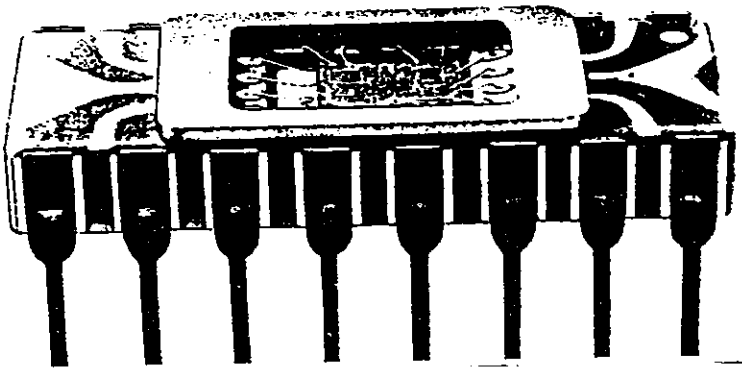
PENGANTAR TEKNIK DIGITAL



MILIK PERPUSTAKAAN IKIP PADANG
DITERIMA TGL 8 - 12 - 94
SUMBER/HARGA hd
KOLEKSI KKI
NO. INVENTARIS 1784/hd/94 - p2(2)
KLASIFIKASI 621.381 tas p2

Oleh
Drs. Elfi Tasrif
Drs. Putra Jaya

Fakultas Pendidikan Teknologi dan Kejuruan
INSTITUT KEGURUAN DAN ILMU PENDIDIKAN
1994

MILIK UPT PERPUSTAKAAN
IKIP PADANG

KATA PENGANTAR

Berkat rakhmat Allah Swt., akhirnya buku "Pengantar Teknik Digital" ini dapat juga penulis selesaikan.

Sasaran utama buku ini adalah para pemula yang akan mengenal teknik digital. Dengan demikian buku ini diharapkan dapat memberikan manfaat bagi pembaca dan menambah kepustakaan dalam bidang teknik elektronika, khususnya teknik digital.

Teknik digital merupakan salah satu cabang ilmu di dalam dunia Elektronika, yang mana perkembangannya saat ini betul-betul luar biasa bahkan kadang lebih cepat dari yang kita duga. Era komputerisasi misalnya, merupakan era yang dimotori dan didasari oleh pengetahuan tentang teknik digital, Khususnya hal-hal yang menyangkut tentang perangkat kerasnya. Lebih lanjut tentang perkembangan dunia elektronika, khususnya teknik digital tersebut dipaparkan pada awal bab buku ini. Pada bab-bab berikut buku ini berisikan konsep-konsep dasar pengetahuan tentang Aljabar Bolean, gerbang-gerbang logika, langkah-langkah dalam menganalisa dan memodifikasi serta simplifikasi rangkaian logika dan pada akhirnya akan ditutup dengan pembahasan tentang rangkaian ilmu hitung.

Selanjutnya dalam kesempatan ini penulis menyampaikan rasa terimakasih kepada pihak-pihak yang telah memberikan saran dan sumbangan pikiran selama penulisan. Dan juga kepada semua pihak yang telah memberikan dukungan baik moril maupun materil sehingga terwujudnya buku ini.

Penulis menyadari bahwa tiada karya yang utuh di dunia fana ini, untuk itu semua kritik membangun dan koreksi dari pembaca, penulis tunggu dengan senang hati demi perbaikan di masa mendatang.

Padang, Maret 1994

Penulis

DAFTAR ISI

	Halaman
KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR TABEL	iii
BAB I. REVOLUSI TEKNIK DIGITAL	1
BAB II. ELEKTRONIKA LOGIKA	3
A. Aljabar Bolean	4
B. Hukum-hukum dan Sifat Persamaan Logika	4
C. Teorema De Morgan	7
BAB III. GERBANG LOGIKA	10
A. Taraf Tegangan	10
B. Gerbang Dasar	13
BAB IV. MODIFIKASI DAN SIMPLIFIKASI RANGKAIAN LOGIKA	24
A. Konversi	24
B. Simplifikasi	29
C. Perencanaan Logika Rangkaian	32
D. Perencanaan Rangkaian Logika Berdasarkan Tabel	35
BAB V. RANGKAIAN ILMU HITUNG	47
A. Rangkaian Penjumlah	47
B. Rangkaian Pengurang	58
DAFTAR BACAAN	76

MILIK UPT PERPUSTAKAAN
IKIP PADANG

DAFTAR TABEL

Halaman

Tabel 4.1. Tabel Ruang Rahasia 35

Tabel 4.2. Tabel dengan Keluaran 1 Jamak 39

Tabel 4.3. Simplifikasi 40

Tabel 4.4. Product of Sums 43

Tabel 4.5. Keluaran dengan Logika 0 Jamak 44

Tabel 5.1. Penjumlahan Biner 49

Tabel 5.2. Subtraksi 60

Tabel 5.3. Pengurang Lengkap 62

BAB I

REVOLUSI TEKNIK DIGITAL

Tingkat keterlibatan bidang elektronika dalam suatu tahapan atau proses saat ini, sering dijadikan tolak ukur atau acuan akan kecanggihan atau bobot dari proses tersebut. Semakin rumit sistem elektronika suatu proses biasanya alat atau proses tersebut cenderung semakin canggih dan berbobot, karena kecanggihan sistem akan menghasilkan produk yang relatif lebih sederhana atau praktis. Indikasi diatas dapat diamati dari semakin banyaknya penggunaan alat yang serba elektronik.

Didalam dunia elektronika itu sendiri dewasa ini, juga terjadi revolusi dan pergerakan yang sedemikian pesatnya. Kemajuan yang pesat ini diawali dari penemuan tabung vakum, seterusnya meningkat kepada transistor dan dari transistor berkembang menjadi era rangkaian terpadu atau *integrated circuit(IC)*. Perkembangan teknologi yang sangat spektakuler ini membawa dampak yang sangat luas terhadap kehidupan, yang mana saat ini manusia cenderung dengan hal-hal yang serba praktis, efektif, cepat dan juga ekonomis.

Guna mengantisispasi harapan-harapan diatas, para pakar ilmu khususnya elektronika melakukan berbagai eksperimen dan revolusi. Salah satu revolusi tersebut adalah usaha untuk mentransformasikan segala kegiatan atau proses yang semula berbentuk linear kebentuk yang serba digital. Proses digitalisasi ini mempunyai dampak sangat luas, karena pada

MILIK UPT PERPUSTAKAAN
IKIP PADANG

umumnya proses monitor atau pengamatan terhadap bentuk digital tersebut jauh lebih mudah dan lebih cepat ditanggapi oleh panca indera,dibandingkan dengan bentuk-bentuk yang linear. Sebagai contoh disini dapat diamati pada computer, spidometer kendaraan bermotor, pada tuner radio dan berbagai peralatan kontrol lainnya. Bahkan banyak pakar elektronika meramalkan bahwa pada masa-masa yang akan datang elektronika tersebut identik dengan digital, dan segala sesuatunya nanti akan berlangsung secara digital.

BAB II

ELEKTRONIKA LOGIKA

Proses akhir suatu rangkaian elektronika, pada dasarnya merupakan suatu urutan atau tatanan ytang mengacu pada dua kondisi atau dua keadaan (ya atau tidak, benar atau salah, hidup atau mati dan sebagainya.). Berdasarkan kepada kondisi yang sedemikian sederhana dan logis ini, maka dapat dinyatakan bahwa proses operasi setiap peralatan elektronika adalah logis dan dapat diperhitungkan atau dikontrol. Kelogisan serta kesederhanaan sirkit yang hanya mengarah pada dua kondisi ini,selanjutnya dijadikan dasar berpijak dalam pengembangan dunia elektronika pada umumnya, khususnya teknik digital dan hal inilah yang disebut sebagai elektronika logika

Sebagaimana layaknya logika, dalam elektronika logikapun sebahagian besar permasalahan yang berupa informasi-informasi, baik informasi yang diterima maupun yang dilewatkan terlebih dahulu diformulasikan kedalam bentuk persamaan-persamaan yang benar-benar dapat dilogiskan serta sederhana. Proses formulasi ini sangat penting dilakukan, sehingga dengan penerapan elektronika logika diharapkan menghasilkan suatu penyelesaian dan penganalisaan yang mudah, bahkan dari suatu rangkaian elektronika yang rumit sekalipun.

