

Bab IV

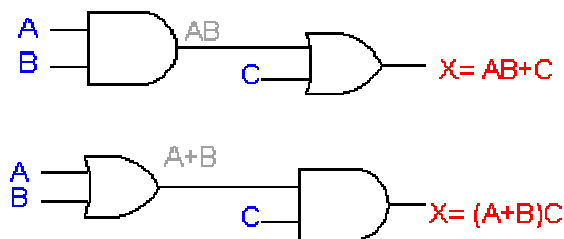
ALJABAR BOOLEAN

- 4.1. Describing Logic Circuits Algebraically**
- 4.2. Evaluating Logic Circuit Outputs**
- 4.3. Implementing Circuits from Boolean Expression**
- 4.4. Boolean Theorems**
- 4.5. DeMorgan's Theorems**
- 4.6. Universality of NAND and NOR Gates**
- 4.7. Alternate Logic-Gate Representations**
- 4.8. Logic Symbol Interpretation**

4.1. Menyatakan Rangkaian Logika Secara Aljabar Boolean

(Describing Logic Circuits Algebraically)

Sebuah rangkaian logika walau serumit apapun dapat dinyatakan ke dalam suatu persamaan yang menyatakan operasi boolean. Hal ini dikarenakan pada prinsipnya sebuah gerbang logika (AND, OR dan NOT) merupakan suatu operasi boolean tingkat dasar. Seperti contoh berikut ini sebuah rangkaian logika yang dinyatakan ke dalam persamaan operasi boolean.

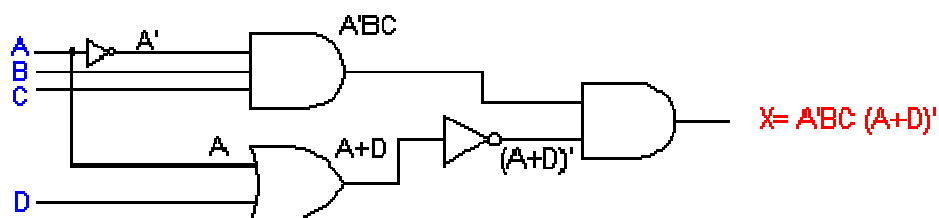


Gambar 4.1. Rangkaian gerbang dan ekspresi aljabar booleannya

Pada ekspresi operasi boolean memuat kedua operasi dasar yaitu AND dan OR, operasi AND dilakukan pertama kali ($X = AB + C$ dimana AB diproses terlebih dulu). Hal ini prioritas perkalian lebih tinggi daripada penjumlahan. Untuk contoh yang kedua ($A+B$) diproses terlebih dulu ($X = (A+B) + C$)

Rangkaian Yang Mengandung Gerbang NOT/Inverter

Apabila sebuah gerbang NOT/inverter terdapat pada rangkaian logika, maka output gerbang ini akan bernilai sama dengan kebalikan inputnya, yang biasanya dinyatakan dengan tanda (').



Gambar 4.2. Rangkaian dengan gerbang NOT dan ekspresi aljabar booleannya

4.2. Analisa Keluaran Rangkaian Logika (*Evaluating Logic Circuit Outputs*)

Apabila persamaan operasi boolean sudah dapat ditentukan, maka logika keluaran dapat ditentukan dengan mudah untuk setiap pasangan masukan yang diberikan. Berikut ini adalah dua buah contoh sederhana bagaimana kita melakukan analisa penentuan logika keluaran sebuah rangkaian logika.

Misalkan masukan yang diberikan untuk sebuah operasi boolean $X = A'BC(A+D)'$ adalah : Let $A = 0, B = 1, C = 1, D = 1$

$$\begin{aligned} X &= A'BC(A+D)' \\ &= 0' * 1 * 1 * (0+1)' \\ &= 1 * 1 * 1 * (1)' \\ &= 1 * 1 * 1 * 0 \\ &= 0 \end{aligned}$$

Kemudian sebagai contoh berikutnya adalah misalkan masukan untuk sebuah operasi boolean $X = [D + ((A+B)C)'] * E$ adalah $A = 0, B = 0, C = 1, D = 1, E = 1$

$$\begin{aligned} X &= [D + ((A+B)C)'] * E \\ &= [1 + ((0+0)1)'] * 1 \\ &= [1 + (0*1)'] * 1 \\ &= [1 + 0'] * 1 \\ &= [1 + 1] * 1 \\ &= 1 \end{aligned}$$

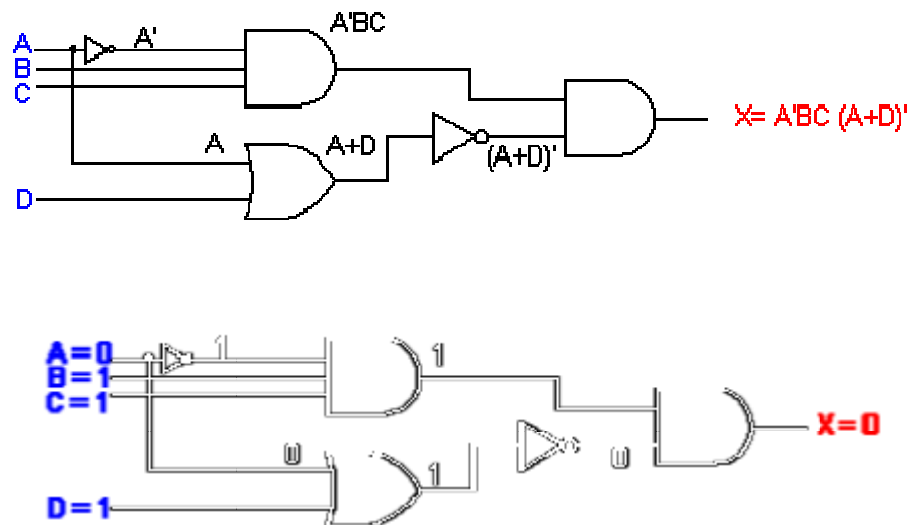
Secara umum, tahapan berikut ini harus dilakukan apabila diinginkan untuk analisa penentuan keluaran rangkaian logika yang telah dinyatakan ke dalam persamaan boolean, yaitu :

1. Pertama, untuk semua pembalikan (Inversion) lakukan dengan merubah nilai logik input dengan kebalikannya.
2. Kemudian, proseslah semua ekspresi boolean yang terdapat dalam kurung kurawal, karena memiliki prioritas lebih tinggi.

3. Kemudian, proseslah operasi AND sebelum OR, kecuali ada tanda kurung yang menyatakan prioritas.
4. Jika terdapat tanda *bar* ($'$) diatas sebuah ekspresi, maka proseslah terlebih dahulu ekspresinya kemudian lakukan pembalikan terhadap hasilnya.

Menentukan Logika Keluaran Dari Sebuah Diagram Gerbang Logika (*Determining Output Level from a Diagram*)

Untuk menentukan logika keluaran sebuah diagram logika untuk setiap masukan yang diberikan juga bisa dilakukan secara langsung dengan mengacu diagram rangkaian gerbang logika, tanpa mencari persamaan operasi boolean terlebih dahulu. Sebagai contoh berikut ini diberikan diagram gerbang logika dan akan ditentukan logika keluarannya.



Gambar 4.3. Penentuan ekspresi aljabar terhadap rangkaian gerbang logika

4.3. Implementasi Rangkaian Gerbang Logika terhadap Ekpresi Boolean (*Implementing Circuits From Boolean Expression*)

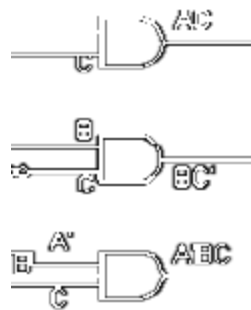
Operasi sebuah rangkaian digital didefinisikan dengan ekspresi/persamaan boolean, selanjutnya berdasar ekspresi boolean tersebut dapat ditentukan rangkaian gerbang logikanya. Misalkan diinginkan untuk membuat rangkaian gerbang logika yang memiliki ekspresi/persamaan keluaran $y = AC + BC' + A'BC$. Ekspresi boolean tersebut memiliki 3 bagian yaitu AC, BC' dan A'BC kemudian

ketiga bagian tersebut digabungkan dengan operasi OR. Untuk itu diperlukan gerbang OR dengan 3 masukan dengan masing-masing masukan yang diberikan adalah AC, BC' dan A'BC.



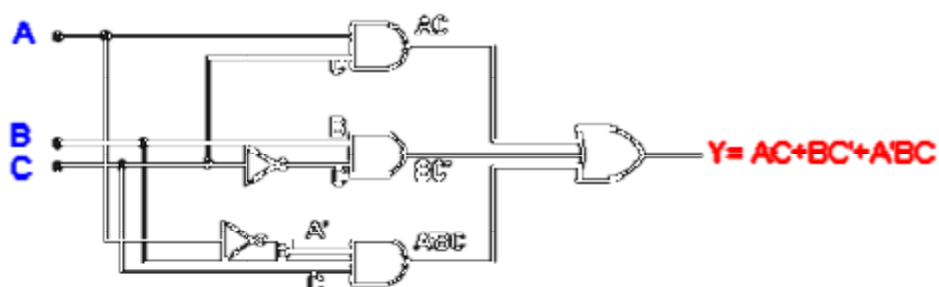
Gambar 4.4. Penentuan ekspresi aljabar terhadap gerbang logika

Tiap masukan dari gerbang OR tersebut adalah hasil keluaran dari gerbang AND. Diperlukan 3 buah gerbang AND untuk semua masukan gerbang OR tersebut.



Gambar 4.5. Penentuan ekspresi aljabar terhadap gerbang logika

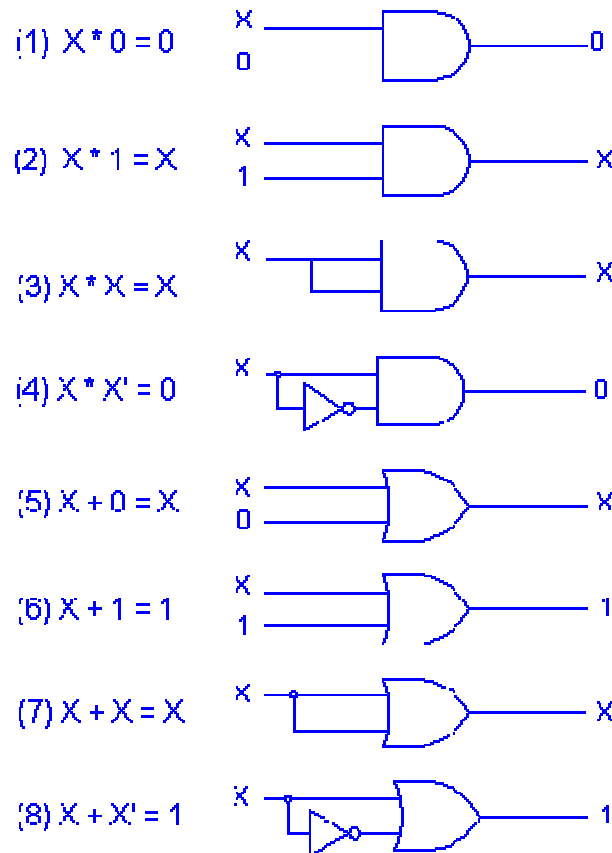
Selanjutnya penggunaan gerbang inverter adalah untuk menghasilkan A' dan C'. Secara keseluruhan rangkaian yang dimaksud adalah seperti gambar berikut ini.



