

Percobaan 6

PENCACAH (*COUNTER*)

Oleh : Sumarna, Jurdik Fisika, FMIPA, UNY
E-mail : sumarna@uny.ac.id

Tujuan :

1. Mempelajari cara kerja pencacah biner sinkron dan tak sinkron,
2. Merealisasikan pencacah biner sinkron dan tak sinkron dengan flip-flop JK,
3. Membuktikan tabel kebenaran pencacah biner sinkron dan tak sinkron.

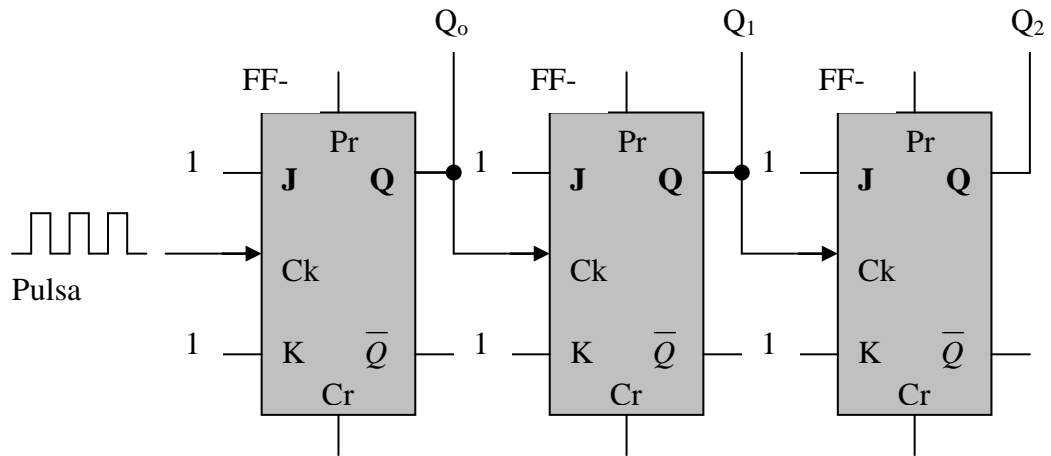
Alat-alat Percobaan :

Catu daya dc +5 volt, IC-7408, IC-7476, IC-7400, resistor, saklar SPDT, papan rangkaian (breadboard), multimeter, LED, kabel-kabel penghubung.

Dasar Teori :

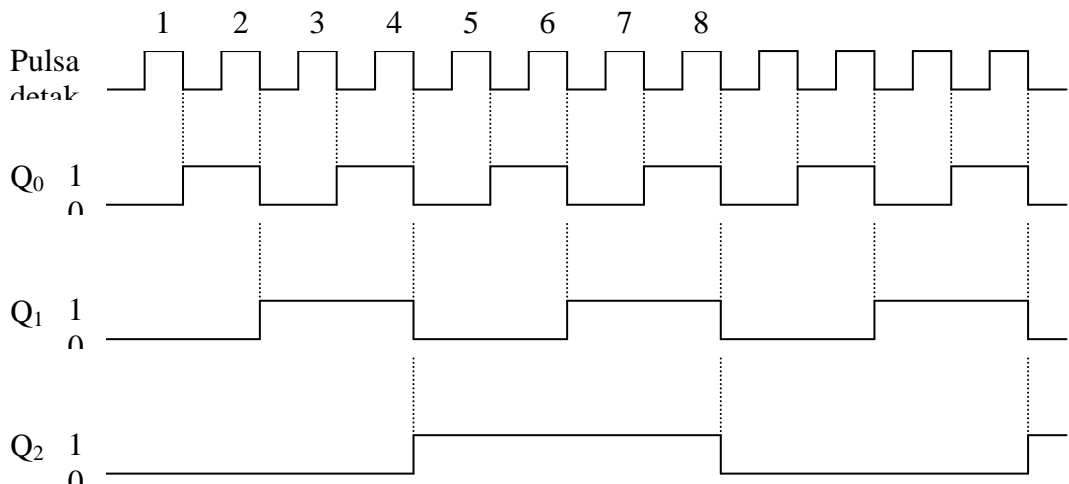
Pencacah (Counter)

Teknologi sistem digital menjadi semakin semarak antara lain karena peranan flip-flop yang dapat menampilkan fungsi baik sebagai pencacah (counter) maupun sebagai register. Dalam kehidupan sehari-hari sering dijumpai mesin-mesin digital yang bekerja sebagai pencatat seperti pencatat waktu, pencatat frekuensi, pencatat bilangan, pencatat banyaknya orang yang memasuki suatu tempat, dan sebagainya. Mesin-mesin pencatat itu bekerja sebagai pencacah. Pada dasarnya mesin-mesin tersebut mencacah pulsa tegangan. Pencacah termasuk dalam kelompok rangkaian sekuensial yang merupakan gabungan antara rangkaian kombinasional dan flip-flop. Dengan demikian flip-flop merupakan komponen utama untuk menyusun rangkaian pencacah. Semua jenis flip-flop yang dilengkapi dengan fasilitas Clock (Ck), Preset (Pr), ataupun Clear (Cr) dapat digunakan untuk menyusun rangkaian pencacah. Tetapi dalam bab ini akan banyak menggunakan flip-flop JK atau JK-MS. Perhatikan Gambar berikut.



Gambar : Tiga FF-JK yang disusun secara serial.

Setiap FF memiliki masukan $J = K = 1$, sehingga keluaran FF itu akan ter-toggle (berubah tingkat logikanya) ketika pada masukan Ck berubah dari 1 (tinggi) ke 0 (rendah). Pulsa masukan hanya dikenakan pada Ck dari FF-0. Keluaran Q_0 dihubungkan ke Ck pada FF-1, dan keluaran Q_1 dihubungkan ke Ck dari FF-2. Bentuk gelombang keluaran pada setiap FF dapat dilihat pada Gambar di bawah ini.



Gambar : Bentuk gelombang keluaran dari tiga FF yang disusun seri.

Sebelumnya dianggap bahwa ketiga keluaran FF pada tingkat logika 0, dan setiap FF hanya dapat berubah keadaan jika pada masukan Ck terjadi transisi dari 1 ke 0 (NGT : Negative Going Transision). Berdasarkan keadaan itu maka hal-hal yang perlu diperhatikan adalah :

- 1) Keluaran FF-0, yaitu Q_0 , mengalami perubahan keadaan tingkat logika (ter-toggle) pada setiap kali pulsa detak pada Ck-nya mengalami transisi dari 1 ke 0, sehingga gelombang keluaran pada Q_0 memiliki frekuensi tepat sama dengan setengah (1/2) dari frekuensi pulsa detak masukan.
- 2) Keluaran FF-1, yaitu Q_1 , mengalami perubahan keadaan tingkat logika (ter-toggle) pada setiap kali pulsa detak pada Ck-nya mengalami transisi dari 1 ke 0, sehingga gelombang keluaran pada Q_1 memiliki frekuensi tepat sama dengan setengah (1/2) dari frekuensi Q_0 atau seperempat (1/4) dari frekuensi pulsa detak masukan.
- 3) Keluaran FF-2, yaitu Q_2 , mengalami perubahan keadaan tingkat logika (ter-toggle) pada setiap kali pulsa detak pada Ck-nya mengalami transisi dari 1 ke 0, sehingga gelombang keluaran pada Q_2 memiliki frekuensi tepat sama dengan setengah (1/2) dari frekuensi Q_1 atau seperdelapan (1/8) dari frekuensi pulsa detak masukan.