A. Aljabar Boolean

Permasalahan-permasalahan yang timbul pada elektronika logika, dapat diselesaikan atau dianalisa dengan menggunakan suatu persamaan khusus yang mempunyai kaidah-kaidah tersendiri. Formnulasi ini disusun dalam tata urut yang logis dan biasa dikenal dengan sebutan *aljabar Boole* atau *aljabar saklar* (dapat dilogiskan dengan menggunakan simulasi saklar).

Hal-hal khusus yang perlu diperhatikan pada aljabar Boolean adalah masalah pembacaan tanda matematis, misalnya saja :

- A.B berarti A dan B (bukan A kali B).
- A + B berarti A atau B (bukan A tambah B).
- $\bar{A}$ berarti kebalikan (invert) A atau bukan A.
- Tanda sama dengan (=) berarti ada keluaran atau kondisi saklar tertutup.

B. Hukum-hukum dan Sifat Persamaan Logika

Pada sebahagian persamaan logika juga diterapkan aturan-aturan yang menyerupai kaidah yang terdapat pada aljabar biasa. Walaupun perbedaan tersebut tidak berlaku untuk semua persamaan, namun hal tersebut sangat penting untuk diketahui karena beberapa diantaranya memiliki perbedaan yang cukup mendasar. Untuk lebih jelasnya dapat dilihat contoh berikut ini:

- $\bar{\bar{A}} = A$
- $A \cdot A = A$ (bukan A^2)
- $A \cdot \bar{A} = 0$
- $A + A = A$ (bukan 2A)
- $A + \bar{A} = 1$
- $A + 1 = 1$
- $A + 0 = A$
- Dan sebagainya

Sedangkan sifat serta hukum-hukum persamaan yang digunakan pada aljabar biasa, juga berlaku pada aljabar Boolean, meskipun tidak semuanya. Sehingga dalam hal ini apapun bentuk hukum-hukum aljabar biasa tersebut dapat kita terapkan, namun tidak akan merobah keluarannya. Adapun hukum-hukum tersebut antara lain :

1. **Hukum Komutatif**

   a. $A \cdot B = B \cdot A$.

   b. $A + B = B + A$.

2. **Hukum Distributif**

   a. $A.(B + C) = AB + AC$.

   b. $A + BC = (A + B).(A + C)$.

Pada persamaan a terlihat bahwa persamaan tersebut sama saja halnya dengan persamaan aljabar biasa, akan tetapi pada persamaan b terjadi penyimpangan dari persamaan kuadrat yang lazim dikenal. Perbedaan tersebut terjadi akibat dari

MILIK UPT PERPUSTAKAAN
IKIP PADANG

penerapan beberapa persamaan Boolean. Adapun persamaan b tersebut dapat dibuktikan sebagai berikut :

$$
\begin{aligned}
-(A + B).(A + C) &= A.A + A.C + A.B + B.C. \\
AA + AC + BA + BC &= A + A.c + A.B + B.C \\
A + AC + BA + BC &= A.(1 + C + B) + BC \\
A(1 + C + B) + BC &= \underline{A + B.C}
\end{aligned}
$$

Pada baris pertama pembuktian persamaan terlihat bahwa A.A = A , dan selanjutnya pada baris ketiga persamaan tersebut disederhanakan lagi, dimana pada setiap operasi logika jika salah satu sukunya bernilai 1 (logika 1) maka otomatis hasilnya adalah 1 tanpa dipengaruhi oleh suku lainnya.

$$(1 + A + B\ C + D\ E + A + AC + ABC + \ldots \quad = 1).$$

3. Hukum Asosiatif

a. $A.(B.C) = (A.B).C$

b. $A + (B + C) = (A + b) + C$

4. Sifat Absorbsi Logika

a. $A.(A + B)$

$$
\begin{aligned}
&= A \\
&= A.A + A.B \\
&= A + A.B \\
&= A.(1 + B) \\
&= A
\end{aligned}
$$

b. $A.(A + B) = A.B$

Bukti : $A.(\bar{A} + B) = A.\bar{A} + A.B$

$= A.B$

c. $A + \bar{A}.B = A + B$

Bukti : $A + \bar{A}.B = (A + B) \times 1$

$= (A + B).(A + \bar{A})$

$= A.A + \overset{0}{A.\bar{A}} + A.B + \bar{A}.B$

$= A + A.B + \bar{A}.B$

$= A.(1 + B) + \bar{A}.B$

$= A + \bar{A}.B$

C. Teorema De Morgan

Salah satu terobosan besar dalam perkembangan aljabar Boolean adalah penemuan yang didapatkan oleh Augustus de Morgan. Ketika itu De Morgan bertitik tolak dari pernyataan "Jika dan hanya jika semua masukan adalah benar (1), maka keluarannya adalah benar (1)". Secara logika pernyataan tersebut ekivalen dengan pernyataan "jika salah satu saja masukannya tidak benar (0), maka keluarannya pun tidak benar" (0) (Milman-Halkias ;1985, h. 173).

Secara matematisnya sesuai dengan logika Boolean maka pernyataan De Morgan ini dapat diformulasikan sebagai berikut :

$$A.B.C... = \overline{\bar{A} + \bar{B} + \bar{C} + ...}$$

Apabila pernyataan diatas dituliskan dalam bentuk pertentangan atau komplemen untuk ruas kiri dan ruas kanannya, sehingga didapat :

$$\overline{A.B.C...} = \overline{\overline{A} + \overline{B} + \overline{C} + ...}$$

Karena pada aljabar Boolean A = A, maka :

$$\boxed{\overline{A.B.C...} = \overline{A} + \overline{B} + \overline{C} + ...}$$ Hk.I De Morgan

Dari persamaan di atas terbukti bahwa pada dasarnya proses perkalian (pada ruas kiri persamaan) dapat dinyatakan dalam bentuk lain yaitu penjumlahan (lihat ruas kanan persamaan). Namun perlu diperhatikan bahwa yang dijumlahkan tersebut adalah komplemen masing-masing suku.

Dengan mengikuti prosedur matematis yang sama hal sebaliknya juga dapat terjadi, yang mana proses penjumlahan juga dapat dikonversikan atau dinyatakan dalam bentuk perkalian komplemen suku-sukunya. Secara Booleannya pernyataan tersebut dapat dinotasikan sebagai berikut :

$$\boxed{\overline{A + B + C + ...} = \overline{A} \,.\, \overline{B} \,.\, \overline{C}}$$ Hk.II De Morgan

Dua teori yang dinyatakan oleh De Morgan inilah yang merupakan sumbangan besar dalam teknik digital khususnya dalam merancang dan menganalisa

rangkaian logika. Untuk aplikasinya hal ini dapat dilihat pada pembahasan bab-bab berikutnya.

MILIK UPT PERPUSTAKAAN
IKIP PADANG

BAB III

GERBANG LOGIKA

Sebagaimana yang telah dijelaskan pada bab terdahulu, bahwa pada elektronika logika hanya dikenal dua kondisi. Kedua kondisi ini dilukiskan sebagai dua bilangan atau konstanta yaitu 0 dan 1 (biner). Penjabaran kedua logika ini kedalam bentuk bilangan biner sangat penting sekali artinya, karena kedua kondisi tersebut selanjutnya akan diolah, dikembangkan serta diformulasikan kedalam suatu bentuk rangkaian yang menggunakan gerbang-gerbang atau pintu. Gerbang atau pintu ini merupakan rangkaian digital dua keadaan dengan satu atau lebih masukan, tetapi hanya mempunyai satu keluaran.(Malvino ;1988, h.23).