Gambar 4.6. Penentuan ekspresi aljabar terhadap gerbang logika dengan gerbang NOT

4.4. Teorema Boolean (*Boolean Theorems*)

Dengan mempelajari berbagai teorema boolean yang ada akan sangat membantu kita dalam penyederhanaan ekspresi/persamaan boolean dan penyederhanaan rangkaian gerbang logika.



Teorema Multivariabel (*Multivariable Theorems*)

(9) $x + y = y + x$ (*commutative law*)

(10) $x * y = y * x$ (*commutative law*)

(11) $x + (y + z) = (x + y) + z = x + y + z$ (*associative law*)

(12) $x (yz) = (xy) z = xyz$ (*associative law*)

(13a) $x (y + z) = xy + xz$

(13b) $(w + x)(y + z) = wy + xy + wz + xz$

(14) $x + xy = x$

(15) $x + x'y = x + y$

Pembuktian teorema no 14 :

$$\begin{aligned}x + xy &= x(1+y) \\ &= x * 1 \text{ [using theorem (6)]} \\ &= x \text{ [using theorem (2)]}\end{aligned}$$

4.5. Teorema DeMorgans (DeMorgan's Theorem)

Teorema DeMorgans sangat berguna dalam penyederhanaan ekspresi/persamaan boolean dimana didalamnya mengandung proses pembalikan dari perkalian atau penjumlahan variabel.

$$(16) \quad (x+y)' = x' * y'$$

$$(17) \quad (x*y)' = x' + y'$$

Dari teorema 16 menyatakan bahwa operasi pembalikan terhadap hasil operasi OR terhadap 2 variabel, akan menghasilkan keluaran yang sama dengan pembalikan terhadap masing-masing variabel kemudian dilakukan operasi AND.

Dari teorema 17 menyatakan bahwa operasi pembalikan terhadap hasil operasi AND 2 variabel, hal ini akan menghasilkan keluaran yang sama dengan operasi pembalikan terhadap masing-masing variabel kemudian dilanjutkan dengan operasi OR.

Contoh :

$$\begin{aligned}X &= [(A'+C) * (B+D)]' \\ &= (A'+C)' + (B+D)' \quad (\text{teorema 17}) \\ &= (A''*C') + (B'+D'') \quad (\text{by theorem 16}) \\ &= AC' + B'D\end{aligned}$$

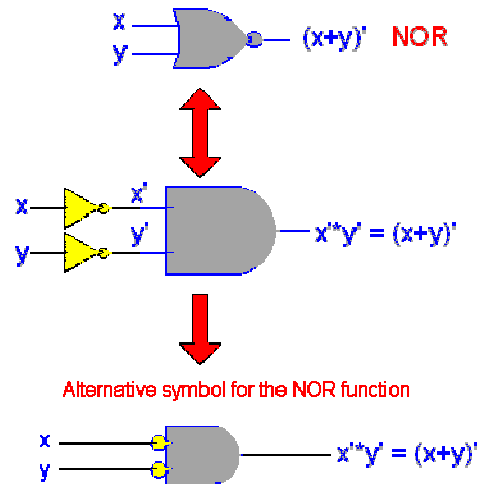
4.5.1. Teorema DeMorgan 3 Variabel (*Three Variables DeMorgan's Theorem*)

$$(18) \quad (x+y+z)' = x' * y' * z'$$

$$(19) \quad (xyz)' = x' + y' + z'$$

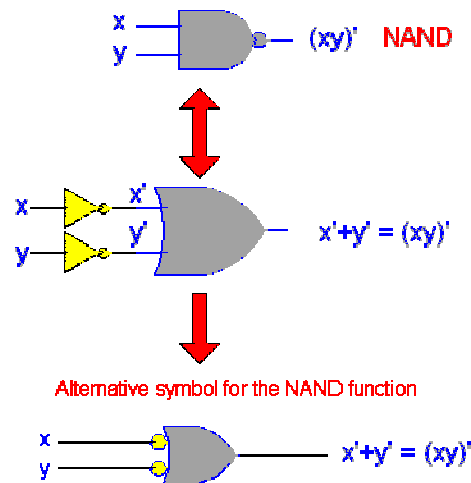
4.5.2. Teorema Implikasi DeMorgans (*Implications of DeMorgan's Theorem*)

Dari teorema (16) disebutkan bahwa : $(x+y)' = x' \cdot y'$



Gambar 4.7. Diagram Implikasi De Morgan $(x + y)' = x' \cdot y'$

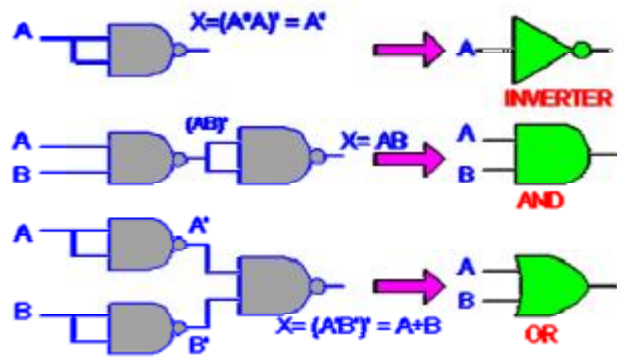
Dari teorema 17 disebutkan bahwa : $(x \cdot y)' = x' + y'$



Gambar 4.8. Diagram Implikasi De Morgan $(x \cdot y)' = x' + y'$

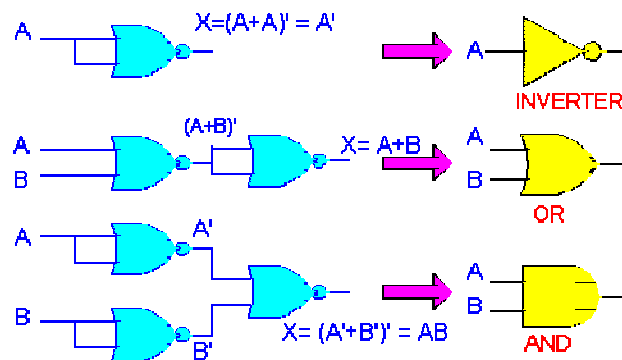
4.6. Penyeragaman Gerbang NAND-NOR (Universality of NAND & NOR Gates)

Suatu ekspresi boolean tertentu bisa diimplementasikan dengan gerbang NAND tanpa menggunakan gerbang yang lain. Hal ini dilakukan dengan tujuan untuk keseragaman gerbang dalam rangkaian.



Gambar 4.9. Ekuivalensi gerbang logika dengan gerbang NAND

Dengan cara yang sama, dapat dibuktikan bahwa gerbang NOR juga bisa diterapkan untuk menyatakan ekspresi boolean tanpa menggunakan gerbang yang lain.



Gambar 4.10. Ekuivalensi gerbang logika dengan gerbang NOR

4.7. Kesamaan Representasi Gerbang Logika

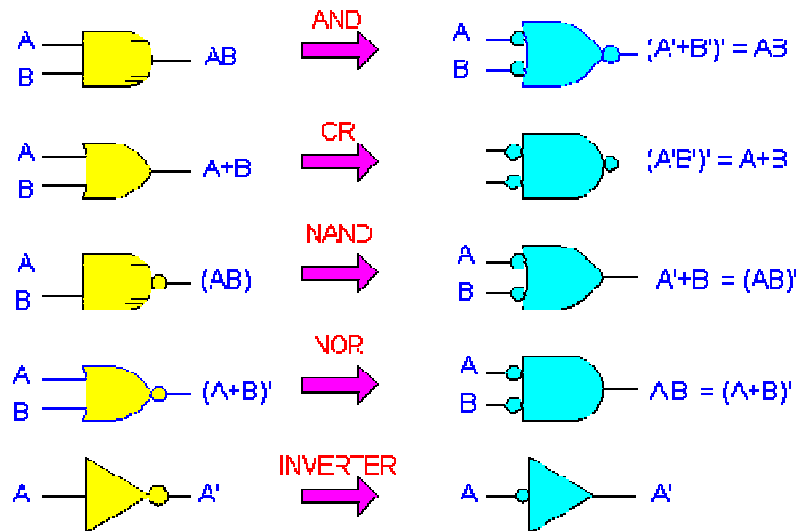
(Alternate Logic Gate Representations)

Pada diagram berikut ini disampaikan kesamaan representasi gerbang logika sebagai alternatif dalam menyatakan ekspresi boolean. Yang disisi sebelah kiri adalah diagram aslinya sedangkan yang disebelah kanan adalah alternatif lain sebagai kesamaannya. Diagram alternatif yang ada didapatkan dengan melakukan manipulasi sesuai dengan 2 langkah berikut ini :

1. Balik nilai logika setiap masukan dan keluaran yang ada pada diagram aslinya. Hal ini dilakukan dengan menambahkan lingkaran kecil (tanda pembalikan nilai logika) pada masukan dan keluaran yang sebelumnya

tidak memiliki tanda, dan sebaliknya menghilangkan tanda jika sebelumnya telah ada tanda pembalikan tersebut.

2. Kemudian dilanjutkan dengan merubah setiap simbol gerbang AND menjadi gerbang OR atau sebaliknya, sedangkan untuk gerbang inverter tidak perlu dirubah simbolnya.