Berdasarkan keterangan di atas, setiap FF membagi frekuensi masukannya dengan 2 (dua). Jika sekiranya ditambahkan FF ke empat pada rantai Gambar di atas dengan keluaran Q_3 akan memiliki frekuensi seper-enam-belas (1/16) dari frekuensi pulsa detak masukan, demikian seterusnya. Secara umum dapat dikemukakan bahwa dengan N buah FF yang disusun serial akan menghasilkan frekuensi keluaran pada FF terakhir sebesar f_N dan dituliskan sebagai :

$$f_N = (1/2^N).f_0$$

dengan f_0 adalah frekuensi detak masukan mula-mula. FF yang disusun seperti pada Gambar di atas membentuk rangkaian yang dikenal sebagai **pembagi frekuensi**.

Pencacah Biner Tak Sinkron (Serial atau Riak)

Di samping fungsi sebagai pembagi frekuensi, rangkaian seperti Gambar di atas juga bekerja sebagai pencacah biner. Hal ini dapat ditunjukkan dengan menyelidiki

sederetan keadaan setiap FF setelah suatu pulsa detak berubah dari 1 ke 0. Deretan keadaan tersebut tampak pada tabel berikut :

Q ₂	Q ₁	Q ₀	
0	0	0	sebelum dikenakan pulsa detak (keadaan awal)
0	0	1	sesudah pulsa 1 (pertama)
0	1	0	sesudah pulsa 2 (ke dua)
0	1	1	sesudah pulsa 3 (ke tiga)
1	0	0	sesudah pulsa 4 (ke empat)
1	0	1	sesudah pulsa 5 (ke lima)
1	1	0	sesudah pulsa 6 (ke enam)
1	1	1	sesudah pulsa 7 (ke tujuh)
0	0	0	sesudah pulsa 8 (ke delapan), kembali berputar ke keadaan awal.

Misalkan nilai kombinasi dari $Q_2Q_1Q_0$ merupakan bilangan biner, di mana Q_2 adalah posisi 2^2 (empatan), Q_1 adalah posisi 2^1 (duaan), dan Q_0 adalah posisi 2^0 (satuan). Delapan deret yang pertama dari keadaan $Q_2Q_1Q_0$ pada tabel di atas akan dikenal sebagai serial perhitungan biner dari 000 ke 111. Sesudah pulsa pertama FF-FF itu dalam keadaan 001 ($Q_2 = 0$, $Q_1 = 0$, dan $Q_0 = 1$) yang menggambarkan biner 001_2 (setara dengan desimal 1); sesudah pulsa ke dua FF-FF itu dalam keadaan 010 ($Q_2 = 0$, $Q_1 = 1$, dan $Q_0 = 0$) yang menggambarkan biner 010_2 (setara dengan desimal 2); sesudah pulsa ke tiga FF-FF itu dalam keadaan 011 ($Q_2 = 0$, $Q_1 = 1$, dan $Q_0 = 1$) yang menggambarkan biner 011_2 (setara dengan desimal 3); dan seterusnya sesudah pulsa ke tujuh FF-FF itu dalam keadaan 111 ($Q_2 = 1$, $Q_1 = 1$, dan $Q_0 = 1$) yang menggambarkan biner 111_2 (setara dengan desimal 7). Pada akhir pulsa ke delapan ketiga FF kembali ke keadaan 000 dan serial biner terulang dengan sendirinya untuk pulsa detak yang berturut-turut. Pada kejadian tersebut tampak bahwa untuk tujuh pulsa masukan yang pertama, fungsi rangkaian itu sebagai pencacah biner di mana keadaan dari FF-FF tersebut menggambarkan bilangan biner yang setara dengan banyaknya pulsa yang telah terjadi. Pencacah di atas menghitung nilai tertinggi $111_2 = 7_{10}$ (8 keadaan) dan selanjutnya kembali ke 000.

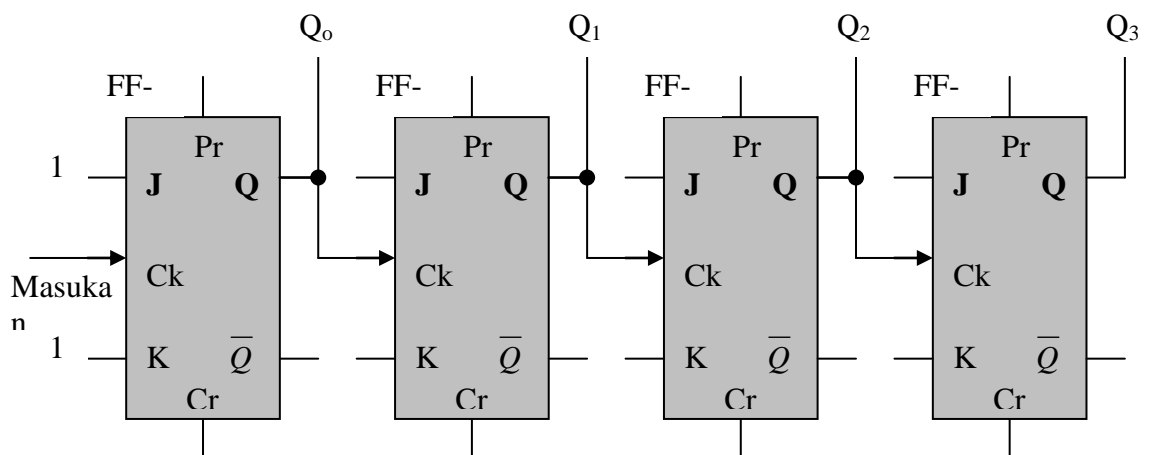
Jenis pencacah dengan konfigurasi seperti pada Gambar di atas dikenal sebagai pencacah **tak sinkron** atau pencacah **serial** atau pencacah **riak** (ripple), di mana setiap

keluaran FF bertindak sebagai sinyal masukan Ck pada FF berikutnya. Keadaan keluaran setiap FF berubah secara tidak bersamaan (tak sinkron) terhadap pulsa detak. Hanya FF-0 yang dikenai pulsa detak. Perubahan keadaan keluaran FF-1 menunggu hingga terpicu dari FF-0. Perubahan keadaan keluaran FF-2 menunggu hingga terpicu dari FF-1, demikian seterusnya.