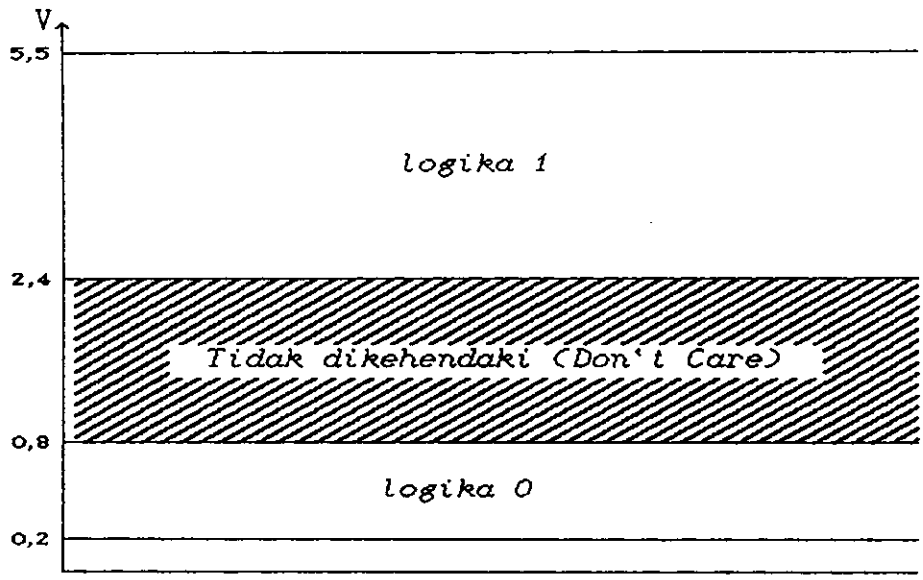
Didalam penulisan lambang logika atau simbol dari gerbang tersebut dikenal dua bentuk simbol; *pertama* mengikuti standar yang ditetapkan oleh IEC (International Electro- technical Commission), *kedua* berdasarkan standar Amerika. Kebanyakan simbol yang digunakan pada buku ini adalah simbol IEC, namun demikian standar Amerika tetap kita gunakan karena sebagian besar lembaran data rangkaian terpadu (IC Data-Sheets) menggunakan simbol Amerika tersebut.(Per Carlsson ;1981, h.60).

A. Taraf Tegangan

Pada ujud sebenarnya konstanta 0 dan 1 adalah merupakan suatu tingkatan tegangan atau taraf tegangan

(voltage level). Berdasarkan fasa pulsanya, taraf tegangan itu sendiri diklasifikasikan atas dua logika yaitu logika positif dan logika negatif.

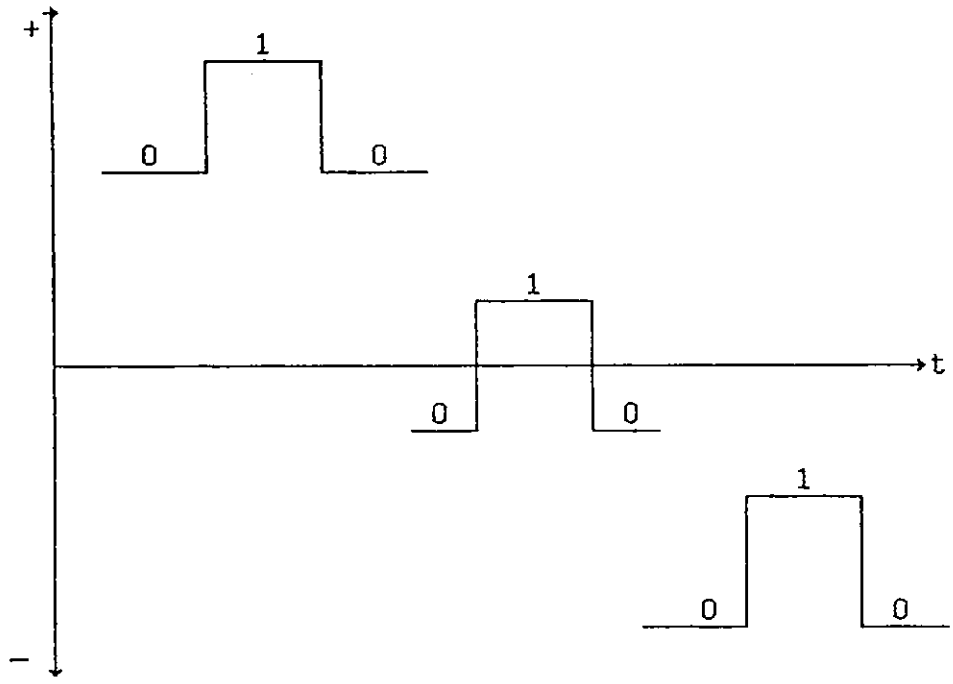
Nilai dari besar kecilnya taraf tegangan, pada umumnya bervariasi untuk produk TTL (Transistor-Transistor Logic) yang berbeda. Sebagai contoh TTL seri 74xx (untuk keperluan industri), memiliki rentang 4,75 sampai 5,25 V_{DC} untuk logika 1, sedangkan pada seri 54xx (untuk keperluan militer), memiliki rentang 4,5 V_{DC} hingga 5,5 V_{DC} buat logika 1 (Arthur B. Williams ;1984, h.9-2a.). Akan tetapi secara globalnya dapat dilukiskan sebagai berikut :



Gambar 3.1 Taraf Tegangan
(F. Suryatmo ;1986, h.113)

1. Logika Positif

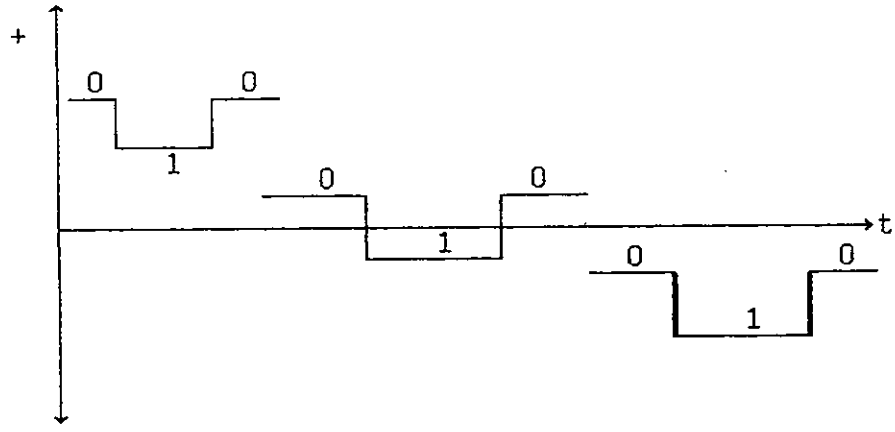
Logika positif pada elektronika logika didasarkan kepada taraf tegangan yang lebih tinggi, yang mana taraf tinggi tersebut dianggap sebagai logika 1. Sebaliknya pada logika positif ini taraf rendah dianggap sebagai logika 0. Dengan demikian taraf tegangan ini tidak memandang apakah tegangan tersebut positif ataukah negatif. Untuk lebih jelasnya dapat diamati gambar berikut ini :



Gambar 3.2. Logika Positif
(Gatot Sudarto ;1980, h.77)

2. Logika Negatif

Kondisi sebaliknya dari logika positif adalah logika negatif. Jika suatu taraf tegangan berada pada kondisi lebih rendah, maka akan dipandang sebagai logika 1, sedangkan yang lebih tingginya akan dipandang sebagai logika 0.



Gambar 3.3 Logika Negatif
(Gatot Sudarto ;1980, h.78)

B. Gerbang Dasar

Semua permasalahan-permasalahan atau perencanaan dalam elektronika logika, dianalisa dan diselesaikan dalam bentuk rangkaian gerbang atau pintu. Banyak sekali model dan jenis rangkaian pintu, namun pada bagian berikut ini akan dibahas beberapa gerbang yang merupakan gerbang dasar.