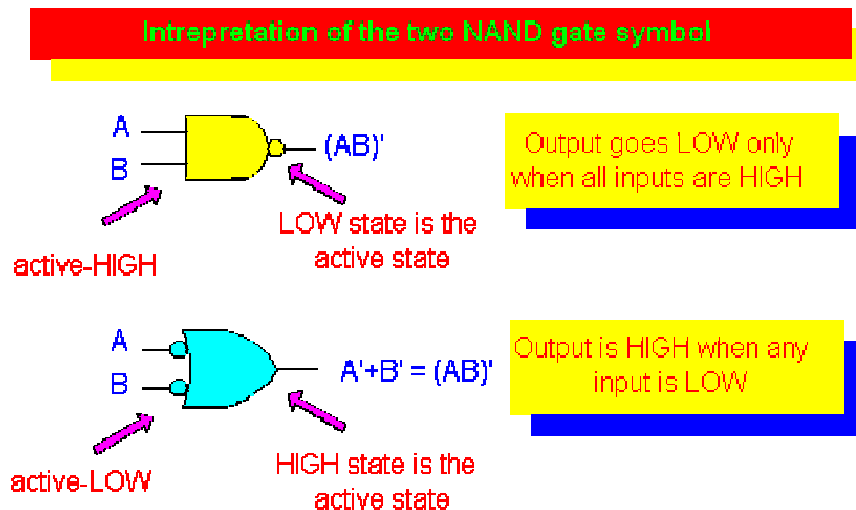
Gambar 4.11. Ekuivalensi gerbang logika NAND dan NOR

Beberapa catatan penting dalam melakukan perubahan suatu simbol dengan kesamaannya adalah sebagai berikut :

1. Kesamaan tersebut adalah benar untuk simbol dengan berbagai masukan.
2. Pada simbol aslinya (yang disebelah kiri) tidak satupun variabel masukan yang memiliki tanda pembalik (inverter), sedangkan pada simbol kesamaannya (yang disebelah kanan) semuanya memiliki tanda pembalik.
3. Baik simbol asli maupun kesamaannya keduanya sama-sama merepresentasikan rangkaian yang sama.
4. Gerbang NAND dan NOR keduanya adalah gerbang yang memiliki pembalik, sehingga simbol asli maupun kesamaannya untuk kedua gerbang tersebut memiliki tanda pembalik pada sisi masukan atau keluarannya.
5. Gerbang AND dan OR keduanya adalah gerbang yang tidak memiliki tanda pembalik, sehingga simbol asli maupun kesamaannya untuk kedua gerbang tersebut memiliki tanda pembalik pada sisi masukan dan keluarannya.

4.8. Interpretasi Simbol Logika (*Logic Symbol Interpretation*)

Apabila saluran masukan atau keluaran sebuah rangkaian logika tidak memiliki tanda pembalik, maka saluran tersebut dikatakan sebagai aktif-HIGH. Sedangkan jika saluran masukan atau keluaran suatu rangkaian logika terdapat tanda pembalik, maka saluran tersebut dikatakan sebagai aktif-LOW. Ada atau tidaknya tanda pembalik tersebut pada suatu saluran akan menentukan apakah saluran tersebut aktif-HIGH atau aktif-LOW.

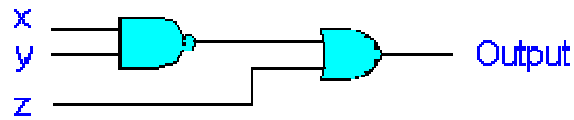


Gambar 4.12. Interpretasi 2 buah simbol untuk gerbang

QUIZ 4

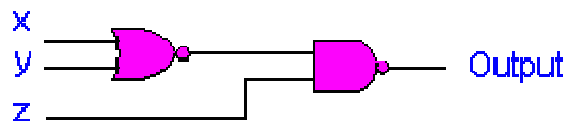
Pilihlah jawaban yang tepat untuk pertanyaan berikut ini !

1. Tentukan ekspresi boolean untuk rangkaian gerbang berikut ini :



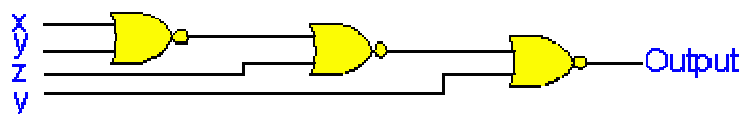
- $x'y'+z$
- $(x'+y')z$
- $x'y'z$
- $x'+y'+z$
- NA

2. Tentukan ekspresi boolean untuk rangkaian gerbang berikut ini :



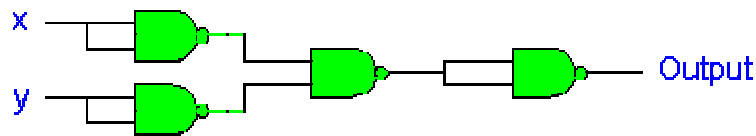
- $x+y+z$
- $x+y+z'$
- $x'y'z$
- $x'+y'+z'$
- NA

3. Tentukan ekspresi boolean untuk rangkaian gerbang berikut ini :



- $xz'+y$
- $xz+y$
- $x'z+y'$
- $x'y'+y'z'$
- $x'y'+y'z$

4. Gerbang logika apa yang ekuivalen dengan rangkaian gerbang NAND berikut ini ?

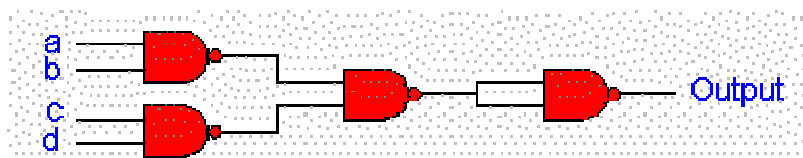


- AND
 - OR
 - NAND
 - NOR
 - Tidak ada jawaban yang benar
5. Ekspresi boolean mana yang ekuivalen dengan fungsi $f=x+yz'$?
- $x(y'+z)$
 - $x(y'+z)$
 - $(y+x)(z'+x)$
 - $(y+x')(x'+z')$
 - Tidak ada jawaban yang benar

6. Ada rangkaian gerbang logika yang bisa dinyatakan hanya dengan menggunakan gerbang :

- AND
- OR
- NOT
- Gerbang apapun bisa
- NAND

7. Tentukan ekspresi boolean untuk rangkaian gerbang berikut ini :



- abcd
- $ab+cd$

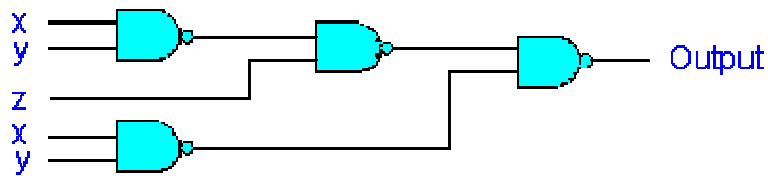
- c. $(a+b)(c+d)$
- d. $a+b+c+d$
- e. $(a'+b')(c'+d')$

8. Gunakan teorema de morgan untuk menentukan F' jika diberikan

$$F = A'B + (C'E)(D+F')$$

- a. $ACE' + BCE' + D'F$
- b. $(A+B')(CE'D'F)$
- c. $A+B+CE'D'F$
- d. $ACE' + AD'F + B'CE' + B'D'F$
- e. NA

9. Tentukan ekspresi boolean untuk rangkaian gerbang berikut ini :



- a. $x'+y'+z'$
- b. $x+y+z$
- c. $x'z'+y'z'$
- d. $xy+z$
- e. z

10. Sederhanakanlah ekspresi boolean : $\{[(AB)'C]D\}'$

- a. $(A'+B')C+D'$
- b. $(A+B')C'+D'$
- c. $A'+(B'+C)D$
- d. $A'+B'+C'+D'$
- e. $A+B+C+D$

Bab V

FLIP-FLOP

5.1. Introduction

5.2. Edge-Triggered Flip-flops

5.3. Pulse-Triggered (Master-Slave) Flip-flops

5.4. Data Lock-Out Flip-flops

5.5. Applications

5.1. Pendahuluan

Flip-flops adalah sistem digital yang memiliki sifat bistabil dan sinkron (synchronous bistable). Sifat sinkron memiliki arti bahwa keluaran flip-flop akan berubah jika dan hanya jika sinyal detak (clock) diberikan. Perubahan keluaran tersebut terjadi secara sinkron dengan sinyal detak yang diberikan.

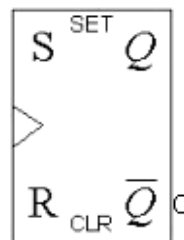
Flip-flop adalah salah satu jenis multivibrator. Ada 3 jenis multivibrator :

1. Monostabel Multivibrator (*one-shot multivibrator*), multivibrator jenis ini hanya stabil di satu keadaan logika (stabil di salah satu dari LOW atau HIGH).
2. Bistable Multivibrator, multivibrator jenis ini stabil di kedua keadaan logika (stabil di LOW dan stabil di HIGH).
3. Astable Multivibrator, multivibrator jenis ini tidak stabil di kedua keadaan logika (tidak stabil di LOW dan tidak stabil di HIGH), sering digunakan sebagai osilator (penghasil sinyal detak).

Berikut ini akan dibahas tentang tiga jenis elemen bistabil, yaitu edge-triggered flip-flop, pulse-triggered (master slave) flip-flop, and data lock-out flip-flop. Kemudian dilanjutkan dengan pembahasan tentang aplikasi tingkat dasar.

5.2. Edge-Triggered Flip-flops

Pada flip-flop jenis ini, keadaan logika rangkaian berubah secara bersamaan ketika terjadi transisi LOW \rightarrow HIGH (*rising edge*) atau HIGH \rightarrow LOW (*falling edge*) pada sinyal detak masukan. Ada 3 macam flip-flop jenis ini yaitu : S-R, J-K dan D.



Gambar 5.1. Edge-Triggered Flip-flops

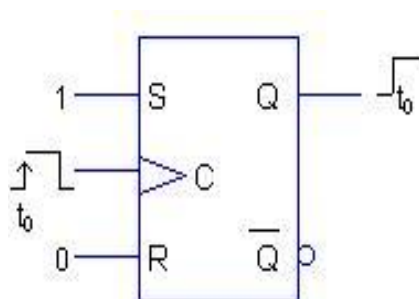
Keterangan :

- Tanda segitiga kecil menyatakan input tersebut bersifat dinamis dan juga menyatakan jenis edge-triggered flip-flop (flip-flop yang terpicu oleh transisi tepian)
- Tepatnya :
 - **Positive Edge Triggered Flip-Flop** (jika tidak terdapat tanda bulat/lingkaran kecil)
 - **Negative Edge Triggered Flip-Flop** (jika terdapat tanda bulat/lingkaran kecil)

Masukan S-R, J-K dan D inputs adalah masukan sinkron karena data pada masukan ini dimasukkan ke flip-flop dan ditransfer ke sisi keluaran hanya pada saat sinyal detak diberikan pada sisi masukan flip-flop. Di sisi lain, masukan direct set (SET) dan clear (CLR) adalah merupakan masukan asinkron, sebagaimana masukan tersebut tidak tergantung pada sinyal detak (clock). Supaya operasi sinkron daripada flip-flop dapat berkerja dengan benar, maka masukan asinkron ini harus dijaga LOW.

5.2.1. Edge-triggered S-R flip-flop

Operasi dasar dari sebuah S-R flip-flop dijelaskan pada tabel kebenaran di bawah ini. Operasi dan tabel kebenaran dari negative edge-triggered flip-flop adalah hampir sama dengan positive-triggered flip-flop. Sedikit perbedaan adalah trigernya terjadi pada saat perubahan logika HIGH ke LOW pada sinyal detak.