Pencacah seperti pada Gambar di atas memiliki $2^3 = 8$ keadaan yang berbeda (dari 000 s/d 111) dan dikenal sebagai pencacah modulo 8 (tepatnya sebagai pencacah biner tak sinkron modulo 8 atau 3 bit). Jika sebuah FF sejenis ditambahkan menyambung, seperti tampak pada Gambar 10.3, maka deretan keadaan yang berbeda yang akan dihitung dalam biner ada sebanyak 16 keadaan dari 0000 s/d 1111. Pencacah tersebut kemudian memiliki modulo 16 atau 4 bit. Secara umum dapat dikemukakan bahwa jika ada N buah FF (atau N bit) yang disusun seperti gambar di bawah ini maka pencacah itu memiliki 2^N keadaan yang berbeda dan dikatakan bahwa pencacah tersebut merupakan pencacah biner (N bit) modulo 2^N . Pencacah modulo 2^N akan mampu menghitung dari nol hingga setinggi 2^N-1 sebelum kembali ke keadaan nol-nya. Jelas bahwa bilangan modulo selalu sama dengan banyaknya keadaan pada mana pencacah terus berubah pada setiap siklus sebelum kembali ke keadaan awalnya.

$$\text{Banyaknya Modulo} = 2^N.$$

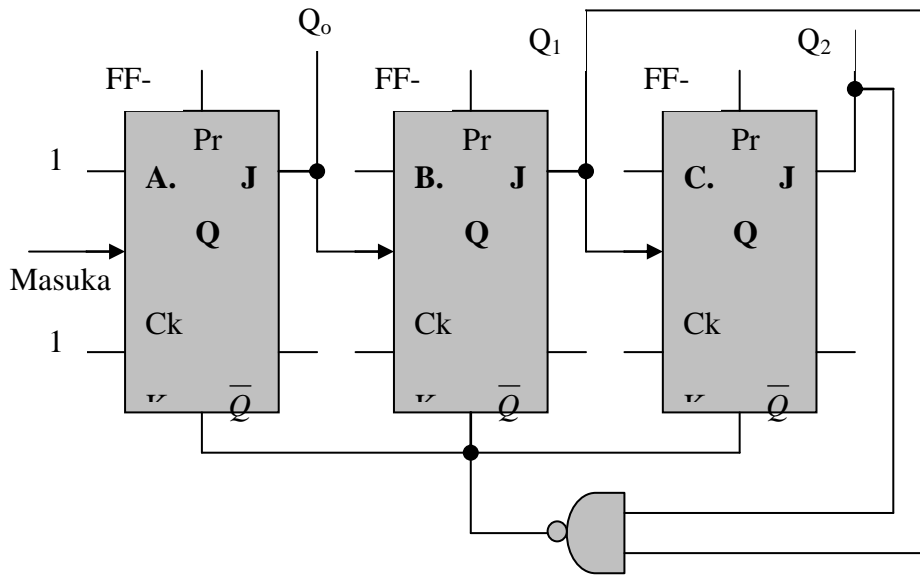
Bilangan modulo suatu pencacah tak sinkron juga menunjukkan pembagian frekuensi yang diperoleh pada FF terakhir.



Gambar : Pencacah biner (4 bit) tak sinkron modulo 16.

Ingat bahwa untuk semua FF ditentukan $J = K = 1$. Detak (pulsa) masukan dikenakan pada Ck dari FF-0. Kemudian keluaran FF-0 akan berubah ke keadaan yang berlawanan (ter-toggle) setiap kali detak masukan membuat transisi dari tinggi ke rendah (transisi negatif). Keluaran (Q) dari FF-0 dikenakan masukan Ck pada FF-1, sehingga keluaran FF-1 akan ter-toggle pada setiap kali Ck-nya mengalami transisi negatif. Dengan cara yang sama, FF-2 dan FF-3 akan ter-toggle setiap kali Ck-nya mengalami transisi negatif. Jika detak dikenakan terus menerus, maka kombinasi keluaran setiap FF menggambarkan bilangan biner (4 bit) dengan keluaran FF-3 sebagai MSB dan keluaran FF-0 sebagai LSB. Pada akhir detak ke 15 keluaran semua FF menunjukkan keadaan 1111. Pada detak ke 16 keluaran FF-0 berubah dari 1 ke 0, yang mengakibatkan keluaran FF-1 berubah dari 1 ke 0 dan seterusnya hingga keluaran semua FF menunjukkan keadaan 0000. Dengan kata lain, pencacah tersebut telah melewati satu siklus penuh (dari 0000 hingga 1111) dan kembali ke keadaan 0000. Dari keadaan 0000 akan dimulai siklus perhitungan baru untuk deretan detak berikutnya.

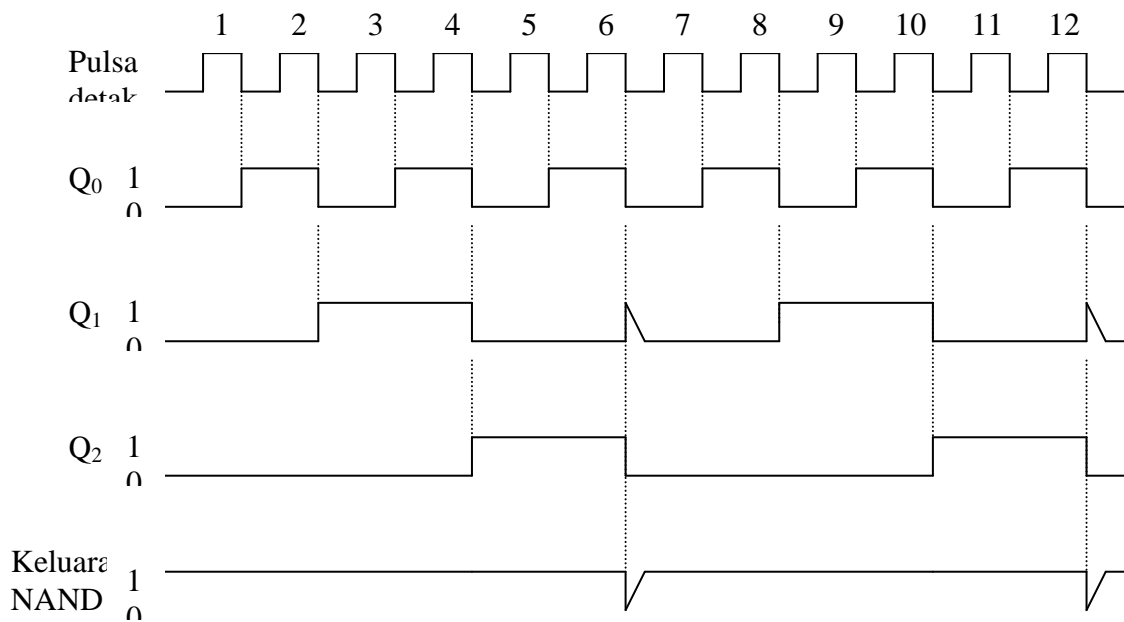
Pencacah biner tak sinkron yang telah dikemukakan di atas terbatas pada bilangan modulo 2^N . Dengan demikian modulo pencacah yang dapat dihasilkan adalah 1, 2, 4, 8, 16, 32, ... 2^N . Nilai tersebut sebenarnya merupakan bilangan modulo terbesar yang dapat diperoleh dari penggunaan N buah FF. Bila diinginkan pencacah dengan bilangan modulo yang lebih kecil dari pada 2^N maka dapat dilakukan dengan memodifikasi pencacah modulo 2^N . Dasar modifikasinya adalah membuat pencacah tersebut melompati keadaan yang secara normal merupakan bagian dari deretan perhitungannya. Cara untuk mengerjakan lompatan tersebut dapat diperhatikan pada Gambar di bawah ini.