MILIK UPT PERPUSTAKAAN
IKIP PADANG

1. **Gerbang And**

Gerbang And terdiri dari n masukan (n >1) dan hanya memiliki satu keluaran. Persamaan keluaran dari gerbang And jika diformulasikan merupakan suatu operasi perkalian Boolean yang mana :

F = A . B (untuk dua masukan)

Dimana : F = Keluaran

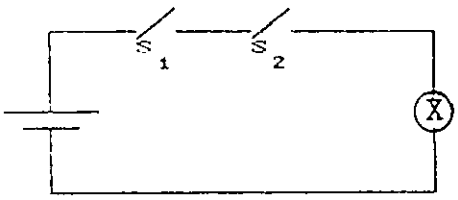
A dan B = Masukan

Menurut persamaan diatas maka dapat dikatakan bahwa pada gerbang And kondisi keluaran akan selalu rendah (Logika 0), jika salah satu saja dari masukannya berada pada taraf rendah (logika 0). Sebaliknya logika 1 hanya terjadi apabila semua masukannya bertaraf tinggi (logika 1).

Berdasarkan persamaan diatas maka gerbang And tersebut dapat kita pandang sebagai operasi saklar yang dihubungkan secara seri pada gambar 3.4a, dan dilambangkan sebagaimana gambar 3.4b.

Pada gambar 3.4 (a) lampu F hanya akan menyala (logika 1) apabila semua saklar (A dan B) ditutup atau diberikan tegangan (logika 1). Jika hanya salah satu atau satu saja saklar tersebut yang ditutup, maka lampu akan tetap mati (logika 0).

Selanjutnya pada gambar 3.4b adalah simbol yang digunakan untuk gerbang Or dan gambar 3.4c merupakan tabel kebenaran untuk semua variasi masukan gerbang Or dua masukan.



a. Konstruksi

b. Simbol

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

c. Tabel Kebenaran

Gambar 3.4 Gerbang AND
(Arthur B.Williams ;1984,h. 9-11)

2. Gerbang OR

Gerbang Or dikatakan juga sebagai gerbang alternatif dari kondisi masukan terhadap keluaran. Menurut Boole gerbang Or ini dapat diformulasikan sebagai operasi penjumlahan (penjumlahan Boolean), atau :

$$F = A + B + C + \ldots + n$$

dimana : F = Keluaran

$$\left.\begin{matrix} A \\ B \\ C \\ n \end{matrix}\right\} = \text{Masukan.}$$

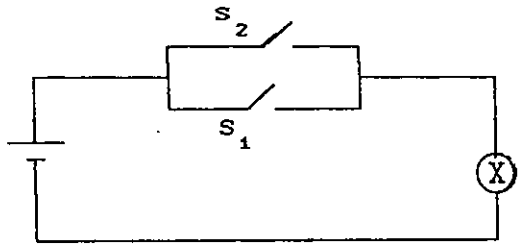
Berdasarkan aljabar Booleannya maka persamaan diatas dapat diartikan bahwa kondisi keluaran akan selalu logika 1, apabila salah satu atau semua masukan logika 1 (1 + A = 1). Sebaliknya keluaran (F) logika 0, hanya terjadi jika semua masukan gerbang tersebut adalah logika 0. Berdasarkan aljabar Booleannya maka persamaan diatas dapat diartikan bahwa kondisi keluaran akan logika 1, apabila salah satu atau semua masukannya berlogika 1 (1 + A + ...+ n = 1). Sebaliknya logika 0 pada keluaran hanya terjadi apabila semua masukannya berlogika 0.

Secara saklarnya pernyataan atau uraian tersebut dapat dilukiskan sebagai hubungan saklar yang terpasang paralel. Untuk lebih jelasnya dapat diamati gambar 3.5.

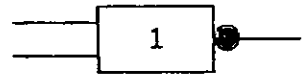
Pada gambar 3.5 ini, sama saja halnya dengan gambar-gambar terdahulu yang mana disini diberikan lengkap mulai dari sistem saklar rangakaian (gambar 3.5a), berikut lambang atau simbol (gambar 3.5b) dan seterusnya tabel kebenarannya (gambar 3.5c).

1784/Hd/94-P2(2)

621.381
tas
P2



a. Saklar



b. Simbol

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

b. Tabel Kebenaran

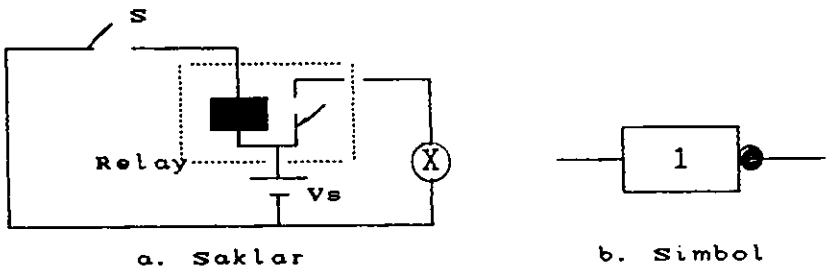
Gambar 3.5 Gerbang OR
(Millman-Halkias ;1985, h.159)

3. Gerbang Inverter

Gerbang Inverter biasa juga disebut dengan gerbang Not. Sebagaimana namanya gerbang ini digunakan sebagai gerbang pembalik atau penjungkir, sehingga gerbang ini akan selalu menghasilakn keluaran yang bertentangan dengan masukannya (komplemen keluaran). Karena gerbang ini hanya terdiri dari satu masukan dan satu keluaran, maka formulasi Booleannya dapat dituliskan menjadi $A = \overline{A}$ atau $B = \overline{B}$ dan seterusnya (A adalah masukan, $\overline{A}$ adalah keluaran).

MILIK UPT PERPUSTAKAAN
IKIP PADANG

Rangkaian ekivalen dari gerbang Inverter atau Not, dapat diamati pada gambar 3.7 berikut :



Gambar 3.7 Gerbang Inverter
(Gatot Sudarto ;1980, h.79)

4. Gerbang NAND

Gerbang Nand adalah gerbang yang terdiri dari beberapa masukan (n masukan), dan hanya memiliki satu keluaran. Gerbang ini dapat dibentuk dengan menggandengkan keluaran gerbang And dengan Inverter, sehingga dapat dikatakan bahwa gerbang Nand tersebut merupakan operasi penjungkiran atau komplementasi dari gerbang And. (NAnd =========> Not And).

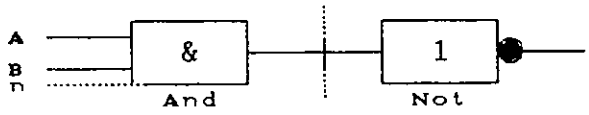
Berdasarkan gambar 3.8 diatas, maka persamaan keluaran gerbang ini dapat didefinisikan menjadi :

$$F = \overline{A \cdot B \cdot n}$$

dimana : F = Keluaran.

$$\left.\begin{matrix} A \\ B \\ n \end{matrix}\right\} = \text{Masukan.}$$

Dari persamaan dan analisa terhadap rangkaiannya maka didapat suatu kesimpulan, bahwa logika 0 pada keluaran gerbang ini hanya terjadi pada saat semua masukannya berlogika 1, sedangkan logika 1 pada keluaran terjadi apabila salah satu atau semua masukannya berlogika 0.



a. Konstruksi

b. Simbol

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

c. Tabel Kebenaran

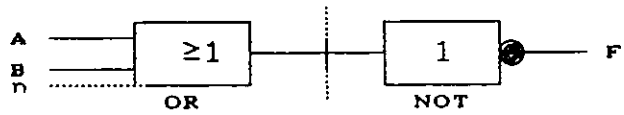
Gambar 3.8 Gerbang NAND
(Millman-Halkias ;1985, h.175)

5. Gerbang NOR

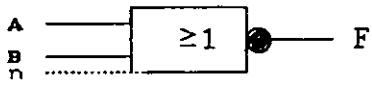
Sama halnya dengan gerbang Nand, gerbang Nor juga merupakan gerbang kombinasi antara OR dengan gerbang Inverter. Dengan demikian maka keluaran gerbang ini merupakan operasi penjungkiran atau komplemen dari gerbang OR.