Gambar 5.2. Edge-triggered S-R flip-flop

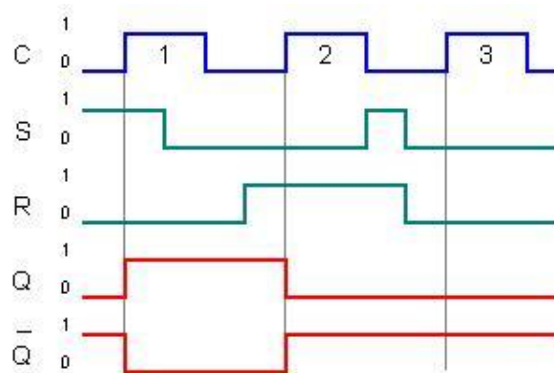
Tabel 5.1. Tabel Kebenaran S-R Flip-Flop

Inputs			Outputs		Comments
S	R	C	Q	Q'	
0	0	↑	Q	Q'	No change
0	1	↑	0	1	RESET
1	0	↑	1	0	SET
1	1	↑	?	?	Invalid

Jika $S = 1$, $R = 0$, Flip-flop akan berada pada keadaan SETS saat perubahan logika LOW ke HIGH.

Catatan : Nilai logika masukan S dan R dapat diubah kapan saja (pada saat sinyal detak LOW maupun $HIGH$) kecuali disekitar transisi sinyal detak).

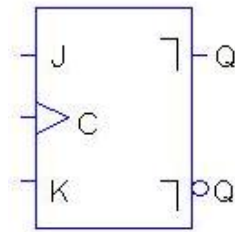
Secara diagram waktu dapat dijelaskan sebagai berikut :



Gambar 5.3. Diagram waktu untuk Edge-triggered S-R flip-flop

5.2.2 Edge-triggered J-K flip-flop

J-K flip-flop bekerja sangat mirip dengan S-R flip-flop. Perbedaannya adalah bahwa pada J-K flip-flop tidak terdapat keadaan operasi INVALID. Pada J-K flip-flop terdapat operasi toggle, yaitu bahwa pada saat masukan J dan K keduanya bernilai logika HIGH, maka keluaran akan berubah nilai logikanya menjadi kebalikan dari nilai logika sebelumnya untuk tiap sinyal detak yang diberikan. Tabel kebenaran J-K flip-flop adalah sebagai berikut :



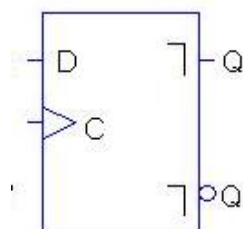
Gambar 5.4. Edge-triggered J-K flip-flop

Tabel 5.2. Tabel Kebenaran J-K Flip-Flop

Inputs			Outputs		Comments
J	K	C	Q	Q'	
0	0	↑	Q	Q'	No change
0	1	↑	0	1	RESET
1	0	↑	1	0	SET
1	1	↑	Q'	Q	Toggle

5.2.3. Edge-triggered D flip-flop

Operasi dari D flip-flop adalah jauh lebih sederhana daripada flip-flop yang lain. D flip-flop hanya memiliki dua masukan yaitu masukan D dan masukan sinyal detak (*clock*). Flip-flop ini banyak digunakan untuk transmisi data digital dan penguncian data. Jika logika masukan D bernilai HIGH dan sinyal detak diberikan maka keluaran akan bernilai sama dengan masukan D yaitu HIGH. Pada keadaan ini flip-flop dinamakan operasi SET. Jika logika masukan D bernilai LOW dan sinyal detak diberikan maka keluaran akan bernilai sama dengan masukan D yaitu LOW. Pada keadaan ini flip-flop dinamakan operasi RESET. Tabel kebenaran dari D flip-flop adalah sebagai berikut :



Gambar 5.5. Edge-triggered D flip-flop

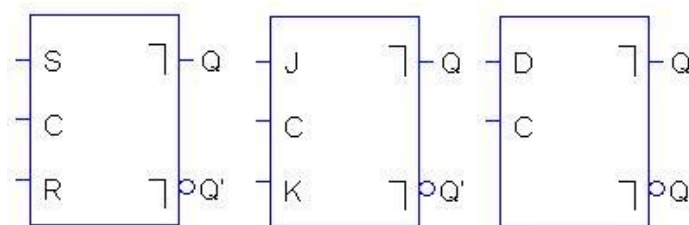
Tabel 5.3. Tabel Kebenaran D Flip-Flop

Inputs		Outputs		Comments
D	C	Q	Q'	
0	↑	0	1	RESET
1	↑	1	0	SET

5.3. Pulse-Triggered (Master-Slave) Flip-flops

Istilah *pulse-triggered* berarti bahwa data akan masuk ke flip-flop pada saat terjadi perubahan logika sinyal detak dari LOW ke HIGH, tetapi keluarannya baru akan terjadi pada saat sinyal detak kembali ke logika LOW. Selama sinyal detak masih bertahan pada logika HIGH, flipflop akan sangat sensitif terhadap perubahan masukan. Untuk itu masukan harus lebih diutamakan daripada sinyal detak. Hal ini berarti bahwa masukan dijaga konstan selama sinyal detak belum kembali berubah ke logika LOW.

Ada 3 jenis *pulse-triggered flip-flops* yaitu S-R, J-K and D. Simbol untuk ketiga flip-flop tersebut adalah seperti gambar berikut ini.

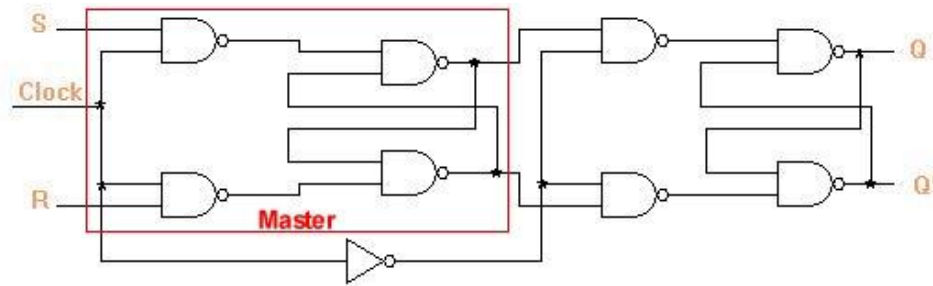


Gambar 5.6. Pulse-Triggered (Master-Slave) Flip-flops

Catatan : Tidak terdapat tanda input dinamik pada sinyal clock, tetapi ada tanda tundaan pada keluaran.

Tabel kebenaran untuk ketiga flipflop tersebut adalah sama dengan dengan flip-flop sebelumnya (*edge-triggered flip-flops*) hanya berbeda dalam hal bagaimana cara memberikan sinyal detak/clock. Flip-flop jenis ini sering dinamakan Master-Slave flip-flop, hal ini dikarenakan konstruksi internal dari flip-flop dibagi dalam dua bagian yaitu master dan slave. Bagian slave adalah sama

dengan bagian master hanya berbeda dalam sinyal detaknya saja. Sinyal detak pada bagian slave adalah kebalikan dari masukan utama sinyal detak dan dikendalikan oleh keluaran dari bagian master. Diagram dari flip-flop jenis ini adalah seperti gambar berikut ini.

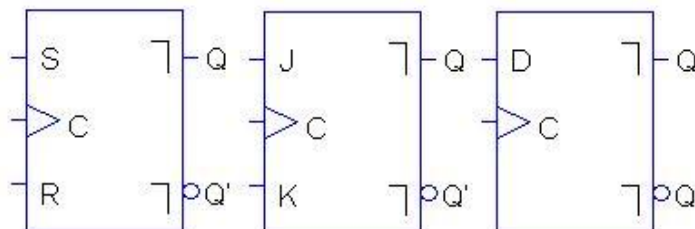


Gambar 5.7. Pulse-Triggered (Master-Slave) S-R Flip-flops

5.4. Data Lock-Out Flip-flops

The data lock-out flip-flop (master – slave) sama dengan pulse-triggered (master-slave) flip-flop kecuali dalam hal masukan sinyal detak dinamik. Sinyal detak dinamik menonaktifkan (*Locks out*) data masukan setelah perubahan logika LOW ke HIGH pada sinyal detak. Sehingga keadaan masukan tidak harus dijaga konstant selama sinyal detak berada pada logika HIGH. Bagian master dari flip-flop jenis ini seperti tipe *edge-triggered*. Bagian slave menjadi bersifat pulse-triggered device untuk menunda perubahan keluaran sehingga perubahan keluaran tersebut terjadi pada saat perubahan logika HIGH ke LOW pada sinyal detak.

Simbol logika dari S-R, J-K dan D jenis lock-out flip-flops adalah seperti yang ditunjukkan di bawah ini. Ketiganya memiliki tanda masukan dinamik dan tundaan keluaran.



Gambar 5.8. Data Lock-Out Flip-flops

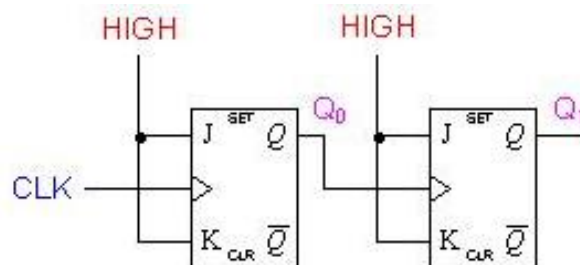
Tabel kebenaran flip-flop jenis ini sama dengan jenis lainnya, hanya berbeda dalam hal bagaimana cara sinyal detak diberikan.

5.5.Applications Flip-Flop

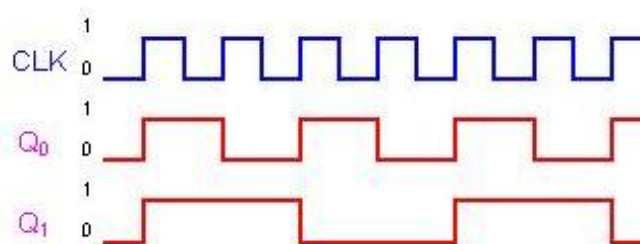
Aplikasi flip-flop dalam sistem digital sangat luas sekali. Beberapa diantaranya akan dibahas berikut ini.

5.5.1. Frequency Division

Pada saat sinyal detak dengan frekuensi sebesar $f_{(clk)}$ diberikan pada masukan clock J-K flip-flop yang dikondisikan operasi toggle, maka keluaran Q adalah berupa gelombang kotak (square wave) dengan frekuensi setengah dari frekuensi sinyal detak ($f_Q = 0,5 \cdot f_{(clk)}$). Jika dilakukan penambahan flip-flop dan menyusunnya secara bertingkat, maka akan didapatkan frekuensi keluaran yang bermacam – macam sesuai dengan pembagian frekuensi.