Gambar : Pencacah biner (3 bit) tak sinkron modulo 6.

Jika gerbang NAND diabaikan maka pencacah tersebut merupakan pencacah biner (3 bit) tak sinkron modulo 8 yang akan mencacah dari 000 s/d 111. Jika gerbang NAND pada gambar 10.4 diperhatikan dapat dijelaskan sebagai berikut :

1. Keluaran NAND dihubungkan ke Clear (Cr) pada setiap FF. Pencacah tidak akan terpengaruh selama keluaran NAND pada keadaan tinggi (Cr setiap FF pada keadaan 1). Ketika keluaran NAND menuju rendah, maka akan meng-clear semua FF sehingga pencacah tersebut segera menuju ke keadaan 000.
2. Kedua masukan NAND masing-masing dihubungkan dengan keluaran FF-1 dan FF-2 sehingga keluaran NAND tersebut akan menuju rendah kapan saja $Q_1 = Q_2 = 0$. Keadaan ini akan terjadi ketika pencacah berubah dari keadaan 101 ke 110 (akhir detak ke 6). Keadaan rendah pada keluaran NAND akan segera meng-clear pencacah tersebut ke keadaan 000. Ketika FF-FF telah di-clear maka keluaran NAND kembali ke tinggi karena keadaan $Q_1 = Q_2 = 0$ tidak lama terjadi. Perhatikanlah Gambar berikut.



Gambar : Bentuk gelombang pencacah biner tak sinkron modulo 6

Bentuk gelombang keluaran pada Q_1 terdapat kenaikan dan gelinciran yang sangat tajam oleh kejadian sesaat dari keadaan 110 sebelum peng-clear-an. Kenaikan dan gelinciran itu sangat sempit sehingga tidak menghasilkan petunjuk yang terlihat pada tampilan. Kenaikan dan gelinciran tersebut dapat menimbulkan persoalan hanya jika keluaran Q_1 digunakan untuk mengendalikan rangkaian lain.

3. Walaupun pencacah menuju ke keadaan 110 tetapi hanya untuk beberapa nanodetik sebelum ke keadaan 000. Sehingga pada dasarnya dapat dikatakan bahwa pencacah tersebut menghitung dari 000 (nol) ke 101 (lima) dan selanjutnya kembali ke 000. Keadaan 110 dan 111 dilompati sehingga pencacah tersebut hanya melalui 6 keadaan yang berbeda, dengan demikian pencacah tersebut merupakan pencacah modulo 6. Deretan pencacahannya dapat diperhatikan pada tabel berikut.

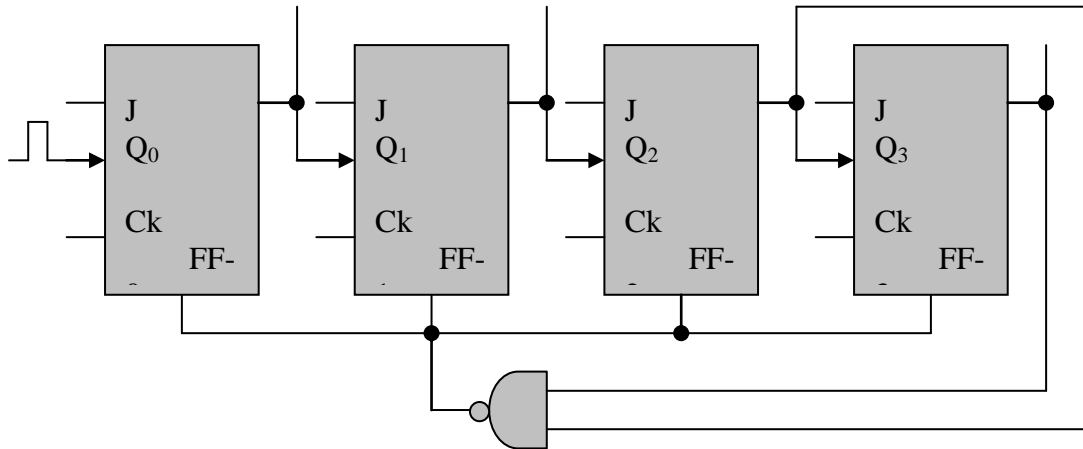
Q ₂	Q ₁	Q ₀
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0

Keadaan sementara 110 diperlukan untuk meng-clear pencacah

Cacah saluran masukan pada gerbang NAND yang digunakan untuk menghasilkan lompatan tidak harus 2 saluran tetapi tergantung dari banyaknya tingkat logika 1 yang digunakan untuk meng-clear pencacahan. Untuk membangun pencacah biner tak sinkron yang memulai penghitungan dari nol dan memiliki bilangan modulo x adalah :

1. Tentukan banyaknya FF terkecil (paling sedikit) sedemikian hingga $2^N \geq x$ dan hubungkan FF-FF tersebut menjadi pencacah tak sinkron.
2. Hubungkan keluaran NAND ke saluran Cr (Clear) setiap FF.
3. Tentukan keluran FF-FF mana yang akan berada pada keadaan tinggi (1) pada suatu hitungan sama dengan x , kemudian hubungkan keluaran FF-FF tersebut dengan saluran masukan gerbang NAND.

Sebagai contoh marilah menyusun pencacah biner tak sinkron modulo 10 yang akan mencacah dari 0000 (nol) hingga 1001 (sembilan). Karena ada 10 keadaan, dan 10 itu terletak di antara $2^3 = 8$ dan $2^4 = 16$, maka diperlukan 4 FF. Pencacah tersebut di-clear ke nol ketika keadaan 1010 (sepuluh) dicapai. Oleh karena itu keluaran FF-1 (Q₁) dan FF-3 (Q₃) harus dihubungkan ke masukan gerbang NAND. Rangkaian pencacah tersebut tampak pada Gambar berikut. Ingat bahwa setiap FF dikenai $J = K = 1$.



Gambar : Pencacah biner tak sinkron modulo 10.

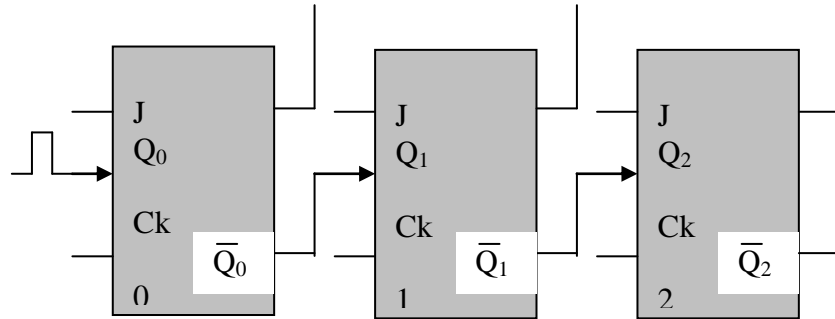
Pencacah modulo 10 pada Gambar di atas juga disebut sebagai pencacah desimal (dekade). Pencacah dekade tersebut melakukan pencacahan dari 0000 (nol) hingga 1001 (sembilan) sehingga pencacah itu juga disebut sebagai pencacah BCD (Binary Coded Decimal). Pencacah BCD menggunakan 10 kode (sandi) kelompok BCD yaitu 0000, 0001, 0010, ... 1000, dan 1001. Pencacah dekade sering digunakan sebagai pembagi frekuensi secara tepat dengan 10.

