwa flip-flop C yang masukan kloknnya dihubungkan dengan pulsa klok masukan.

Hal ini akan terjadi kalau diatur  $J_C = K_C = 1$ .

Untuk flip-flop B, yang diharapkan sebagai pulsa klok adalah  $Q_C$  : hal ini karena setiap  $Q_B$  berubah (dari 0 ke 1 atau dari 1 ke 0) selalu bersamaan dengan perubahan dari  $Q_C$  dari 1 ke 0 (pulsa efektif).

Jadi pada rangkaian, masukan (input) klok flip-flop B dihubungkan dengan  $Q_C$ . (lihat gambar 2.4).



Gambar 2. 4 : Rangkaian pencacah biner tak sinkron.  
modulo-8.

Selanjutnya untuk pengaturan harga  $J_B$  dan  $K_B$  untuk setiap harga  $Q_A Q_B Q_C$  yang mungkin, dari tabel (2. 8) dapat dilihat bahwa pada keadaan 0, 2, 4 dan 6, karena  $Q_C$  berubah dari 0 ke 1 kalau ada pulsa masuk, sedang perubahan  $Q_C$  dari 1 ke 0 merupakan pulsa klok yang tidak

diisi dengan X (lihat tabel 2. 9).

Sedang untuk keadaan yang lain, harga dari  $J_B$  dan  $K_B$  diatur sesuai dengan perubahan  $Q_B$  yang dikehendaki.

Untuk flip-flop A, yang dapat diharapkan sebagai pulsa klok adalah  $Q_B$  dan  $Q_C$ , dan keduanya mungkin karena setiap  $Q_A$  berubah selalu bersamaan dengan perubahan  $Q_B$  dari 1 ke 0 lebih sedikit dibandingkan dengan  $Q_C$ , maka dengan memilih  $Q_B$  sebagai pulsa klok dari flip-flop A, harga X untuk  $J_A$  dan  $K_A$  akan lebih banyak. (lihat tabel 2. 9).

Dari tabel (2. 9) tersebut terlihat bahwa kalau dipilih semua  $X = 1$ , maka akan diperoleh persamaan-persamaan sebagai berikut :

$$J_A = K_A = 1$$

$$J_B = K_B = 1$$

Pulsa	$Q_A Q_B Q_C$
0	0 0 0
1	0 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1
8	0 0 0

Tabel (2. 8) Tabel pencacahan untuk pencacah biner tak sinkron modulo-8.

$Q_A Q_B Q_C$	$Q_A Q_B Q_C$ n+1	J	$K_A$	$J_B$	$K_B$
0 0 0	0 0 1	X	X	X	X
0 0 1	0 1 0	X	X	1	X
0 1 0	0 1 1	X	X	X	X
0 1 1	1 0 0	1	X	X	1
1 0 0	1 0 1	X	X	X	X
1 0 1	1 1 0	X	X	1	X
1 1 0	1 1 1	X	X	X	X
1 1 1	0 0 0	X	1	X	1

Tabel (2.9) : Tabel kebenaran dari J dan K setiap flip-flop sebagai fungsi dari  $Q_A$ ,  $Q_B$  dan  $Q_C$  untuk pencacah biner tak sinkron

Dan rangkaian untuk pencacah tak sinkron modulo-8 ini ditunjukkan oleh gambar 2.4.

#### 2.1.7. PENCACAH BINER TAK SINKRON MODULO-6

Untuk pencacah ini tabel pencacahannya nampak pada tabel (2.10).

Pulsa	$Q_A Q_B Q_C$
0	0 0 0
1	0 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	0 0 0
7	dst

Tabel 2. 10 : Tabel pencacahan untuk pencacah biner tak Sinkron modulo-6.

Pada tabel (2. 10) nampak bahwa, karena  $Q_C$  selalu berubah setiap ada pulsa masuk, maka perlu diatur  $J_C = K_C = 1$ . Dan nampak pula pada tabel tersebut bahwa perubahan dari  $Q_A$  dan  $Q_B$  dari 0 ke 1 atau dari 1 ke 0 selalu bersamaan dengan perubahan  $Q_C$  dari 1 ke 0.