Gambar 5.9. Rangkaian Pembagi Frekuensi dengan J-K Flip-Flop



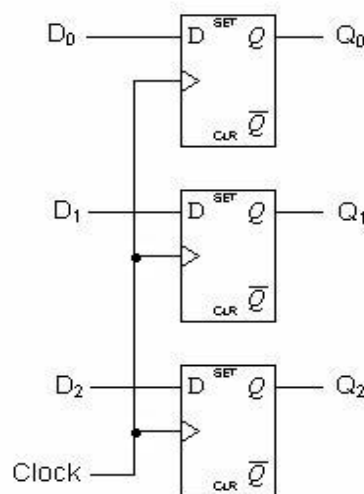
Gambar 5.10. Diagram Waktu Pembagi Frekuensi dengan J-K Flip-Flop

Gelombang kotak yang dihasilkan pada Q flip-flop yang kedua memiliki frekuensi $0,25 \cdot f_{(clk)}$. Hal ini dikarenakan frekuensi sinyal detak masukan telah dibagi dua oleh flip-flop yang pertama, kemudian dibagi lagi dengan dua oleh

flip-flop kedua. Jika ada n buah flip-flop yang disusun secara bertingkat, maka keluaran akhirnya akan memiliki frekuensi sebesar : $f_Q = f_{(clk)} / (2^n)$.

5.5.2. Parallel Data Storage

Dalam sistem digital, pada umumnya data disajikan dalam bentuk sekumpulan digit biner yang membentuk sebuah angka, kode atau informasi lain. Sehingga diperlukan cara untuk mengirim data tersebut secara paralel dan berbarengan menggunakan sekumpulan flip-flop. Operasi ini dapat dijelaskan sebagai berikut :



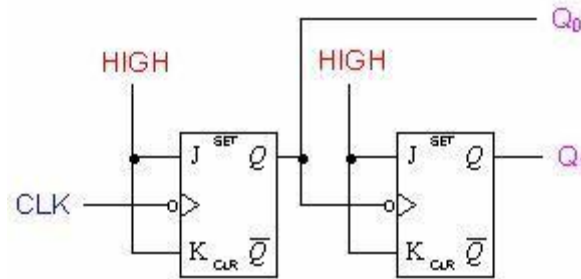
Gambar 5.11. Rangkaian Paralel Data Storage dengan D Flip-Flop

Ketiga saluran data dihubungkan pada masukan D dari sebuah flip-flop. Jika ketiga masukan sinyal detak ketiga flip-flop tersebut dihubungkan ke sebuah sumber sinyal detak yang sama, maka data masukan pada D akan dikirim ke sisi keluaran pada saat yang berbarengan dengan perubahan nilai logika LOW ke HIGH pada sinyal detak. Register, sebagai kumpulan flip-flop yang digunakan sebagai penyimpan data akan dibahas lebih lanjut pada bagian berikutnya.

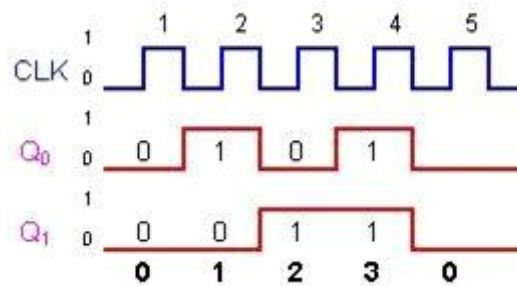
5.5.3. Counting

Aplikasi lain dari flip-flop yang sangat penting adalah pencacah digital (*digital counters*), pembahasan secara mendalam akan disampaikan pada bagian

selanjutnya. Sistem pencacah yang mencacah dari 0 sampai 3 dijelaskan seperti gambar disamping. Pencacahan 2 digit biner terjadi setiap 4 pulsa detak diberikan. Ketika hasil pencacahan mencapai 3, akan direset kembali ke 0 dan akan kembali mulai mencacah.



Gambar 5.12. Pencacah 2 bit dengan J-K Flip-Flop



Gambar 5.13. Diagram Waktu Pencacah 2 bit dengan J-K Flip-Flop

Bab VI

COUNTER

- 6.1. Introduction**
- 6.2. Asynchronous (Ripple) Counters**
- 6.3. Asynchronous Decade Counters**
- 6.4. Asynchronous Up-Down Counters**
- 6.5. Synchronous Counters**
- 6.6. Synchronous Decade Counters**
- 6.7. Synchronous Up-Down Counters**
- 6.8. Applications**

6.1. Pendahuluan

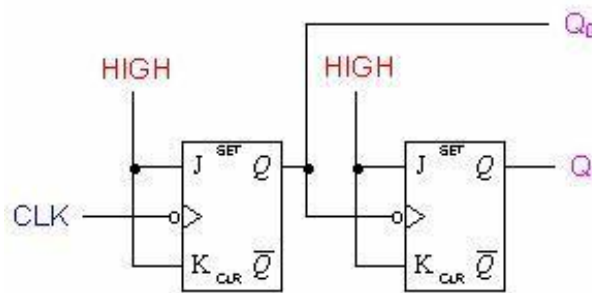
Rangkaian pencacah sering digunakan pada komputer dan sistem digital yang lain. Jika pencacah diinginkan mampu mengingat hasil pencacahan sebelumnya, maka diperlukan memori pada pencacah tersebut. Pada bab sebelumnya telah disampaikan bagaimana flip-flop dapat digunakan untuk membuat pencacah digital. Berapa flip-flop yang digunakan dan bagaimana menghubungkan flip-flop tersebut akan menentukan jumlah pencacahan yang bisa terjadi.

Counters dapat dikelompokkan menjadi 2 berdasarkan bagaimana cara melakukan pentriggeran (Clocking), yaitu :

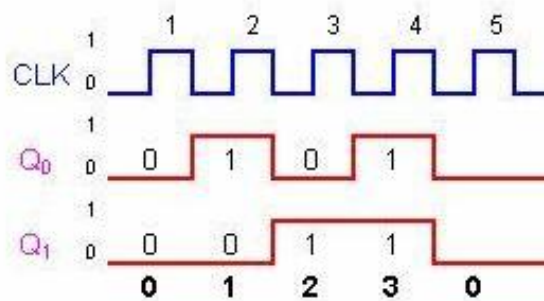
1. *Asynchronous (Ripple) Counters* – pada pencacah jenis tak-sinkron flip-flop pertama ditrigger dari sinyal detak yang bersumber dari luar, selanjutnya keluaran Q atau Q' akan digunakan untuk mentrigger flip-flop berikutnya, demikian seterusnya secara beruntun.
2. *Synchronous Counters* – pada pencacah sinkron semua flip-flop ditrigger oleh sinyal detak yang sama dan secara simultan/berbarengan.

6.2. Asynchronous(Ripple) Counters

Pada gambar berikut ini ditunjukkan sebuah pencacah tak-sinkron 2 bit.



Gambar 6.1. Pencacah tak-sinkron 2 bit



Gambar 6.2. Diagram Waktu Pencacah tak-sinkron 2 bit

Catatan : Pada diagram sinyal diatas, perubahan Q_0 , Q_1 dan Clock digambarkan terjadi secara simultan walaupun untuk pencacah asinkro. Hal ini dengan tujuan untuk kesederhanaan dalam analisa. Pada kenyataannya terdapat jeda waktu antara sinyal detak (CLK) dengan perubahan Q_0 dan Q_1 .

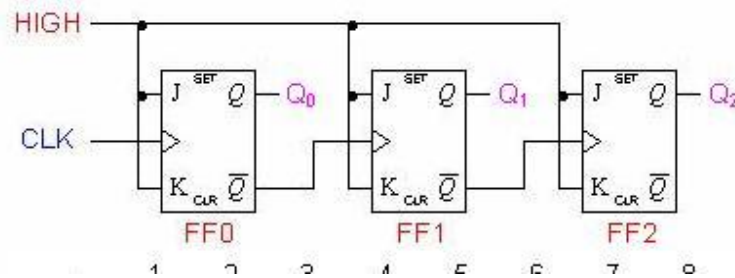
Sinyal detak eksternal dihubungkan pada masukan clock pada flip-flop yang pertama. Dengan demikian flip-flop pertama ini akan berubah keadaannya setiap kali terjadi perubahan HIGH ke LOW pada sinyal clock, sedangkan pada flip-flop kedua berubah hanya pada saat ditrigger oleh perubahan HIGH ke LOW pada keluaran Q dari flip-flop pertama. Karena adanya sifat penunda pada flip-flop, maka perubahan sinyal clock dan perubahan keluaran Q tidak terjadi pada waktu yang benar-benar bersamaan.

Biasanya semua masukan CLEAR digabung menjadi satu, sehingga sebuah sinyal clock tunggal dapat mereset semua flip-flop sebelum pencacahan dimulai. Sinyal clock diberikan pada flip-flop pertama akan dilanjutkan secara berulang ke flip-flop berikutnya layaknya sebuah riak di atas permukaan air, dari sinilah alasan kenapa pencacah ini dinamakan *Ripple Counter*.

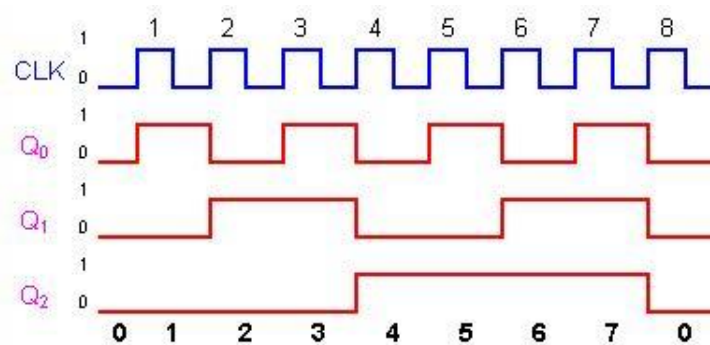
Ripple counter 2 bit seperti gambar diatas memiliki 4 macam keadaan yang berbeda, masing-masing keadaan keluaran tersebut menyatakan nilai pencacahan. Dengan cara yang sama, jika pencacah dengan n buah flip-flop maka akan memiliki 2^n macam keadaan logika keluaran. Jumlah macam keadaan keluaran pencacah ini dinamakan modulo dari pencacah. Sehingga untuk pencacah 2 bit

memiliki 4 macam keadaan logika keluaran yang berbeda dinamakan pencacah *modulo 4*.

Berikut ini disampaikan contoh pencacah asinkron 3 bit lengkap dengan diagram sinyal terhadap waktu.