Semua pencacah yang telah dikemukakan sebelumnya merupakan pencacah naik, karena telah menghitung dari nol hingga bilangan tertinggi yang dapat dicapai. Selain itu dapat pula disusun pencacah turun yang akan mencacah dari cacahan tertinggi hingga nol. Untuk itu perhatikan ilustrasi pencacah turun berikut.

	Q ₂	Q ₁	Q ₀		Q ₂	Q ₁	Q ₀		Q ₂	Q ₁	Q ₀
(7)	1	1	1		1	1	1		1	1	1
(6)	1	1	0		1	1	0		1	1	0
(5)	1	0	1		1	0	1		1	0	1
(4)	1	0	0		1	0	0		1	0	0
(3)	0	1	1		0	1	1		0	1	1
(2)	0	1	0		0	1	0		0	1	0
(1)	0	0	1		0	0	1		0	0	1
(0)	0	0	0		0	0	0		0	0	0

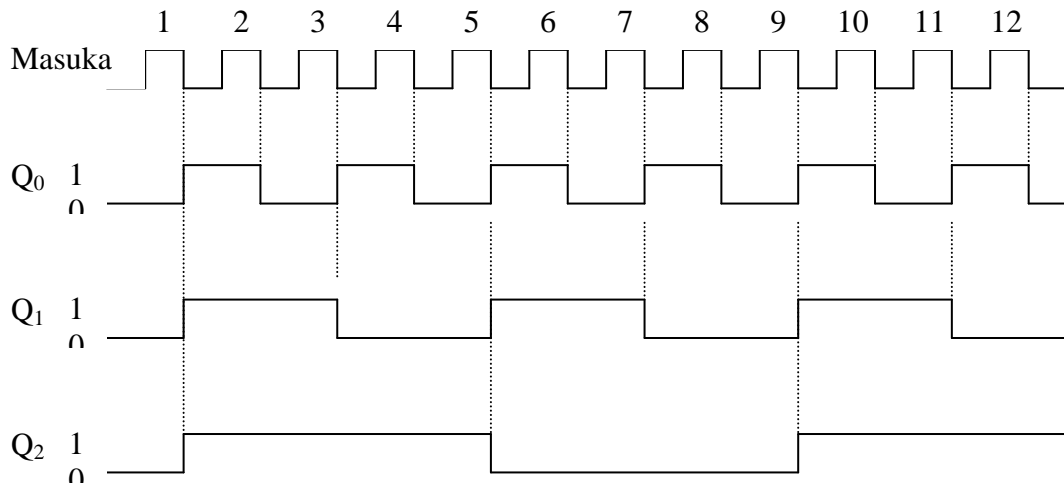
Dst.

Terlihat bahwa pada pencacah turun, setiap FF kecuali yang pertama (FF-0) harus berubah keadaan (ter-toggle) ketika keluaran FF sebelumnya mengalami transisi dari rendah ke tinggi (transisi positif). Jika FF-FF tersebut dengan Ck yang merespon transisi negatif (dari tinggi ke rendah), maka di depan Ck pada setiap FF dapat dipasang sebuah inverter (NOT). Cara lain dapat dikerjakan dengan mengendalikan setiap masukan Ck melalui keluaran komplemen (\bar{Q}) dari FF sebelumnya. Untuk lebih jelasnya, perhatikan Gambar berikut.



Gambar : Pencacah biner tak sinkron turun.

Detak masukan tetap dikenakan pada Ck dari FF-0, keluaran komplemen FF-0 (\bar{Q}_0) dikenakan pada Ck untuk FF-1, dan keluaran komplemen FF-1 (\bar{Q}_1) dikenakan pada Ck untuk FF-2. Keluaran penacahan masih tetap diambilkan dari keluaran $Q_2Q_1Q_0$. Bentuk gelombang pada $Q_2Q_1Q_0$ dapat dilihat pada Gambar di bawah ini.



Gambar : Bentuk gelombang keluaran pencacah turun.

Bentuk gelombang pada Gambar tersebut menunjukkan bahwa Q_1 (keluaran FF-1) berubah keadaan ketika Q_0 (keluaran FF-0) berubah dari rendah ke tinggi (\bar{Q}_0 berubah dari tinggi ke rendah), dan Q_2 berubah keadaan ketika Q_1 berubah dari rendah ke tinggi (\bar{Q}_1 berubah dari tinggi ke rendah). Pencacah turun tidak banyak digunakan sebagaimana pencacah naik. Pencacah turun banyak digunakan dalam keadaan di mana telah diketahui suatu jumlah yang diinginkan. Dalam keadaan ini pencacah turun di-preset pada jumlah yang diinginkan dan selanjutnya diijinkan mencacah turun berdasarkan pulsa yang dikenakan. Ketika pencacahan mencapai nol diperlukan rangkaian gerbang logika untuk menunjukkan bahwa sejumlah pulsa yang telah di-preset telah terjadi.

Pencacah Biner Sinkron (Paralel)

Pada pencacah biner tak sinkron setiap FF tidak berubah secara bersamaan (tidak terjadi sinkronisasi terhadap detak masukan), tetapi perubahannya terjadi secara serial pada mana perubahan keadaan suatu FF menunggu pemicuan dari FF sebelumnya. Hal ini menyebabkan terjadinya akumulasi waktu tunda. Keterbatasan tersebut dapat diatasi dengan menggunakan pencacah sinkron (paralel) pada mana semua FF dipicu secara bersamaan (paralel) oleh detak masukan. Karena detak masukan dikenakan pada setiap FF maka digunakan beberapa gerbang dan cara untuk mengendalikan kapan suatu FF berubah keadaan dan kapan FF itu tetap oleh adanya pengaruh detak masukan. Untuk keperluan pengendalian itu dikerjakan melalui saluran masukan FF, misal J dan K. Dengan demikian, dalam merancang pencacah sinkron perlu menentukan agar J dan K setiap FF sebagai fungsi keluaran. Untuk mengurangi banyaknya fungsi biasanya digunakan FF-D dan FF-T, karena kedua FF tersebut memiliki satu saluran masukan. Jika dibandingkan dengan pencacah tak sinkron, maka pada pencacah sinkron berlaku :

- a. Saluran C_k untuk semua FF dihubungkan bersama sehingga detak masukan dikenakan pada setiap FF secara bersamaan (simultan).
- b. Hanya FF paling depan, yang berkedudukan sebagai LSB, yang dapat dikenai $J = K = 1$ secara permanen. Saluran J dan K pada FF yang lain dikendalikan melalui kombinasi keluaran FF-FF yang sesuai.