Dengan demikian  $Q_C$  dapat digunakan sebagai input klok dari flip-flop A dan flip-flop B.

Sedang tabel kebenaran dari  $J_A$ ,  $K_A$ ,  $J_B$  dan  $K_B$  dapat dilihat pada tabel (2. 11).

Perubahan  $Q_C$  dari 0 ke 1 merupakan pulsa klok yang tidak efektif sehingga J dan K dari flip-flop A dan flip-flop B pada perubahan ini dapat diisi dengan X.

Dari tabel (2. 11) mudah dilihat bahwa  $K_A = K_B = 1$  sedang untuk  $J_A$  dan  $J_B$  dengan membuat map Karnoughnya dapat dibuktikan bahwa :

$$J_A = Q_B$$

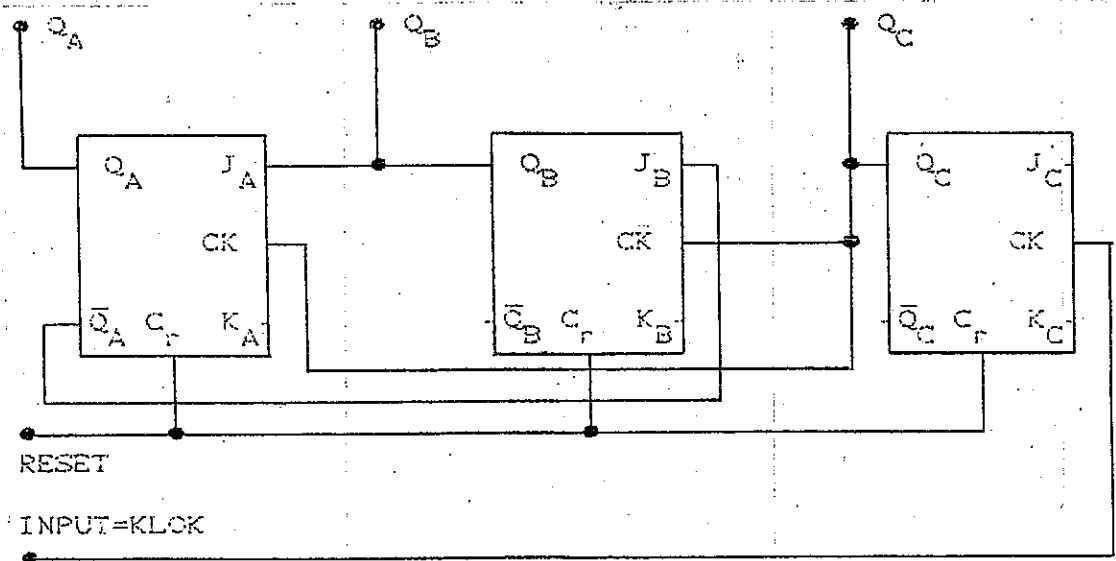
$$J_B = Q_A$$

Dengan demikian rangkaian dari pencacah biner tak sinkron modulo-6 dapat digambarkan seperti nampak pada gambar 2. 5.



$Q_A$	$Q_B$	$Q_C$	$Q_A Q_B Q_C$	$J_A$	$K_B$	$J_B$	$K_B$
0	0	0	0 0 1	X	X	X	X
0	0	1	0 1 0	0	X	1	X
0	1	0	0 1 1	X	X	X	X
0	1	1	1 0 0	1	X	X	1
1	0	0	1 0 1	X	X	X	X
1	0	1	0 0 0	X	1	0	X
.....							
1	1	0	0 0 0	X	X	X	X
1	1	1	0 0 0	X	X	X	X

Tabel 2.11 : Tabel kebenaran dari J dan K setiap flip-flop sebagai fungsi dari  $Q_A$ ,  $Q_B$  dan  $Q_C$  untuk pencacah biner tak sinkron modulo-6.



Gambar 2.5 : Rangkaian pencacah biner tak sinkron modulo-6.