Gambar 6.3. Pencacah tak-sinkron 3 bit

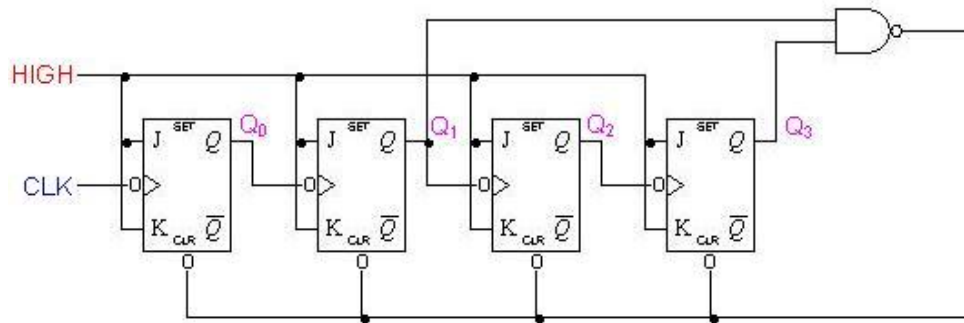


Gambar 6.4. Diagram Waktu Pencacah tak-sinkron 3 bit

6.3. Asynchronous Decade Counters

Pada pembahasan pencacah biner sebelumnya dijelaskan bahwa pencacah memiliki 2^n kemungkinan keadaan keluaran. Namun demikian tidak tertutup kemungkinan suatu pencacah memiliki lebih sedikit kemungkinan keadaan keluaran. Pencacah ini dirancang jumlah pencacahan tertentu sesuai keinginan kita. Hal ini dilakukan dengan memaksa pencacah untuk mereset kembali pencacahannya sebelum mencapai cacahan maksimum.

Salah satu contoh jenis pencacah ini adalah *decade counter*. Pencacah jenis ini juga sering dinamakan sebagai pencacah modulo 10. Pencacah ini dibuat dengan melakukan modifikasi terhadap pencacah biner 4 bit (modulo 16). Rangkaian *decade counter* adalah seperti pada gambar berikut ini.



Gambar 6.5. 4. Asynchronous Decade Counters

Pada saat pencacahan mencapai 1010, maka semua flip-flop di-RESET. Sinyal reset diambil dari menggabungkan keluaran flip-flop ke 2 dan 4 melalui sebuah gerbang AND. Urutan hasil pencacahan decade counter adalah seperti tabel berikut ini.

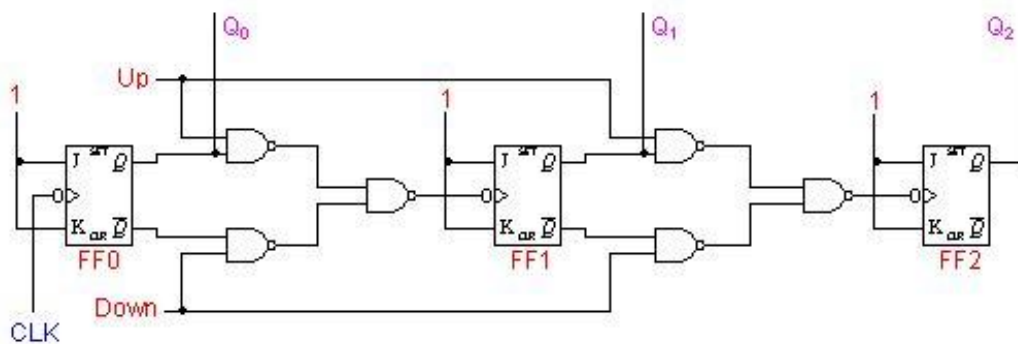
Tabel 6.1. Pencacahan pada Asynchronous Decade Counters

Clock Pulse	Q3	Q2	Q1	Q0
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

6.4. Asynchronous Up-Down Counters

Pada penggunaan pencacah dalam suatu sistem digital yang lebih kompleks, seringkali sebuah pencacah harus mampu melakukan pencacahan naik maupun turun. Rangkaian dibawah ini menunjukkan sebuah pencacah 3 bit yang bisa mencacah naik maupun turun (*3-bit up-down counter*). Pencacah ini akan mencacah naik atau turun tergantung pada sinyal kendali UP/DOWN.

Pada saat masukan UP diberi logika HIGH, dan masukan DOWN diberi logika LOW, rangkaian gerbang NAND antara flip-flop FF0 dan FF1 akan meloloskan sinyal keluaran Q dari FF0, dan akan masuk sebagai clock bagi FF1. Demikian juga keluaran Q dari FF1 akan lolos dan masuk ke clock FF2. Dengan semikian pencacah akan mencacah naik (*count up*).



Gambar 6.6. Asynchronous Up-Down Counters

Tabel 6.2. Pencacahan pada Asynchronous Up-Down Counters

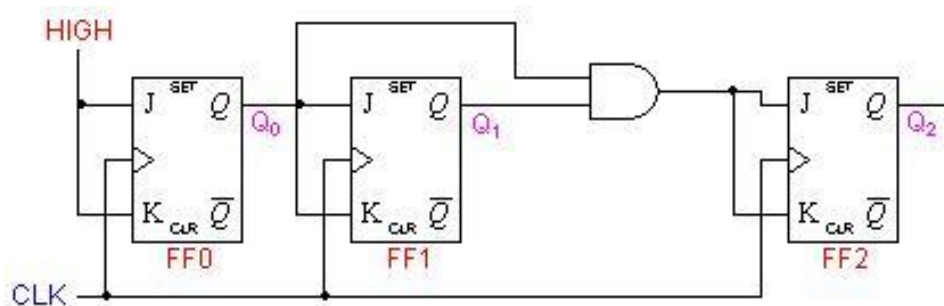
FF2	FF1	FF0
0	0	0
1	1	1
1	1	0
1	0	1
1	0	0
0	1	1
0	1	0
0	0	1

Pencacah asinkron yang bisa naik-turun bekerja lebih lambat dibanding dengan pencacah naik atau pencacah turun. hal ini dikarenakan adanya tambahan penundaan yang terjadi pada rangkaian gerbang NAND.

Pada saat masukan UP diberi logika LOW, dan masukan DOWN diberi logika HIGH, rangkaian gerbang NAND antara flip-flop FF0 dan FF1 akan meloloskan sinyal keluaran \bar{Q} dari FF0, dan akan masuk sebagai clock bagi FF1. Demikian juga keluaran \bar{Q} dari FF1 akan lolos dan masuk ke clock FF2. Dengan semikian pencacah akan mencacah turun (*count down*).

6.5. Synchronous Counters

Pada pencacah sinkron, masukan clock pada semua flip-flop digabungkan menjadi satu dan dihubungkan dengan satu sumber sinyal clock yang sama. Dengan semikian semua flip-flop akan berubah keadaan secara bersamaan (serempak). berikut ini adalah rangkaian pencacah sinkron 3 bit (**3-bits synchronous counter**). Masukan J dan K pada flip-flop FF0 diberikan logika HIGH. Masukan J dan K pada flip-flop FF1 dihubungkan dengan keluaran Q dari FF0. Sedangkan masukan J dan K dari FF2 dihubungkan dengan keluaran q dari FF0 dan FF1 melalui sebuah gerbang AND.



Gambar 6.7. Synchronous Counters

Tabel 6.3. Pencacahan pada Synchronous Counters

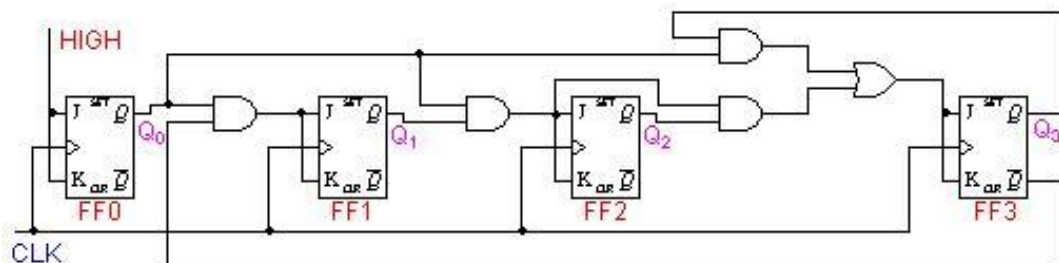
FF2	FF1	FF0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Jika sinyal clock diberikan, maka pada saat setelah sinyal clock yang ke 3 maka keadaan keluaran flip-flop FF0 dan FF1 keduanya berlogika HIGH. Sehingga pada sinyal clock yang ke-4 akan merubah keluaran FF2.

Kelebihan dari pencacah sinkron adalah dalam hal waktu tunda. Tidak ada akumulasi waktu tunda, hal ini karena semua flip-flop di-triger secara serempak. Sehingga memungkinkan pencacahan secara lebih cepat dibanding dengan pencacah asinkron.

6.6. Synchronous Decade Counters

Sama seperti pencacah dekade tak-sinkron, pencacah dease sinkron mencacah dari 0 sampai 9 dan kemudian akan kembali lagi ke 0. Hal ini terjadi karena keadaan logika keluaran 1010 digunakan untuk mereset pencacah kembali ke 0000. Rangkaian pencacah jenis ini adalah sebagai berikut.



Gambar 6.8. Synchronous Decade Counters

Tabel 6.4. Pencacahan pada Synchronous Decade Counters

Clock Pulse	Q3	Q2	Q1	Q0
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

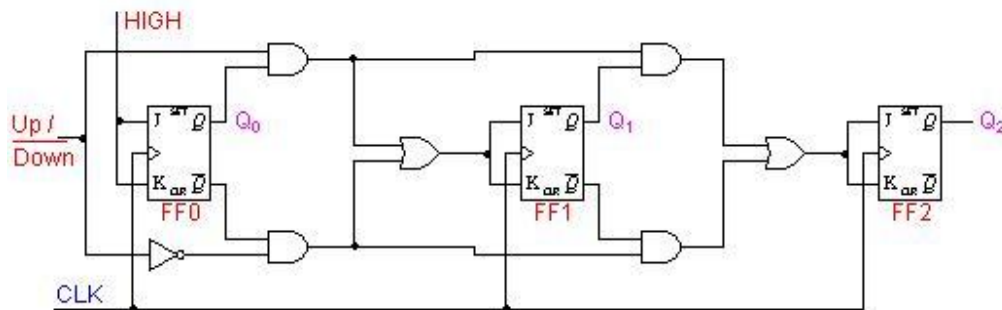
Dari tabel disimpulkan :

- Q0 bersifat togel pada setiap sinyal clock yang diberikan.

- Q1 berubah pada clock berikutnya setiap kondisi Q0=HIGH dan Q3=LOW
- Q2 berubah pada clock berikutnya setiap kondisi Q0=Q1=HIGH
- Q3 berubah pada clock berikutnya setiap kondisi Q0=Q1=Q2=HIGH (cacahan ke 7) atau pada saat Q0=Q3=HIGH (cacahan ke 9)

6.7. Synchronous Up-Down Counters

Rangkaian pencacah naik-turun 3 bit sinkron dan tabel pencacahannya ditunjukkan pada gambar di bawah ini. Sama seperti pencacah naik-turun 3 bit tak-sinkron, pencacah ini juga mempunyai masukan kendali untuk pencacahan naik atau pencacahan turun.