- c. Pencacah sinkron memerlukan lebih banyak persambungan (rangkaian) bila dibandingkan pencacah tak sinkron yang setara.
- d. Pencacah sinkron lebih cepat karena dapat menghitung frekuensi detak yang lebih tinggi untuk selang waktu yang sama.

Untuk merancang pencacah sinkron dengan suatu jenis FF perlu mengetahui tabel transisi siklus pencacahannya dan tabel eksitasi dari FF yang bersangkutan. Selanjutnya, dari kedua tabel itu ditampung dalam suatu tabel kebenaran untuk menentukan fungsi saluran masukannya terhadap keluarannya. Fungsi tersebut dapat ditentukan dengan bantuan peta Karnough. Untuk lebih jelasnya, berikut ini dikemukakan cara merancang suatu pencacah biner sinkron modulo 5 yang menggunakan FF-JK. Untuk pencacah tersebut jelas diperlukan 3 buah FF.

Tabel eksitasi untuk FF-JK (telah dibahas dalam bab Flip-flop) adalah sebagai berikut :

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

Tabel transisi untuk pencacah biner sinkron modulo 5 adalah sebagai berikut :

Q_2	Q_1	Q_0	Pencacahan		
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	-	-	-
1	1	0	-	-	-
1	1	1	-	-	-

Bilangan biner tertinggi hasil pencacahannya adalah 100. Dengan demikian keadaan atau angka biner 101, 110, dan 111 tidak boleh terjadi. Keluaran pencacah tersebut adalah

$Q_2Q_1Q_0$ di mana Q_2 merupakan MSB dan Q_0 sebagai LSB. Masukan Ck setiap FF dihubungkan dengan detak masukan. Fungsi saluran masukan (J dan K) terhadap saluran keluaran (Q) setiap FF tertuang dalam tabel kebenaran seperti berikut :

No. baris	Q_2	Q_1	Q_0	J_2	K_2	J_1	K_1	J_0	A.
0	0	0	0	0	X	0	X	1	X
1	0	0	1	0	X	1	X	X	1
2	0	1	0	0	X	X	0	1	X
3	0	1	1	1	X	X	1	X	1
4	1	0	0	X	1	0	X	1	X
5	0	0	0	X	X	X	X	X	1
6	0	0	1	X	X	X	X	1	X
7	0	1	0	X	X	X	X	X	1

Untuk menentukan J_2 , K_2 , J_1 , K_1 , J_0 dan K_0 sebagai fungsi dari variabel keluaran (Q_2 , Q_1 , dan Q_0) perlu dicermati satu per satu berdasarkan tabel eksitasi FF-JK. Misalkan untuk $Q_2Q_1Q_0 = 000$, oleh karena setelah terjadi detak masukan keadaannya harus berubah menjadi $Q_2Q_1Q_0 = 001$, ini berarti terjadi perubahan pada Q_0 dari 0 menjadi 0, dan berdasarkan tabel eksitasi haruslah $J_2 = 0$ dan $K_2 = X$. Nilai J dan K tersebut berlaku untuk $Q_2Q_1Q_0$ dari 001 dan 010. Sedangkan untuk $Q_2Q_1Q_0 = 011$ pada mana Q_2 berubah dari 0 menjadi 1, maka haruslah $J_2 = 1$ dan $K_2 = X$. Dengan cara yang sama dapat diteruskan untuk transisi yang lain. Untuk $Q_2Q_1Q_0 = 100$ pada detak berikutnya harus berubah menjadi 000 (kembali ke keadaan awal pencacahan), maka $J_2 = X$ dan $K_2 = 1$ karena Q_2 mengalami perubahan dari 1 ke 0. Selanjutnya, keluaran pencacah tidak diijinkan untuk menuju ke keadaan 101, 110, dan 111, maka untuk keadaan-keadaan tersebut harga J_2 dan K_2 boleh diisi sembarang, atau $J_2 = X$ dan $K_2 = X$. Cara tersebut harus diteruskan untuk nilai-nilai J_1 , K_1 , J_0 dan K_0 . Jika semua telah terisi, untuk mendapatkan fungsi keadaan masukan terhadap keluarannya akan dituangkan dalam peta Karnough seperti berikut. Oleh karena ada 6 masukan (3 FF), yaitu J_2 , K_2 , J_1 , K_1 , J_0 dan K_0 , maka akan diperoleh 6 peta Karnough. Tetapi peta Karnough untuk masukan LSB, yaitu J_0 dan K_0 , tidak perlu dibuat karena pada umumnya berlaku $J_0 = K_0 = 1$.

Q_2Q_1	$\bar{Q}_2\bar{Q}_1$	\bar{Q}_2Q_1	Q_2Q_1	$Q_2\bar{Q}_1$
Q_0				
\bar{Q}_0	0	0	X	X
Q_0	0	1	X	X

$$J_2 = Q_1Q_0$$

Q_2Q_1	$\bar{Q}_2\bar{Q}_1$	\bar{Q}_2Q_1	Q_2Q_1	$Q_2\bar{Q}_1$
Q_0				
\bar{Q}_0	X	X	X	1
Q_0	X	X	X	X

$$K_2 = 1$$

Q_2Q_1	$\bar{Q}_2\bar{Q}_1$	\bar{Q}_2Q_1	Q_2Q_1	$Q_2\bar{Q}_1$
Q_0				
\bar{Q}_0	0	X	X	0
Q_0	1	X	X	X

$$J_1 = Q_0$$

Q_2Q_1	$\bar{Q}_2\bar{Q}_1$	\bar{Q}_2Q_1	Q_2Q_1	$Q_2\bar{Q}_1$
Q_0				
\bar{Q}_0	X	0	X	X
Q_0	X	1	X	X

$$K_1 = Q_0$$

Q_2Q_1	$\bar{Q}_2\bar{Q}_1$	\bar{Q}_2Q_1	Q_2Q_1	$Q_2\bar{Q}_1$
Q_0				
\bar{Q}_0	1	1	X	1
Q_0	X	X	X	X

$$J_0 = 1$$

	Q_2Q_1	$\bar{Q}_2\bar{Q}_1$	\bar{Q}_2Q_1	Q_2Q_1	$Q_2\bar{Q}_1$
Q_0					
\bar{Q}_0	X	X	X	X	X
Q_0	1	1	X	X	X

$$K_0 = 1$$

Dari semua peta Karnaugh tersebut di atas, maka dapat diperoleh suatu sistem persamaan (tidak unik) sebagai berikut :