### 2.1.8. PENCACAH B C D TAK SINKRON

Dari urutan pencacahannya yang terlihat pada tabel dapat disimpulkan bahwa :

1. Agar  $Q_D$  berubah setiap ada pulsa masuk, maka perlu diatur  $J_D = K_D = 1$
2. Sebagai pulsa klok dari flip-flop C adalah  $Q_D$ , sedangkan sebagai pulsa klok dari flip-flop B adalah  $Q_C$ .

Selanjutnya seperti pada pembicaraan sebelumnya, perlu dibuat tabel kebenaran dari :  $J_A$ ,  $K_A$ ,  $J_B$ ,  $K_B$ ,  $J_C$  dan  $K_C$  dan ini dapat dilihat pada tabel. Untuk keadaan dibawah garis putus-putus semua diisi dengan X karena keluaran (output) dari pencacah tidak pernah mempunyai harga ini.

Dari tabel 2.12 tersebut mudah dilihat bahwa untuk  $K_A$ ,  $J_B$  dan  $K_C$  kalau dipilih semua  $X = 1$ , maka didapat persamaan :

$$K_A = J_B = K_B = K_C = 1$$

Sedangkan untuk  $J_A$  dan  $J_C$  dengan map Karnough nya mudah dibuktikan bahwa :

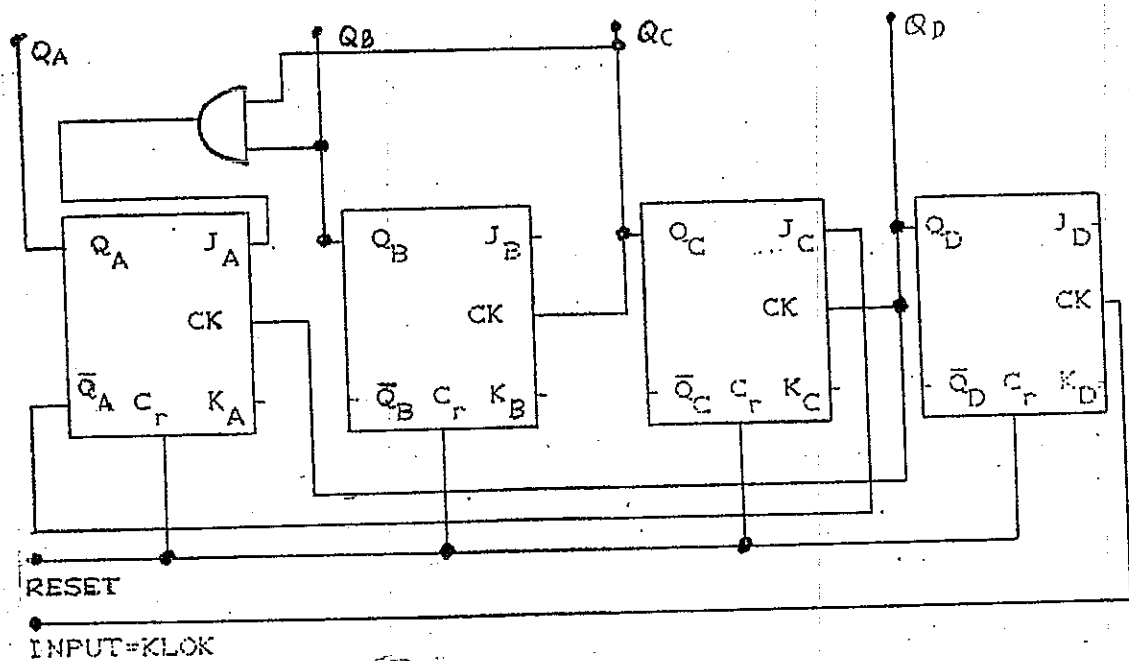
$$J_A = Q_B Q_C$$

$$J_C = Q_B$$

Dari persamaan yang diperoleh maka dapat disusun rangkaian BCD tak sinkron seterti gambar 2.6