UPT PERPUSTAKAAN
PADANG

Berdasarkan gambar 3.9, maka persamaan keluaran dari gerbang ini dapat diformulasikan menjadi $F = \overline{A + B}$. Sedangkan keluaran gerbang NOR ini hanya akan berlogika 1 apabila semua variabel masukannya berlogika 0, sebaliknya apabila salah satu atau semua masukannya berlogika 1, maka keluarannya akan selalu berlogika 0.



a. Konstruksi



b. Simbol

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

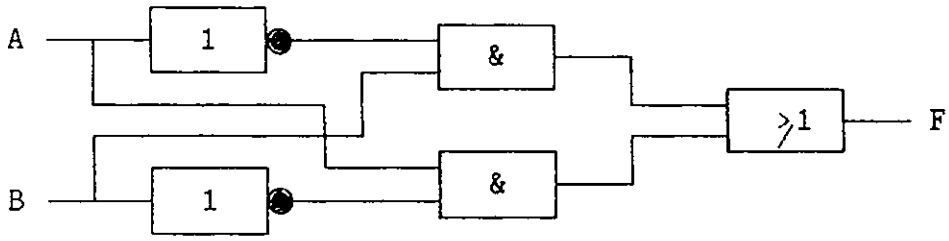
c. Tabel Kebenaran

Gambar 3.9 Gerbang NOR
(Börje Lindell ;1981, h.35)

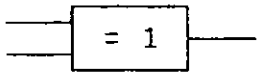
6. Gerbang Exclusive-OR

Exclusive-OR biasa disingkat dengan Ex-OR atau dituliskan dengan XOR saja. Gerbang ini terdiri dari

beberapa masukan dan hanya memiliki satu keluaran. Secara Konstruksinya gerbang ini dibentuk dari kombinasi antara gerbang AND, OR, dan gerbang NOT.



a. Konstruksi



b. Simbol

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

c. Tabel Kebenaran

Gambar 3.10 Gerbang XOR
(Per Carlsson ;1981, h. 56)

Gambar 3.10 merupakan salah satu contoh dari gerbang XOR yang terdiri dari dua masukan. Jika analisa terhadap masukan diselesaikan, maka persamaan keluaran dari XOR tersebut menjadi :

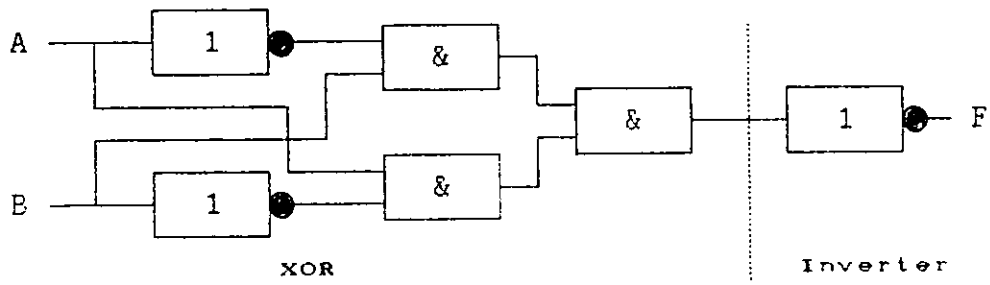
$F = \overline{A}B + A\overline{B}$ biasa dituliskan dengan $F = A \oplus B$.

Berdasarkan persamaan keluaran ini dan juga tabel kebenaran diatas, dapat disimpulkan bahwa keluaran (F) XOR akan berlogika 1 apabila salah satu saja dari masukannya berlogika 1. Jika terdapat lebih dari satu masukan tersebut yang berlogika 1,

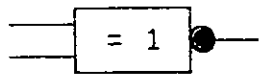
atau semua masukannya berlogika 0, maka keluarannya juga akan berlogika 0.

7. Gerbang Exclusive-NOR

Sama halnya dengan gerbang XOR, gerbang ini biasa disingkat saja dengan XNOR. Karena operasi dari gerbang Nor adalah operasi penjungkiran atau pembalikan dari gerbang Or, maka demikian juga halnya dengan XNor juga merupakan operasi pembalikan dari XOR.Dengan demikian dapat didefinisikan bahwa XNOR adalah XOR yang keluarannya dipasang gerbang Inverter.



a. Konstruksi



b. Simbol

A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

c. Tabel Kebenaran

Gambar 3.11 Gerbang XNOR
(Per Carlsson ;1981, h. 56)

Berdasarkan konstruksi dan tabel kebenarannya, maka notasi keluaran dari XNOR tersebut dapat di ekspresikan menjadi :

$$F = \overline{A + B}$$

$$= \overline{\bar{A}.B + A.\bar{B}}$$

Dengan mengaplikasikan teorema De Morgan dan beberapa persamaan-persamaan Boole lainnya, maka persamaan diatas dapat disederhanakan sebagai berikut :

$$F = \overline{\bar{A}.B + A.\bar{B}}$$

$$= \overline{\bar{A}.B} \,.\, \overline{A.\bar{B}}$$

$$= (A + \bar{B}) (\bar{A} + B)$$

$$= A.\bar{A} + A.B + \bar{A}.\bar{B} + B.\bar{B} \longrightarrow A.\bar{A} = 0$$

$$= \underline{A.B + \bar{A}.\bar{B}}$$

BAB IV

MODIFIKASI DAN SIMPLIFIKASI RANGKAIAN LOGIKA

Kendala yang paling sering muncul didalam merakit dan menganalisa suatu rangkaian logika atau rangkian gerbang adalah dalam hal konversi. Konversi merupakan suatu proses vital yang memungkinkan para peminat atau perancang untuk melakukan analisa, memodifikasi dan menyederhanakan suatu rangkaian logika.

Keberhasilan didalam melakukan konversi memang merupakan suatu tahapan yang menentukan. Akan tetapi hanya dengan mengandalkan konversi, tidaklah menjamin untuk didapatkan suatu rangkaian atau persamaan yang betul-betul efisien. Efisiensi ini terutama sekali ditandai dari segi kuantitas gerbang yang digunakan. Rangkain yang terdiri dari banyak gerbang merupakan suatu pemborosan dan akan menghasilkan suatu persamaan keluaran yang relatif lebih rumit dan sukar untuk dimengerti, bahkan kadang juga sukar untuk diaplikasikan.

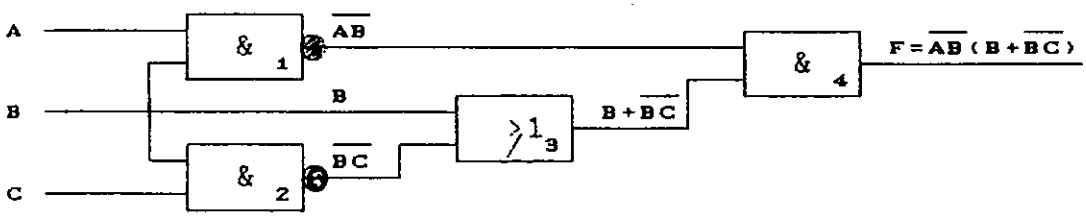
Untuk itu pengetahuan tentang konversi itu sendiri juga sangat perlu ditunjang atau dilengkapi dengan pengetahuan tentang metoda-metoda modifikasi dan simplifikasi rangkaian dan persamaan.

A. Konversi

Proses konversi yaitu suatu proses terjemahan atau pembalikan dari suatu bentuk (*Convert*). Ada dua proses konversi yang paling umum dilakukan.

UPT PERPUSTAKAAN
IP PADANG

Pertama, yaitu konversi dari suatu rangkaian gerbang menjadi persamaan logika. Proses ini tidak terlalu sulit dilakukan, untuk ini hanya dituntut pengetahuan dasar tentang gerbang-gerbang logika. Proses ini dimulai dari masukan rangkaian kemudian diteruskan penelusuran hingga keluaran rangkaian atau gerbang terakhir. Sebagai contoh dapat diamati gambar 4.1 berikut :



Gambar 4.1 Konversi

Amati gambar 4.1, Gerbang 1 adalah gerbang Nand dengan masukan A dan B, sehingga keluarannya adalah $\overline{AB}$ (komplemen AB), sedangkan gerbang 2 juga gerbang Nand dengan masukan BC, maka keluarannya adalah $\overline{BC}$. Gerbang 3 adalah OR, yang mempunyai masukan berasal dari keluaran gerbang 2 dan B maka keluarannya menjadi B + $\overline{BC}$. Gerbang 4 masukannya berasal dari keluaran gerbang 1 dan gerbang 3, gerbang ini adalah gerbang AND dan merupakan gerbang terakhir dengan keluaran $\overline{AB}$ (B + $\overline{BC}$), yang mana keluaran dan persamaan ini merupakan persamaan keluaran rangkaian keseluruhan. Proses penelusuran persamaan, yang dilakukan mulai dari masukan gerbang 1 dan seterusnya hingga keluaran gerbang 4 inilah yang disebut dengan proses konversi.