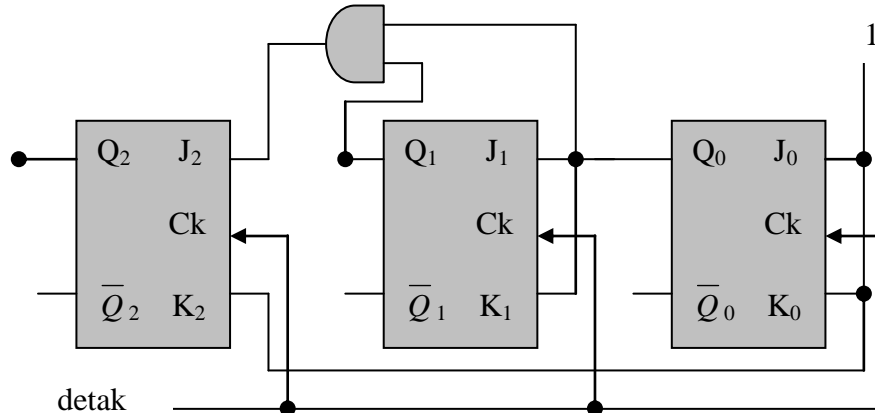
$$J_0 = K_0 = 1$$

$$J_1 = K_1 = Q_0$$

$$J_2 = Q_1Q_0 \text{ dan}$$

$$K_2 = 1.$$

Atas dasar persamaan tersebut, maka diagram rangkaian pencacah biner sinkron modulo 5 tampak pada Gambar berikut seperti berikut :

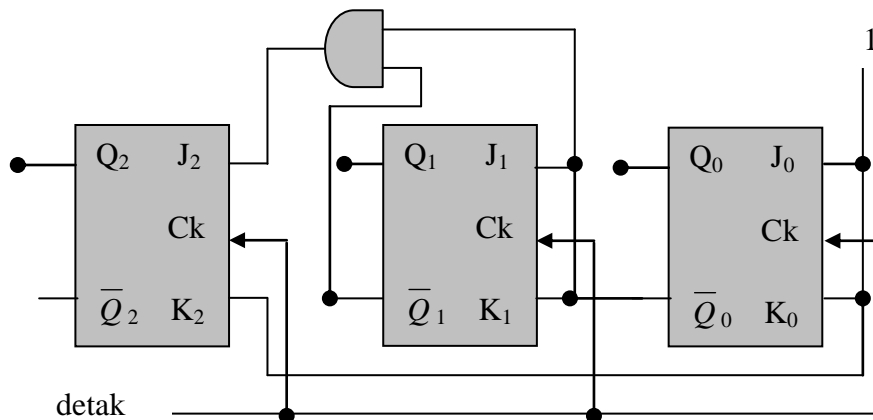


Gambar : Pencacah biner sinkron modulo 5 dengan FF-JK.

Semua FF yang berdetak dapat digunakan sebagai komponen rangkaian pencacah biner. Tetapi kebanyakan FF yang digunakan adalah jenis FF-T dan FF-D, oleh karena kedua jenis FF tersebut masing-masing hanya mempunyai satu saluran masukan. Dengan demikian persambungan rangkaian pencacah yang dibuat menjadi lebih sederhana. Perlu diingat

bahwa tabel eksitasi setiap jenis FF tidaklah sama. Sehingga dalam membuat tabel kebenaran harus mengacu pada tabel eksitasi dari FF yang dipilih.

Pencacah paralel yang telah dikemukakan merupakan pencacah naik, yakni mencacah dari bilangan kecil ke bilangan yang semakin besar untuk setiap siklus pencacahan. Ada kalanya juga diperlukan pencacah turun, yakni mencacah dari bilangan yang besar ke bilangan yang lebih kecil untuk setiap siklus pencacahan. Sebagaimana pada pencacah serial, pencacah paralel turun dapat disusun dengan memanfaatkan keluaran komplemen dari FF (\bar{Q}) untuk memberikan masukan kepada gerbang logika yang lain (biasanya gerbang AND). Sebagai ilustrasi, pencacah paralel naik pada Gambar 10.9 dapat diubah menjadi pencacah paralel turun dengan menghubungkan keluaran \bar{Q}_1 dan \bar{Q}_0 ke masukan gerbang AND berturut-turut yang ditempati oleh Q_1 dan Q_0 . Keluaran hasil pencacahan tetap diambilkan dari $Q_2Q_1Q_0$. Perhatikan Gambar berikut.

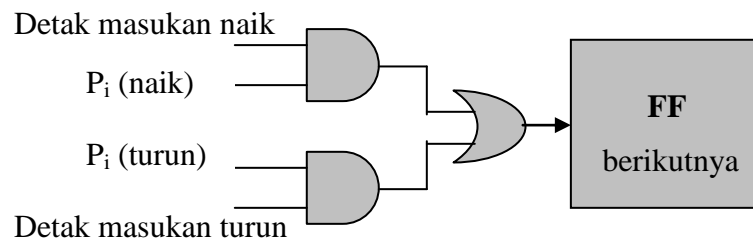


Gambar : Pencacah biner sinkron turun modulo 5 dengan FF-JK.

Pencacah seperti Gambar di atas itu akan mencacah mengikuti deretan berikut :

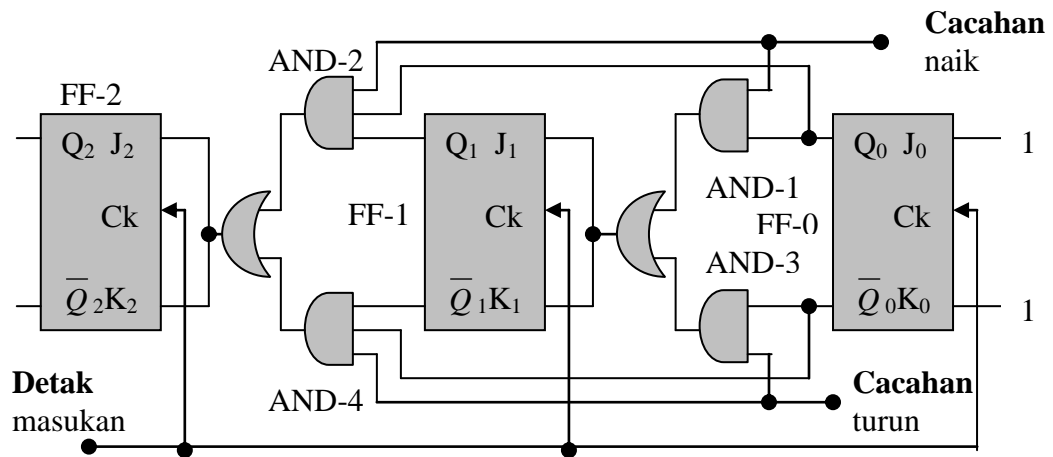
\bar{Q}_2	\bar{Q}_1	\bar{Q}_0
1	1	1
1	1	0
1	0	1
1	0	0
0	1	1

Untuk tujuan fleksibilitas dapat dibangun pencacah naik/turun, artinya satu rangkaian pencacah tersebut dapat difungsikan sebagai pencacah naik atau pencacah turun tergantung kebutuhan. Salah satu cara untuk membuat fungsi tersebut adalah mengendalikannya dengan suatu rangkaian pengendali pada masukan. Rangkaian tersebut digunakan untuk mengendalikan apakah keluaran normal FF (Q) atau keluaran komplemennya (\bar{Q}) diumpankan ke masukan J dan K pada FF berikutnya. Rangkaian pengendali untuk mengatur pencacah naik atau turun itu tampak pada Gambar berikut.