$(Q_A Q_B Q_C Q_D)$	$(Q_A Q_B Q_C Q_D)$	$J_A$	$K_A$	$J_B$	$K_B$	$J_C$	$K_C$
0 0 0 0	0 0 0 1	X	X	X	X	X	X
0 0 0 1	0 0 1 0	0	X	X	X	1	X
0 0 1 0	0 0 1 1	X	X	X	X	X	X
0 0 1 1	0 1 0 0	0	X	1	X	X	1
0 1 0 0	0 1 0 1	X	X	X	X	X	X
0 1 0 1	0 1 1 0	0	X	X	X	1	X
0 1 1 0	0 1 1 1	X	X	X	X	X	X
0 1 1 1	1 0 0 0	1	X	X	1	X	1
1 0 0 0	1 0 0 1	X	X	X	X	X	X
1 0 0 1	0 0 0 0	X	1	X	X	0	X
1 0 1 0	1 0 1 0	X	X	X	X	X	X
1 0 1 1	1 0 1 1	X	X	X	X	X	X
1 1 0 0	1 1 0 0	X	X	X	X	X	X
1 1 0 1	1 1 0 1	X	X	X	X	X	X
1 1 1 0	1 1 1 0	X	X	X	X	X	X
1 1 1 1	1 1 1 1	X	X	X	X	X	X

Tabel 2.12 : Tabel kebenaran dari J dan K setiap flip-flop sebagai fungsi dari  $Q_A$ ,  $Q_B$ ,  $Q_C$  dan  $Q_D$  untuk pencacah BCD tak sinkron.



Gambar 2. 6. Pencacah BCD Tak Sinkron

## 2.2. REGISTER

Register adalah suatu rangkaian logika yang dibentuk oleh beberapa flip-flop yang disusun sedemikian rupa sehingga mampu menyimpan dan memproses informasi. Di dalam sistem elektronika digital register pada umumnya digunakan untuk menyimpan data secara sementara, untuk kemudian diproses atau diganti dengan data yang lain.

Pada suatu pencacah frekuensi yang bekerja terus menerus misalnya, agar setiap kali hasil pengukuran dapat ditampilkan selama pengukuran berikutnya dikerjakan, data dari setiap pengukuran disimpan pada register yang dihubungkan dengan penampil lewat suatu decoder. Data ini setiap kali diganti data pengukuran yang baru.

Penyimpanan dan pengambilan data dari suatu register dapat dilakukan dengan dua cara yaitu :

- Cara Paralel :

Data yang terdiri beberapa bit dimasukkan atau dikeluarkan secara serempak.

- Cara Serial :

Data bit demi bit dimasukkan atau dikeluarkan secara beruntun.

Sehingga berdasar pada operasi tersebut, register dibedakan menjadi empat macam yaitu :

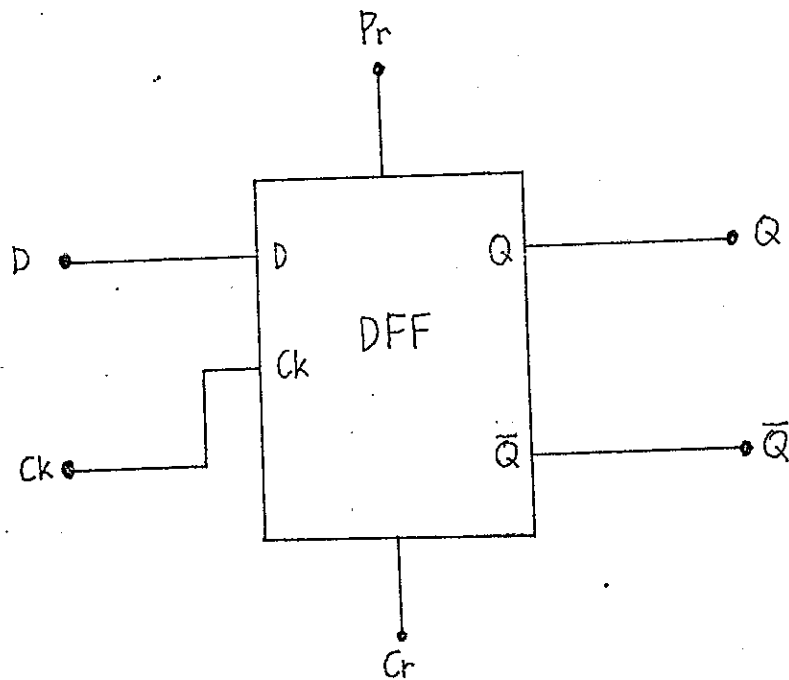
1. Paralel input - paralel output
2. Serial input - paralel output
3. Serial input - serial output
4. Paralel input - serial output