Proses konversi *kedua* yaitu, konversi dari persamaan Boolean ke rangkaian logika. Bagi pemula proses ini kadang merupakan kesulitan tersendiri, kendala tersebut biasanya timbul akibat kekeliruan dalakibat kekeliruan melakukan langkah proses. Untuk persamaan-persamaan sederhana yang hanya menggabungkan beberapa gerbang, memang proses tersebut tidak merupakan masalah, karena pembuataan rangkaian logikanya dapat dimulkai dari mana saja (dapat dimulai dari masukan, maupun keluaran). Akan tetapi bagi suatu persamaan yang panjang dan rumit, dimana biasanya merupakan kombinasi dari banyak tanda yang bervariasi atau gerbang yang digunakan bervariasi maka proses atau langkah awal kerja harus diatur sedemikian. Misalnya pada persamaan sebagai berikut :

$$F = \overline{\{ \overline{(\overline{A.B}) + C.\overline{D}} \}.\{ \overline{(\overline{C + D}).(\overline{A + B})} \}}$$

Persamaan di atas yang cukup panjang jika diamat-amati saja, lalu dicoba-coba menyelesaikan secara gamblang atau acak, maka hal ini akan memboroskan waktu perencanaan, dengan hasil yang cenderung keliru. Disamping itu dengan cara menerka model penyelesaian memang masih ada kemungkinan benar, namun satu hal yang sulit sekali dihindarkan adalah suatu bentuk rangkaian yang rumit sekali, dengan pengawatan yang tumpang tindih dan simpang siur.

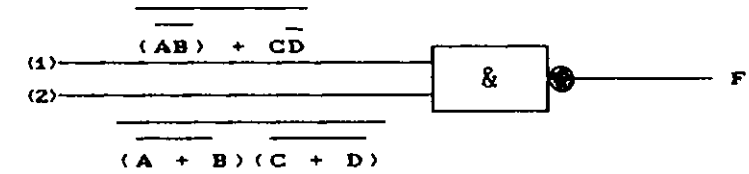
Untuk menghindarkan kerumitan dan kecenderungan salah tersebut, maka perlu diikuti prosedur yang tepat dan praktis. Adapun cara tersebut yaitu, *memulai disain rangkaian dari keluaran rangkaian, dan gunakan gerbang berdasarkan tanda-tanda setiap persamaan.* Sebagai contoh penyelesaian tersebut dapat dilihat tata urut prosedur dibawah berikut. (Sebagai contoh diambil persamaan diatas).

1. Pada persamaan diatas, tanda yang paling dominan ialah tanda yang memisahkan antara kedua kurung kurawal ({}), dalam hal ini tanda titik (gerbang And) yang dikombinasi dengan garis panjang diatas, sehingga secara keseluruhan dalam hal ini berarti adalah NOT-AND (NAND), lihat gambar 4.2a.

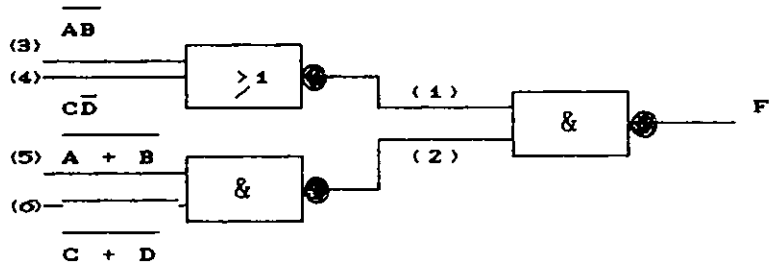
2. Berikutnya kita selesaikan masukan 1 dan 2. Pada masukan 1, merupakan kombinasi antara tanda tambah dengan garis lurus (NOT-OR ====> NOR). Pada masukan dua adalah kombinasi antara titik dengan garis lurus (NAND). Dengan demikian disain tersebut dapat dilanjutkan sebagaimana gambar 4.2b.

3. Langkah selanjutnya kita selesaikan untuk setiap masukan tersebut yang mana pada :
   - masukan 3 adalah gerbang NAND.
   - Masukan 4 adalah gerbang AND dengan masukan D diinverter.
   - Masukan 5 dan 6 adalah gerbang NOR.

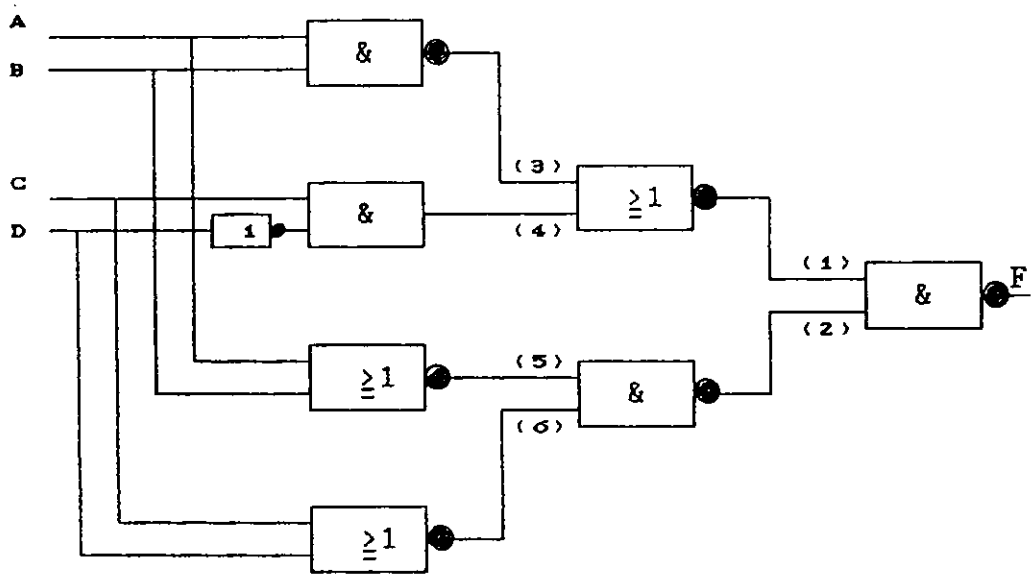
Sehingga gambar akhir dari perencanaan tersebut adalah sebagai mana yang tampak pada gambar 4.2c.



(a)



(b)



(c)

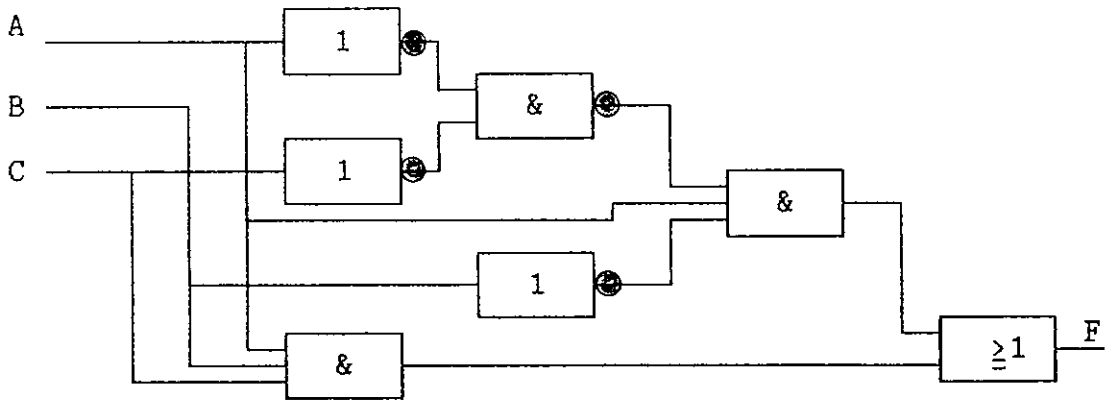
Gambar 4.2
Proses Konversi
(Ronald J. Tocci;1980, h.)