Gambar 6.9. Synchronous Up-Down Counters

Tabel 6.5. Pencacahan pada Synchronous Decade Counters

Up / Down	Q2	Q1	Q0	Down
	0	0	0	
	0	0	1	
	0	1	0	
	0	1	1	
	1	0	0	
	1	0	1	
	1	1	0	
	1	1	1	

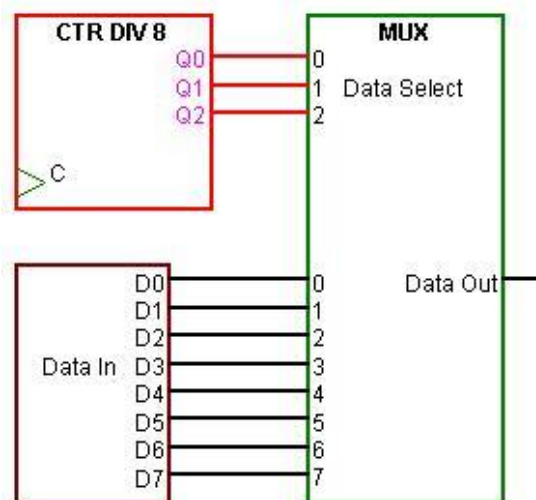
Berdasarkan tabel, maka dapat disimpulkan :

- Untuk keadaan (Naik dan turun), Q0 bersifat togel pada setiap sinyal clock yang diberikan
- Untuk pencacahan naik, Q1 berubah nilai logikanya pada tiap clock berikutnya jika Q0=HIGH

- Untuk pencacahan turun, Q1 berubah nilai logikanya pada tiap clock berikutnya jika Q0=LOW
- Untuk pencacahan naik, Q2 berubah nilai logikanya pada tiap clock berikutnya jika Q0=Q1=HIGH
- Untuk pencacahan turun, Q2 berubah nilai logikanya pada tiap clock berikutnya jika Q0=Q1=LOW

6.8. Aplikasi

Pencacah digital sangat banyak penggunaannya dalam berbagai aplikasi sistem digital. Sebagai contoh adalah jam digital dan konversi data paralel menjadi serial (*multiplexing*). Berikut ini disajikan contoh penggunaan pencacah digital untuk konversi data paralel menjadi data serial. Data digital 8 bit yang masuk secara serempak pada saluran paralel dinamakan sebagai data paralel (*parallel data*). Data yang keluar secara berurutan dari satu saluran disebut data serial (*serial data*).



Gambar 6.10. Aplikasi Counter pada Konversi Data Paralel ke Data Serial

Keluaran Q dari pencacah modulo 8 dihubungkan dengan jalur masukan pemilih data (*data select*) pada multiplexer 8 bit. Data 8 bit paralel diberikan pada jalur masukan data (*data input*) pada multiplexer.

Pada saat pencacahan terjadi dari 0 sampai 7, tiap-tiap bit data masukan, dimulai dari D0 secara berurutan diloloskan ke sisi keluaran multiplekser (*data out*).

Setelah 8 kali sinyal clock diberikan pada pencacah, maka data paralel 8 bit telah dikonversikan menjadi data 8 bit serial dan dikirim ke jalur keluaran daripada multiplekser. Kemudian pencacah akan reset kembali ke posisi 0.

Bab VII

REGISTER GESER (*SHIFT REGISTER*)

- 7.1. Pendahuluan**
- 7.2. Serial In - Serial Out Shift Registers**
- 7.3. Serial In - Parallel Out Shift Registers**
- 7.4. Parallel In - Serial Out Shift Registers**
- 7.5. Parallel In - Parallel Out Shift Registers**
- 7.6. Bidirectional Shift Registers**
- 7.7. Shift Register Counters**
- 7.8. Aplikasi**

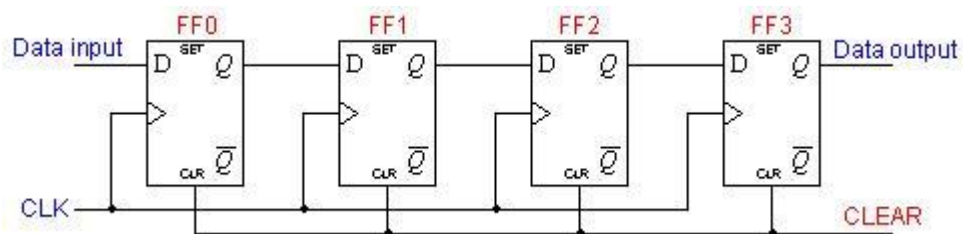
7.1. Pendahuluan

Shift registers adalah salah satu jenis rangkaian logika sekuensial yang banyak difungsikan untuk penyimpanan data digital. *Shift registers* terdiri dari dari sekelompok flip-flop yang terhubung secara berantai sehingga keluaran sebuah flip-flop akan menjadi masukan bagi flip-flop yang berikutnya. Kebanyakan flip-flop tidak memiliki karakteristik sekuensial internal. Semua flip-flop digerakkan oleh sebuah sinyal detak (clock) secara bersama dan fungsi set atau reset dijalankan secara serentak terhadap seluruh flip-flop yang ada.

Pada bab ini, akan dibahas jenis *shift registers* yang paling dasar, seperti serial input-serial output, serial input-paralel output, paralel input-serial output, paralel input-paralel output dan shift register dua arah. Pencacah khusus yaitu pencacah jenis *shift registers* juga akan dibahas dalam bab ini.

7.2. Serial In - Serial Out Shift Registers

Shift registers 4 bit dapat dibuat dengan menggunakan 4 buah D flip-flop sebagaimana ditunjukkan pada gambar di bawah ini.



Gambar 7.1. Shift Register 4 jenis Serial input-Serial Output

Prinsip kerja dari rangkaian tersebut adalah sebagai berikut :

1. Pertama-tama register di reset (Cleared) akan menyebabkan semua keluaran akan bernilai nol (logika LOW).

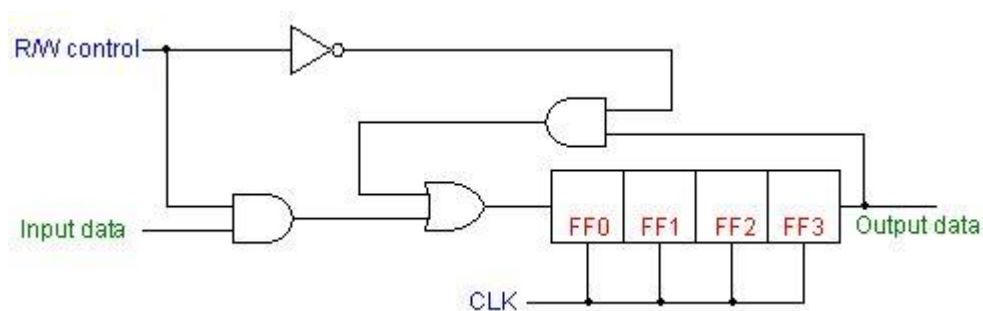
CLEAR	FF0	FF1	FF2	FF3
	0	0	0	0

2. Masukan data diberikan secara sekuensial (berurutan) pada input D pada flip-flop yang pertama (flip-flop FF0)

3. Pada setiap kali sinyal detak diberikan, maka akan terjadi pergeseran data dari kiri ke kanan sebesar satu bit.
4. Misalkan data yang dimasukkan adalah 1001, bit LSB dari data tersebut akan digeser dari flip-flo FF0 sampai ke flip-flop FF3. Dengan demikian data 4 bit tersebut telah disimpan ke dalam register setelah terjadi 4 kali sinyal detak berturut-turut.
5. Jika diinginkan mengeluarkan data dari register secara serial, dapat dilakukan dengan 2 cara yaitu secara *destructively readout* atau *non-destructively readout*.
6. Untuk pembacaan cara pertama (*destructive readout*), data asli yang tadinya tersimpan dalam register akan hilang sehingga register akan berisi 0000 sma keadaannya ketika dilakukan reset terhadap register tersebut.



7. Untuk menghindari kehilangan data tersebut dapat dilakukan dengan cara kedua (*non-destructive reading*) yaitu dengan menambahkan dua buah gerbang AND, sebuah gerbang OR dan sebuah gerbang NOT. Susunan rangkaiannya adalah seperti gambar berikut ini.



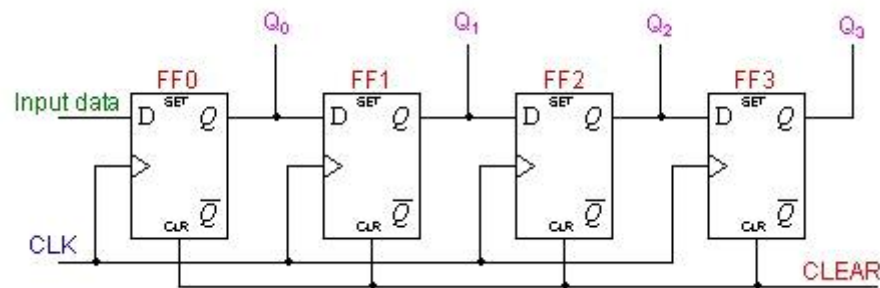
Gambar 7.2. Non-destructively readout pada Shift Register 4 jenis Serial input-Serial Output

8. Pada cara kedua ini, data akan masuk ke dalam register pada saat masukan *R/W control* diberi logika HIGH, dan untuk mengeluarkan data yang telah

tersimpan pada register dilakukan dengan memberikan logika LOW pada masukan *R/W control* tersebut.

7.3. Serial In - Parallel Out Shift Registers

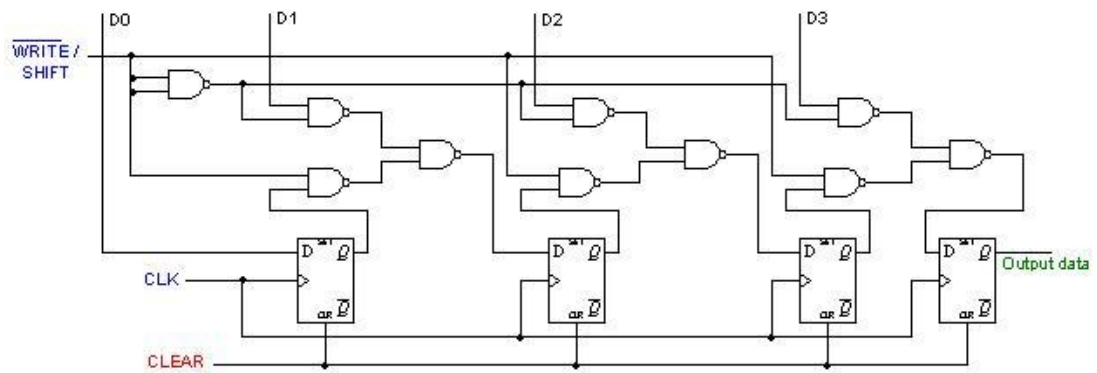
Untuk register jenis ini, data biner dimasukkan secara serial dengan cara yang sama dengan jenis register yang telah dibahas sebelumnya. Perbedaan terdapat dalam hal bagaimana caranya data dikeluarkan dari dalam register. Jika sebuah data telah disimpan dalam register, maka tiap bit data tersebut menempati pada jalur output pada tiap-tiap flip-flop, dan semua bit data tersebut dapat diambil secara serentak. Susunan rangkaian flip-flop untuk register jenis ini adalah sebagaimana pada gambar berikut ini.