Karena register terdiri atas flip-flop, akan ditinjau flip-flop yang digunakan yaitu Flip-Flop D. Jika suatu flip-flop J-K diubah dengan penambahan suatu pembalik sehingga K merupakan komplemen dari J maka sistem yang dihasilkan disebut flip-flop D (delay). Keluaran  $Q_{n+1}$  sesudah pulsa yang bersangkutan sama dengan masukan  $D_n$  sebelum pulsa yang sama, seperti ditunjukkan dalam tabel 2.13.

Flip-flop tipe D merupakan suatu sistem biner yang digunakan untuk menimbulkan penundaan sinyal. Hit data pada saluran D dipindahkan ke keluarannya pada pulsa clock berikutnya, dengan demikian sistem ini berfungsi sebagai alat penunda satu bit.

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

Tabel 2.13 : Tabel Kebenaran dari Flip-flop D

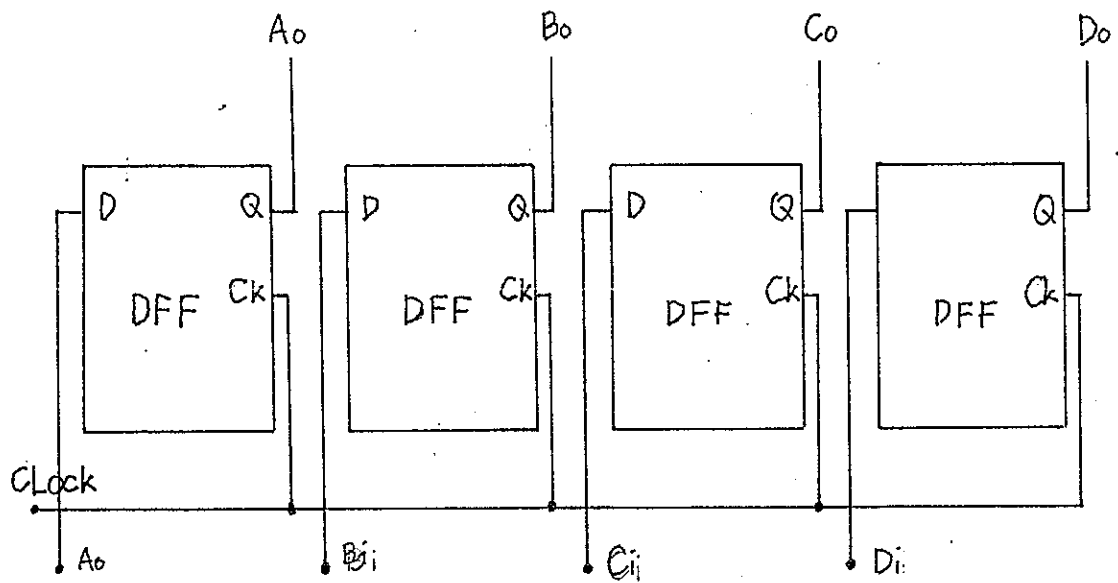


Gambar 2.7 : D Flip-flop dengan Preset dan Clear

#### 2.2.1. REGISTER GESER 4 BIT: PARALEL INPUT-PARALEL OUTPUT

Dengan memakai empat buah flip-flop D dan beberapa gerbang nalar dapat dibentuk sebuah register 4 bit paralel input - paralel output, seperti nampak pada gambar 2.8.