B. Simplifikasi

Simplifikasi adalah suatu proses penyederhanaan (Simplify) persamaan logika. Tujuan utama penyederhanaan adalah untuk mendapatkan suatu bentuk rangkaian yang paling sederhana. Kesederhanaan tersebut mutlak diperlukan dalam suatu rangkaian logika, karena semakin sederhana rangkaian, berarti semakin sedikit gerbanng yang diperlukan, dan otomatis akan menurunkan biaya pembuatan.

Pada umumnya proses simplifikasi dilakukan dengan menerapkan kaidah-kaidah Boolean, sebagaimana yang sudah dibahas pada bab II. Sebagai ilustrasi dapat dilihat contoh-contoh berikut ini :

Contoh 1:



Gambar 4.3

Simplifikasi

(Ronald J. Tocci :1980, h. 44)

Penyederhanaan atau simplifikasi dari persamaan diatas dapat dimulai dari keluaran persamaan $\{ F = ABC + A\overline{B} . (\overline{\overline{A}.\overline{C}})\}$. Persamaan keluaran ini dapat dipecah-pecah atau diuraikan melalui penyelesaian pada kelompok yang ada dalam kurung terlebih dahulu ($\overline{\overline{A}.\overline{C}}$), yang mana bagian ini dapat diselesaikan dengan menggunakan aturan De Morgan. Selanjutnya dapat diikuti penyelesaian tersebut sebagai berikut :

$$
\begin{aligned}
F &= ABC + A\overline{B}.(\overline{\overline{A}.\overline{C}}) \\
&= ABC + A\overline{B}.(\overline{\overline{A}} + \overline{\overline{C}}) \\
&= ABC + A\overline{B}.(A + C) \\
&= ABC + A\overline{B}A + A\overline{B}C \\
&= ABC + A\overline{B} + AB\overline{C}
\end{aligned}
$$

Berdasarkan hukum Distributif (lihat Bab II), maka persamaan diatas dapat disederhanakan lagi menjadi :

$$F = A.C (B + \overline{B}) + A\overline{B}$$

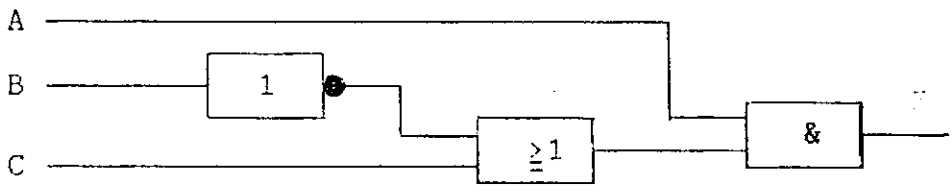
karena $B + \overline{B} = 1$, maka :

$$F = A.C + A.\overline{B}$$

Dengan menerapkan sifat Absorbsi Logika (lihat Bab II), maka persamaan akhir tersebut dapat ditulis sebagai berikut :

$$F = A. (C + \overline{B})$$

Jika persamaan tersebut dikonversikan kedalam rangkain logika, maka didapat rangkaian sebagaimana gambar 4.4, yang merupakan gambar rangkaian hasil dari simplifikasi tersebut, dan penyederhanaan dari gambar 4.3 sebelumnya.



Gambar 4.4

Simplifikasi Persamaan

Contoh 2 :

Sederhanakanlah persamaan berikut ini :

$$Y = PQR + PQ\overline{R} + P\overline{Q}R$$

Untuk penyelesaian persamaan diatas dapOat dilakukan dengan dua cara atau metoda.

Metoda I :

$$Y = PQR + PQ\overline{R} + P\overline{Q}R$$

$$= PQ (R + \overline{R}) + P\overline{Q}R$$

$$= PQ + P\overline{Q}R$$

dengan melakukan faktorisasi pada variabel P, selanjutnya persamaan ini dapat ditulis menjadi :

$$Y = P (Q + \overline{Q}R)$$

Berdasarkan sifat absorbsi logika (bab II), maka simplifikasi akhir dari persamaan diatas menjadi :

$$Y = P (Q + R)$$

Metoda II :

Metoda kedua ini dilakukan dengan menerapkan hukum Distributif dari awal penyelesaiannya. Pada persamaan diatas suku-suku pertama dan kedua memiliki variabel PQ, sehingga persamaan tersebut dapat disimplifikasikan menjadi :

$$\begin{aligned} Y &= PQR + PQ\overline{R} + P\overline{Q}R \\ &= PQ (R + \overline{R}) + P\overline{Q}R \\ &= PQ + P\overline{Q}R \quad ========> \quad R + \overline{R} = 1 \end{aligned}$$

Selanjutnya suku-suku diatas memiliki variabel P, sehingga didapat :

$$\begin{aligned} Y &= P (Q + \overline{Q}R) \\ &= P (Q + R) \end{aligned}$$

C. Perencanaan Logika Rangkaian

Pengetahuan tentang jenis-jenis gerbang logika, tabel kebenaran, dan aljabar Boolean serta konversi, merupakan suatu dasar yang sangat vital didalam merancang suatu rangkaian logika. Berikut ini akan dijelaskan beberapa metoda dalam mendisaian rangkaian logika, khususnya rangkaian kombinasi gerbang logika. Jadi dalam sub bab ini tidak akan dibahas masalah yang menyangkut rangkaian umpan balik (Feedback), memory dan sebagainya.

Kebutuhan terhadap rangkaian logika umumnya bertitik tolak atau ditentukan oleh kebutuhan. Sehingga dalam hal ini yang terjadi tidak hanya sekedar memodifikasi rangkaian kebentuk tertentu, namun seringkali yang terjadi adalah bagaimanacaranya merencanakan atau mendisaian suatu rangkaian baru sesuai dengan kebutuhan tersebut.

Alur kerja yang diperlukan untuk mendisaian suatu rangkain baru tersebut mungkin saja beragam untuk setiap disainer yang berbeda. Namun alur yang biasa dilakukan adalah dimulai dari logika berfikir terhadap keadaan yang ditemui.

Proses logika tersebut umumnya diawali dengan merumuskan semua permasalahan, dengan menyusun logika keinginan atau kebutuhan tersebut menjadi suatu bentuk yang saling kontradiksi (dua keadaan yang saling berlawanan). Rumusan logika yang sudah direncanakan tersebut selanjutnya disusun dalam bentuk tabel yang dikenal dengan tabel kebenaran.

Perumusan masalah yang diungkapkan dalam tabel kebenaran tersebut, selanjutnya diekspresikan dalam bentuk bilangan biner (bilangan yang mampu mengindentifikasikan dua kondisi yang saling berlawanan). Dengan demikian tabel kebenaran ini merupakan tabel yang menghimpun data-data dan merupakan kesimpulan terhadap informasi yang diterima. Hal ini dapat dikatakan demikian, karena tersebut tidak hanya berisikan kumpulan

MILIK UPT PERPUSTAKAAN
IKIP PADANG

data, tapi juga sekaligus menjelaskan hubungan antara masukan dan keluaran.

Sebagai ilustrasi dari rumusan tabel kebenaran yang bertitik tolak dari latar belakang masalah tersebut, dapat kita lihat contoh-contoh berikut ini :

Contoh 1 :

a. Permasalahan

Sebuah ruangan rahasia digunakan untuk menyimpan berkas-berkas penting suatu perusahaan (kantor). Ruangan ini dilengkapi dengan kunci elektronika yang hanya dapat dibuka oleh orang-orang tertentu, misalnya A, B, dan C. Untuk langkah pengamanan ketiga orang tersebut tidak dapat membuka pintu ruangan seorang diri. Dengan demikian pintu ini hanya dapat dibuka apabila dilakukan oleh lebih satu orang atau ketiganya secara bersamaan memasukan kuncinya secara bersamaan. Rancanglah rangkaian logika yang dapat memenuhi persyaratan diatas !

b. Rumusan Masalah

Dalam hal ini yang menjadi keluaran rangkaian (F) adalah pintu ruangan.