Gambar : Rangkaian pengendali pencacah naik/turun.

Dari gambar tersebut itu tampak adanya dua jalur masukan, yaitu masukan detak untuk pencacah naik dan masukan detak untuk pencacah turun. Dengan demikian pada masukan C_k dari setiap FF, kecuali FF yang pertama (LSB), perlu ditambahkan gerbang AND dan OR dalam konfigurasi seperti gambar 10.11. Untuk lebih jelasnya perhatikan contoh rangkaian pencacah naik/turun 3 bit (modulo 8) pada Gambar di bawah ini. Pencacah itu akan mencacah dari 000 naik menuju 111 ketika masukan pengendali Cacahan-naik bernilai 1 dan mencacah dari 111 turun menuju 000 ketika masukan pengendali Cacahan-turun berharga 1. Selama saluran Cacahan-naik dalam keadaan 1 dan saluran Cacahan-turun pada keadaan 0 maka AND-1 dan AND-2 yang aktif menyalurkan detak sedangkan AND-3 dan AND-4 tidak bekerja. Hal ini memungkinkan keluaran Q_0 dan Q_1 terus menuju ke masukan J dan K pada FF berikutnya sehingga pencacah itu akan mencacah naik terhadap detak masukan. Hal yang sebaliknya terjadi ketika saluran Cacahan-naik = 0 dan Cacahan-turun = 1.



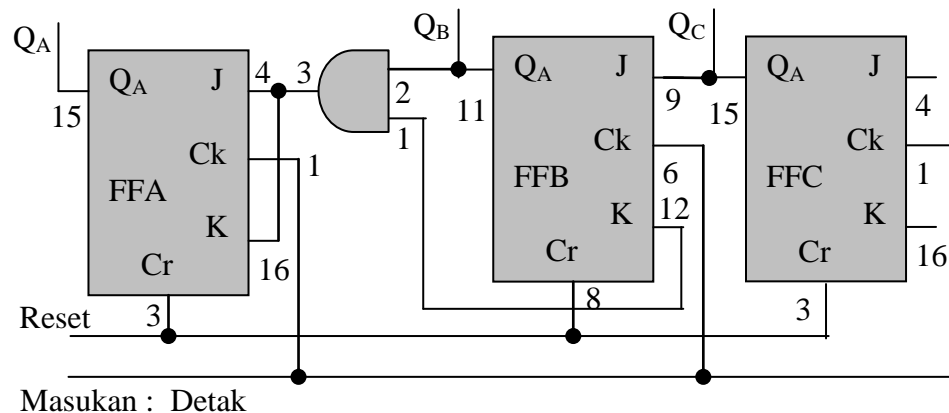
Gambar : Pencacah paralel naik/turun modulo 8.

Langkah-langkah Percobaan :

Pencacah Biner Sinkron

Wujudkanlah rangkaian berikut pada papan rangkaian yang telah tersedia (matikan dahulu sumber dayanya) menggunakan :

Dua buah IC-7476, IC₁ digunakan untuk FFA dan FFB, dan setengah dari IC₂ digunakan sebagai FFC (ingat IC-7476 terdiri dari dua JKFF). Angka-angka pada gambar menunjukkan nomor pin dari IC. Untuk IC-7476 pin 5 adalah Vcc (+5 volt dc) dan pin 13 adalah GDN (tanah). Sebuah IC-7408 yang digunakan sebagai gerbang AND. Pin 14 dihubungkan ke Vcc (+5 volt dc) dan pin 7 ke GND.

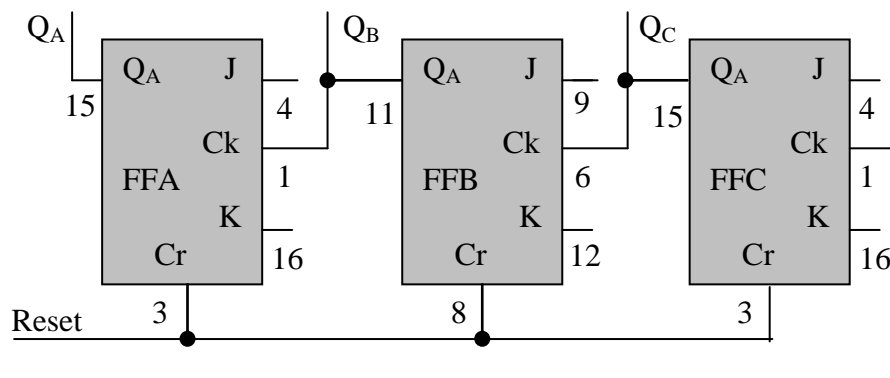


Hubungkan Reset dan Input pada generator bit (saklar yang keluarannya dapat diatur 0 atau 1), sedangkan Q_A , Q_B , Q_C pada indikator LED. Aturlah mula-mula Reset = 1, clock = 1 dan kemudian hidupkan sumber daya. Resetlah pencacah dengan mengubah input reset sebentar ke 0 sehingga semua input 0. Masukkanlah pulsa input sebanyak 8 kali dan catat keadaan dari Q_A Q_B Q_C pada setiap pulsa input masuk. Satu pulsa input diperoleh dengan mengubah input ke 1 sebentar kemudian 0 lagi. Lakukan percobaan ini sebanyak 2 atau 3 kali dan hasil dari masing-masing percobaan masukkan ke tabel berikut.

Pulsa Masukan ke	Q_A	Q_B	Q_C
0			
1			
2			
3			
4			
5			
6			
7			

Pencacah Biner Tak Sinkron

Wujudkanlah rangkaian berikut pada papan rangkaian yang telah tersedia (matikan dahulu sumber dayanya) menggunakan dua buah IC-7476, IC₁ digunakan untuk FFA dan FFB, dan setengah dari IC₂ digunakan sebagai FFC (ingat IC-7476 terdiri dari dua JKFF). Angka-angka pada gambar menunjukkan nomor pin dari IC. Untuk IC-7476 pin 5 adalah Vcc (+5 volt dc) dan pin 13 adalah GDN (tanah).



Masukan : Detak

Hubungkan Reset dan Input pada generator bit (saklar yang keluarannya dapat diatur 0 atau 1), sedangkan Q_A , Q_B , Q_C pada indikator LED. Aturilah mula-mula Reset = 1, clock = 0 dan kemudian hidupkan sumber daya. Resetlah pencacah dengan mengubah input reset sebentar ke 0 sehingga semua input 0. Masukkanlah pulsa input sebanyak 8 kali dan catat keadaan dari Q_A Q_B Q_C pada setiap pulsa input masuk. Satu pulsa input diperoleh dengan mengubah input ke 1 sebentar kemudian 0 lagi. Lakukan percobaan ini sebanyak 2 atau 3 kali dan hasil dari masing-masing percobaan masukkan ke tabel berikut.

Pulsa Masukan ke	Q_A	Q_B	Q_C
0			
1			
2			
3			
4			
5			
6			
7			