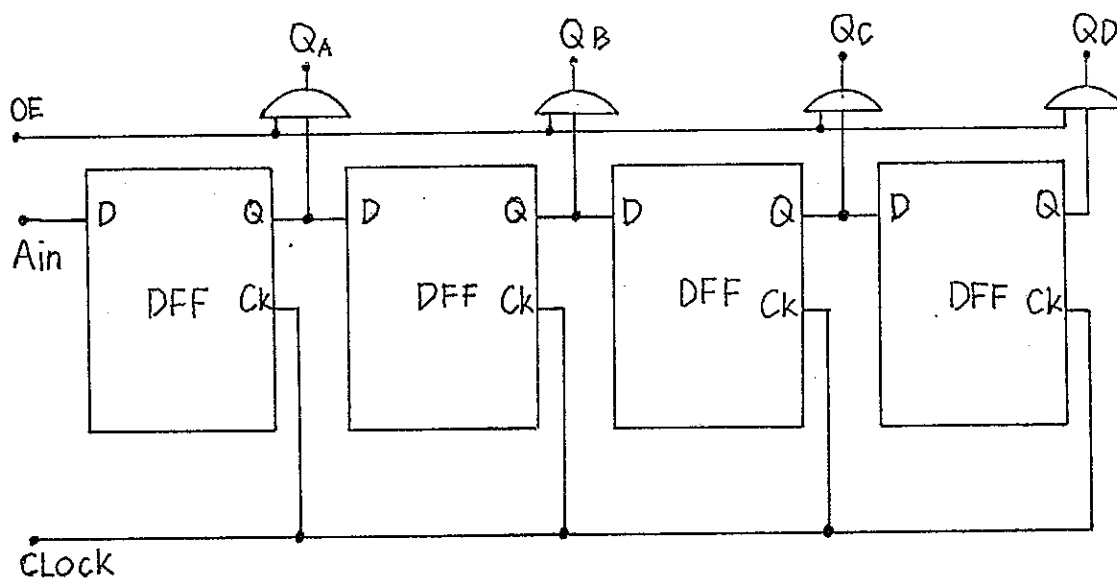
Jika data 4 bit dimasukkan secara bersama-sama pada input yaitu A B C D, dan kemudian memberikan pulsa clock, maka  $Q_A Q_B Q_C Q_D$  akan sama dengan  $(ABCD)_{in}$ .



Gambar 2.8 : Rangkaian register 4 bit paralel in - paralel out.

#### 2.2.2. REGISTER GESER 4 BIT : SERIAL INPUT-PARAREL OUTPUT

Untuk dapat memasukan data secara serial, maka output dari flip-flop yang satu dihubungkan ke input flip-flop berikutnya seperti nampak pada gambar (2.9)



Gambar 2.9 : rangkaian register 4 bit serial in - parallel out.

Dengan sistem seperti pada gambar (2.9) diatas maka sesaat setiap pulsa klok masuk, akan terjadi perubahan sehingga  $Q_A = A_{in}$ ,  $Q_B = Q_A$ ,  $Q_C = Q_B$  dan  $Q_D = Q_C$ . Atau dengan kata lain terjadi pergeseran ke kanan, sehingga untuk memasukan data secara serial, bit demi bit dari data dari data dimasukan pada  $A_{in}$ , sinkron dengan terjadinya pulsa klok.

Misal saja untuk memasukan data 1110 secara serial.



Atur  $A_{in} = 0$  pada pulsa klok pertama dan  $A_{in} = 1$  pada pulsa klok kedua, ketiga dan keempat.

Sételah pulsa keempat seharusnya terjadi  $Q_A Q_B Q_C Q_D = 1110$ , sehingga kalau kemudian diatur  $OE = 1$  maka data akan keluar secara paralel pada  $(ABCD)_{out}$ .

### 2.2.3. REGISTER GESER 4 BIT : SERIAL INPUT-SERIAL OUTPUT

Andaikan data 4 bit telah dimasukan secara serial pada register gambar (2.9) sehingga andaikan harga  $Q_A Q_B Q_C Q_D = 1110$ .