Gambar 7.3. 4 Bits Serial In - Parallel Out Shift Registers

7.4. Parallel In - Serial Out Shift Registers

Untuk jenis register berikutnya adalah Parallel In - Serial Out Shift Registers. Sebagai contoh adalah 4 bit parallel in - serial out shift register dapat dilihat pada gambar dibawah ini. Rangkaian tersebut menggunakan D flip-flop dan gerbang NAND untuk memasukkan data ke dalam register.

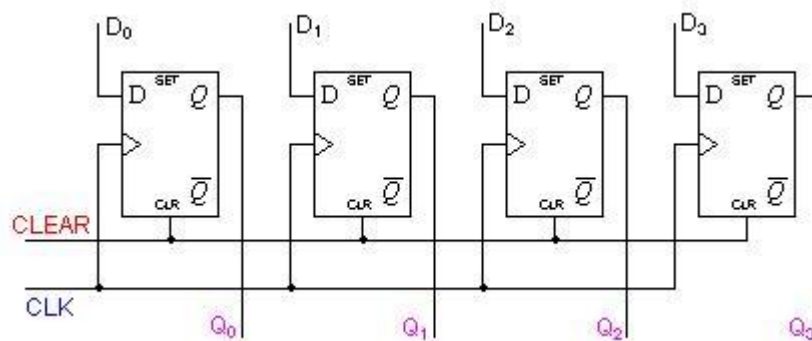


Gambar 7.4. 4 Bits Parallel In - Serial Out Shift Registers

D0, D1, D2 dan D3 adalah sebagai paralel input, dimana D0 adalah sebagai bit paling besar nilainya (*Most Significant Bit*) dan D3 adalah bit yang paling rendah nilainya (*Least Significant Bit*). Untuk memasukkan data ke dalam register dilakukan dengan memberikan logika LOW pada masukan WRITE/SHIFT. Data yang telah tersimpan dalam register dapat dikeluarkan pada jalur OUTPUT DATA dengan memberikan logika HIGH pada masukan WRITE/SHIFT. Data akan keluar bit demi bit untuk setiap sinyal CLK yang diberikan. Untuk mengosongkan isi register secara serempak dilakukan dengan memberikan logika HIGH pada jalur masukan CLEAR.

7.5. Parallel In - Parallel Out Shift Registers

Untuk register jenis ini memiliki masukan data maupun keluaran secara serempak. Rangkaian berikut ini menjelaskan sebuah 4 bits parallel in-parallel out shift register yang disusun menggunakan 4 buah D flip-flop.



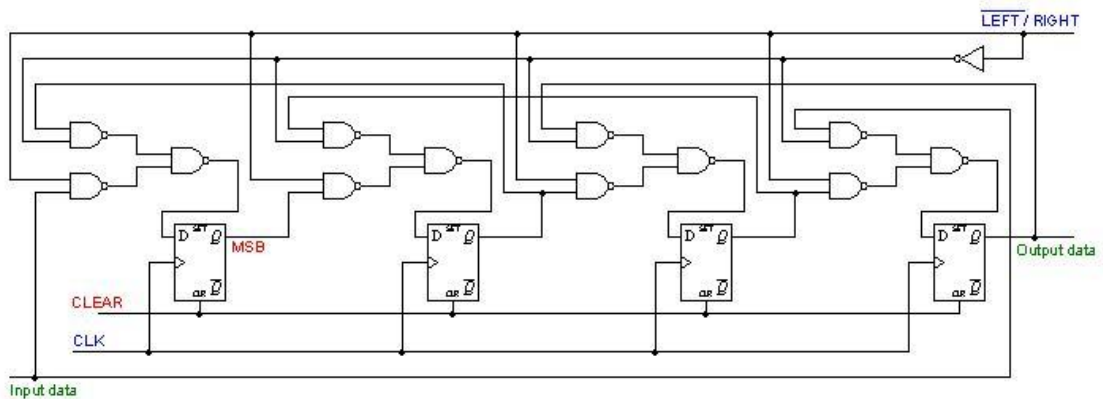
Gambar 7.5. 4 Bits Parallel In - Parallel Out Shift Registers

D0, D1, D2 dan D3 adalah paralel input dan Q0, A1, A2 dan A3 adalah paralel output. Setiap sinyal detak diberikan pada jalur masukan CLK, semua bit data pada masukan D akan langsung ditransfer ke jalur keluaran Q pada masing-masing flip-flop yang ada.

7.6. Bidirectional Shift Registers

Jenis register yang telah dibahas hanya menyangkut satu arah pergeseran data yaitu geser kanan. Setiap pergeseran data ke kanan satu bit mengandung arti sama dengan pembagian data dengan bilangan 2. Jika operasi pergeseran data dibalik kekiri, maka untuk setiap pergeseran 1 bit data tersebut mengandung arti sama dengan perkalian data dengan bilangan 2.

Dengan menambahkan beberapa gerbang NAND dan NOT dengan susunan seperti pada gambar di bawah, maka sebuah shift register bisa dioperasikan geser ke kanan atau ke kiri (*bidirectional shift register* atau *reversible*). Gambar di bawah ini memberikan contoh 4 bits shift register yang bersifat *bidirectional shift* atau *reversible*.



Gambar 7.6. 4 Bits Serial In - Serial Out Bidirectional Shift Registers

Dalam hal ini gerbang NAND disusun sebagai gerbang OR untuk memilih data input dari sisi kiri (pergeseran data ke kanan) atau data input dari sisi kanan (pergeseran data ke kiri). Untuk operasi tersebut dilakukan dengan mengatur nilai logika pada masukan LEFT/RIGHT. Jika logika pada masukan LEFT/RIGHT bernilai LOW maka arah pergeseran data yang terjadi adalah ke kiri (LEFT),

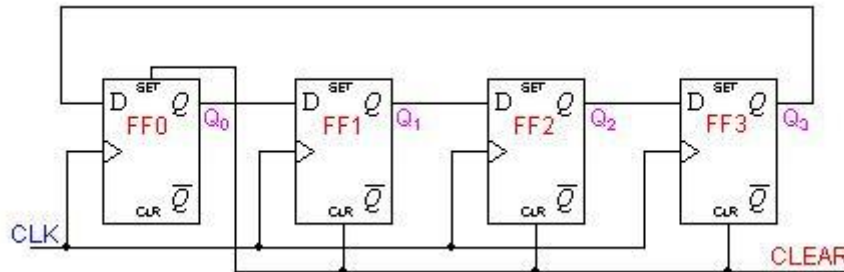
sebaliknya jika HIGH maka arah pergeseran data yang terjadi adalah ke kanan (RIGHT).

7.7. Shift Register Counters

Ada dua jenis pencacah shift register (*shift register counter*) yang paling utama, yaitu Ring Counter dan Johnson Counter. Kedua jenis pencacah shift register tersebut sebenarnya merupakan shift register jenis serial output yang disambungkan kembali ke serial input.

7.7.1. Ring Counters

Berikut ini adalah rangkaian 4 bit ring counter yang tersusun dari D flip-flop. Keluaran D flip-flop pada tiap tingkat diumpankan ke masukan D flip-flop pada tingkat berikutnya untuk setiap perubahan logika LOW – HIGH pada masukan CLK. Jika masukan CLEAR diberikan logika HIGH, semua flip-flop kecuali flip-flop tingkat pertama (FF0) akan direset sehingga keluaran Q berlogika LOW.



Gambar 7.7. 4 Bits Ring Counter

Clock Pulse	Q3	Q2	Q1	Q0
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	1	0	0	0

←

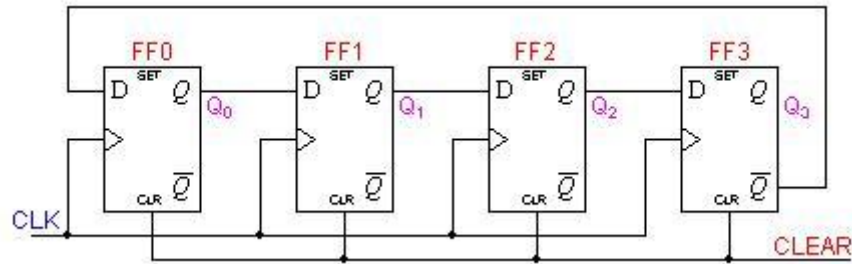
CLEAR

FF0	FF1	FF2	FF3
1	0	0	0

Gambar 7.8. Hasil pencacahan 4 Bits Ring Counter

7.7.2. Johnson Counters

Berikut ini disampaikan rangkaian 4 bits johnson counter yang tersusun dari D flip-flops. Johnson counter merupakan variasi pengembangan daripada ring counter dengan output terbalik pada tingkat paling akhir seperti terlihat pada gambar 6.9.



Gambar 7.9. 4 Bits Johnson Counter

Johnson counter juga sering disebut sebagai twisted ring counter. Hasil pencacahan dari johnson counter adalah seperti gambar berikut ini.

Clock Pulse	Q3	Q2	Q1	Q0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	1	1	1
4	1	1	1	1
5	1	1	1	0
6	1	1	0	0
7	1	0	0	0

CLEAR

FF0	FF1	FF2	FF3
0	0	0	0

Gambar 7.10. Hasil pencacahan 4 Bits Johnson Counter

7.8. Aplikasi Shift Register

Aplikasi shift register adalah sangat luas dalam berbagai hal. Beberapa diantaranya adalah :

- Untuk menghasilkan penundaan waktu (time delay)

Shift register serial input-serial output dapat digunakan untuk rangkaian penunda waktu. Besarnya penundaan waktu dapat diatur dengan :

- § Jumlah tingkatan yang ada pada shift register
- § Kecepatan sinyal detak (Clock) yang diberikan pada shift register
- Untuk penyederhanaan rangkaian logika kombinasional
- Untuk mengkonversi data serial menjadi data paralel

DAFTAR PUSTAKA

- [1] Wijaya Widjanarka, Ir., *Teknik Digital* , Erlangga, Jakarta, 2006.
- [2] Bartee, T.C., Ph.D., *Digital Computer Fundamental* , 6th Ed, Mc Graw Hill, Inc, 1985.
- [3] Tocci, R. J., *Digital System : Principle and Applications* , 4th ed, Prentice Hall Inc., New Jersey.
- [4] Tokheim, R.L., M.Sc., *Digital Principle* , 2nd ed., Mc graw Hill, Inc, Singapore, 1984.
- [5] Malvino, A. P., Ph.D., *Electronic Principles*, 3rd ed., Mc Graw Hill, Singapore, 1984.