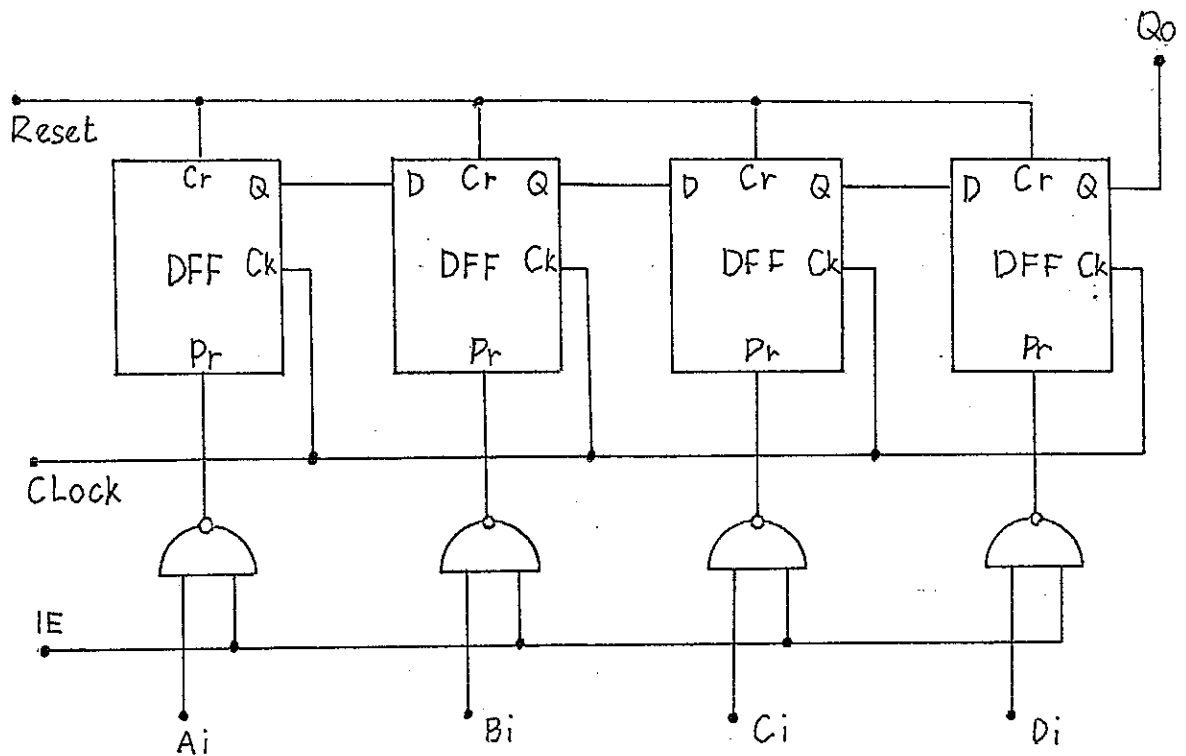
Maka data tersebut dapat dikeluarkan secara serial lewat  $D_o$  dengan cara memasukan pulsa klok tiga kali. Dengan demikian : mula-mula  $D_o = Q_D = 0$ , maka kemudian

- setelah pulsa klok pertama  $Q_D = Q_C = 1$  ;
- setelah pulsa klok kedua  $Q_D = Q_B = 1$  ;
- setelah pulsa klok ketigaa  $Q_D = Q_A = 1$  ;

Dengan demikian data yang ada pada  $Q_A Q_B Q_C Q_D$  dikeluarkan bit demi bit lewat  $Q_D (=d_o)$  dengan data paling awal adalah  $Q_D$  dan yang paling akhir adalah  $Q_A$ .

### 2.2.4. REGISTER 4 BIT : PARALEL INPUT-SERIAL OUTPUT

Rangkaian register model ini dapat dilihat pada gambar 2.10.



Gambar 2.10 : Rangkaian register 4 bit paralel in - serial out

Seperti telah dikemukakan di muka bahwa DFF dapat bekerja sebagaimana mestinya harus diatur  $Pr=Cr=1$ . Dari keadaan ini reset diubah sebentar ke 0 kemudian ke 1 maka harga outputnya akan berubah ke 0 semuanya.

Kalau dari keadaan ini kemudian harga IE diubah ke 1, harga  $Q_A$ ,  $Q_B$ ,  $Q_C$ ,  $Q_D$ , masing-masing berubah ke 1 atau tetap 0 sesuai dengan harga inputnya. Cara inilah yang digunakan untuk memasukkan data secara paralel.

### 2.3. DEKODER

Dalam suatu sistem digital, instruksi-instruksi maupun bilangan-bilangan dikirim dengan deretan pulsa atau tingkatan-tingkatan biner. Misalnya, disediakan karakter 4-bit untuk pengiriman instruksi, maka jumlah instruksi berbeda yang dapat dibuat adalah 16. Informasi ini diberi kode (sandi) biner. Di pihak lain seringkali timbul kebutuhan akan suatu saklar multiposisi yang dapat dioperasikan sesuai dengan kode tersebut. Dengan kata lain, untuk masing-masing dari enam belas kode itu, hanya satu saluran yang dieksitasi pada setiap saat. Proses untuk identifikasi suatu kode tertentu ini disebut pedekodean (dekoding).

Sistem BCD (Binary-Coded-Decimal (desimal-dalam kode-biner)). Sistem kode ini menterjemahkan bilangan-bilangan desimal dengan menggantikan setiap digit desimal dengan suatu kombinasi dari 4-digit biner. Mengingat bahwa dari 4 digit biner dapat dibuat 16 macam kombinasi digit yang berbeda, maka 10 diantaranya dapat digunakan untuk menyatakan digit desimal dari 0 sampai 9. Dengan ini kita mempunyai pilihan kode BCD yang luas. Salah satu pilihan yang disebut "BCD alamiah (natural binary decimal)" ini adalah kode 8421 yang digambarkan oleh 10 baris pertama dalam tabel 2.17. Kode ini merupakan kode dengan bobot (weighted code) karena digit desimal dalam kode 8421 ini sama dengan

karena digit desimal dalam kode 8421 ini sama dengan jumlah dari hasil perkalian masing-masing bit dalam kode yang disandi, dengan pangkat berturut-turut dari angka 2 mulai dari LSB. Untuk bilangan desimal N-digit diperlukan N gugus biner 4-bit bagi representasi BCD yang bersangkutan. Gugus 4-bit pertama disebelah kanan memberi representasi digit desimal dari 0 sampai 9, gugus kedua memberi representasi digit puluhan, gugus ketiga memberi representasi digit ratusan, dan seterusnya. Sebagai contoh bilangan desimal 64 memerlukan dua gugus biner 4-bit seperti ditunjukkan dalam tabel 2.14

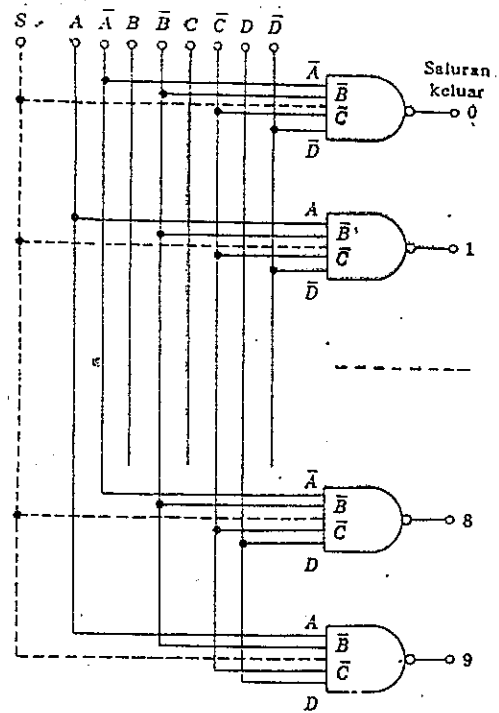
---

Faktor bobot.....!	80	40	20	10	!	8	4	2	1
Kode BCD .....	0	1	1	0	!	0	1	0	0
Digit desimal.....!		6			!		4		

---

Tabel 2.14. Representasi BCD untuk bilangan desimal 64

Suatu pendekode BCD-ke-desimal diperlihatkan dalam Gambar 2.11. Unit MSI ini mempunyai empat masukan A, B, C dan D, dan 10 saluran. Disamping itu harus disediakan pula hubungan-hubungan tanah dan catu-daya. Masukan komplementer A, B, C dan D diperoleh dengan pembalik pada serpih yang sama. Pendekode ini bekerja sebagai suatu saklar 10 posisi yang memberi tanggapan kepada suatu instruksi masukan BCD.



Gambar 2.11. Dekode BCD-ke-desimal.