- Pintu terbuka = Logika 1
- Pintu tertutup = Logika 0

Sebagai masukan adalah kunci pintu.

- Kunci dimasukan = logika 1
- Kunci tidak dimasukan = Logika 0

Alternatif-alternatif atau variasi masukan yang mungkin terjadi pada setiap proses memasukan kunci, dan kondisi yang timbul pada pintu:

- A memasukan kunci, pintu tertutup.
- B memasukan kunci, pintu tertutup.
- C memasukan kunci, pintu tertutup.
- A dan B memasukan kunci, pintu terbuka.
- A dan C memasukan kunci, pintu terbuka.
- B dan C memasukan kunci, pintu terbuka.
- A, B dan C memasukan kunci, pintu terbuka.

c. Tabel Kebenaran

Tabel 4.1. Tabel Ruangan Rahasia

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

D. Perencanaan Rangkaian Logika Berdasarkan Tabel

Ada beberapa metoda dalam merencanakan rangkaian logika, yang mana setiap metoda tersebut tergantung dengan kondisi persyaratan atau informasi yang diberikan (Tocci;1980, h. 55).

Informasi yang diberikan dalam merencanakan rangkaian logika tersebut, lazimnya ditampung dalam bentuk tabel kebenaran. Oleh karena itu dengan adanya tabel ini, maka dengan otomatis rangkaian logikanya sudah dapat diujudkan, melalui persamaan keluaran atau ekspresi Boole yang diadapat dari tabel tersebut.

Bentuk umum dari ekspresi Boolean ini pada dasarnya dapat diklasifikasikan kedalam dua bagian (Term), yaitu Bentuk Penjumlahan (Sum) dan Bentuk Perkalian (Product). Seringkali kedua bentuk atau kelompok tersebut dalam suatu persamaan sama-sama digunakan, misalnya :

$$F = \underbrace{A.B.C}_{I} + \underbrace{\bar{A}.B.C}_{II} + \underbrace{B.C}_{III}$$

Pada persamaan diatas "term" I, II, dan III adalah bentuk perkalian antar variabel (Variabel A, B, C, dan $\bar{A}$). Sedangkan ketiga term itu sendiri nantinya dilakukan penjumlahan. Sehingga secara keseluruhan (F) persamaan tersebut dapat dikatakan sebagai "Jumlah dari hasil kali-hasil kali (Sum of Products)". Adakalanya persamaan Boolean juga diekspresikan sebagai "Perkalian dari Hasil jumlah-hasil jumlah (Product of Sums)", sebagaimana yang dapat kita lihat pada contoh berikut ini :

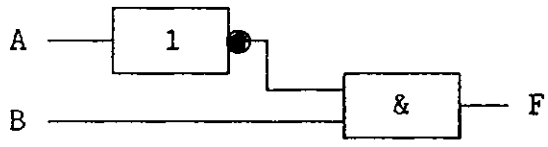
$$F = \underbrace{\underbrace{(A + B + C)}_{I}.\underbrace{(A + \bar{B})}_{II}.\underbrace{(B + \bar{C})}_{III}}_{IV}$$

dimana:

- I, II, dan III : Penjumlahan antar variabel (Sum).
- IV .....: Perkalian dari hasil jumlah-hasil Jumlah. (Product of Sums).

a. Jumlah dari Hasil Kali-hasil kali (Sum of Products)

Pada awal sub bab ini sudah dijelaskan bahwa metoda ini merupakan bentuk kombinasi dari proses penjumlahan dengan proses perkalian. Merajuk kepada apa yang sudah dilengkapi pada bab terdahulu, sebetulnya hal ini sudah diuraikan secara implisit. Sebagai contoh dapat kita lihat rangkaian berikut ini:



A	B	F
0	0	0
0	1	1
1	0	0
1	1	0

Gambar 4.5
Sum of Products
(RJ Tocci; 1980, h.55)

Gambar a menunjukan rangkaian yang terdiri dari dua masukan (A dan B) dengan satu keluaran (F).

Titik tolak proses ini terutama sekali dilakukan dengan cara mengamati atau menggaris bawahi *hanya*

MILIK UPT PERPUSTAKAAN
IKIP PADANG

untuk setiap keluaran (F) pada tabel kebenaran yang memiliki nilai logika 1. Tabel kebenaran yang ada pada gambar b menunjukan bahwa keluaran rangkaian ini akan berlogika 1, hanya pada saat masukan A = 0, dan B = 1. Kondisi A = 0 ini akan sama dengan $\bar{A}$ = 1, sehingga secara matematis uraian diatas dapat kita nyatakan sebagai berikut :

"Keluaran F = 1, *hanya jika* $\bar{A}$ = 1 *dan* B = 1"

atau

$$F = \bar{A}.B$$

Suatu hal yang patut digaris bawahi pada pernyataan diatas adalah bahwa operasi tersebut merupakan operasi gerbang AND (Operasi perkalian logika / product), dimana keluaran gerbang ini hanya akan berlogika 1 apabila kedua masukannya (Semua masukan, untuk jumlah masukan yang lebih dari dua) berlogika 1. Dengan demikian apabila ada variabel masukan yang berlogika 0, maka variabel tersebut harus dikomplementasikan terlebih dahulu. (misalnya pada contoh diatas A = 0 ======> dikomplementasikan menjadi $\bar{A}$ = 1).

Adakalanya tabel kebenaran tersebut tidak hanya memiliki satu keluaran yang berlogika 1, sebagaimana

contoh diatas, namun seringkali lebih misalnya saja tabel 4.2 berikut ini :

Tabel 4.2. Tabel dengan Keluaran 1 Jamak

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

Pada tabel diatas ada dua kemungkinan masukan yang menyebabkan keluaran akan logika 1. Pernyataan matematis yang tepat untuk tabel diatas adalah :

- " Keluaran F = 1, hanya jika $\bar{A}$ = 1 dan B = 1 "

atau

- " Keluaran F = 1, hanya jika A = 1 dan $\bar{B}$ = 1 "

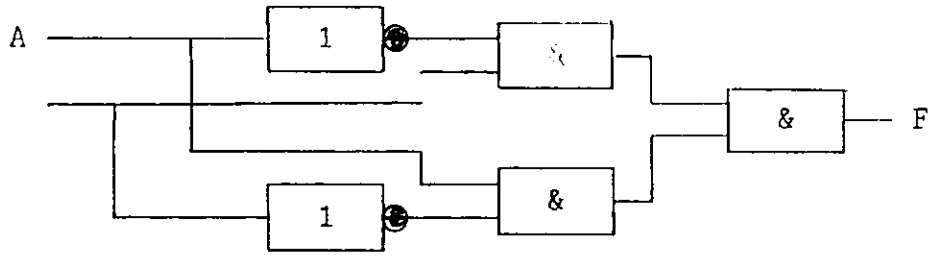
Selanjutnya persamaan ini kita formulasikan menjadi persamaan Boolean, dimana *dan* adalah operasi *perkalian (product)*, serta *atau* adalah operasi *penjumlahan (Sum)*, sehingga didapat persamaan sebagai berikut :

$$F = \bar{A}.B + A.\bar{B}$$

Dari persamaan diatas maka selanjutnya akan dapat dengan mudah dibuatkan gambar rangkaian logika

sebagaimana gambar 4.6.

Ilustrasi lain dari metoda ini dapat dicobakan pada tabel kebenaran yang kita dapatkan berdasarkan rumusan masalah pada awal bab ini (sub bab c, contoh 1) sebagaimana tabel 4.3 .



Gambar 4.6
Hasil Simplifikasi

Tabel 4.3. Simplifikasi

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Berdasarkan tabel 4.3, keluaran yang berlogika 1, yaitu pada baris 4, 6, 7, dan 8. Sehingga pernyataan

yang dapat dibuat dari tabel diatas adalah :

- Baris 4, F = 1, hanya jika $\overline{A}$ = 1, B = 1, dan C = 1
- Baris 6, F = 1, hanya jika A = 1, $\overline{B}$ = 1, dan C = 1
- Baris 7, F = 1, hanya jika A = 1, B = 1, dan $\overline{C}$ = 1
- Baris 8, F = 1, hanya jika A = 1, B = 1, dan C = 1

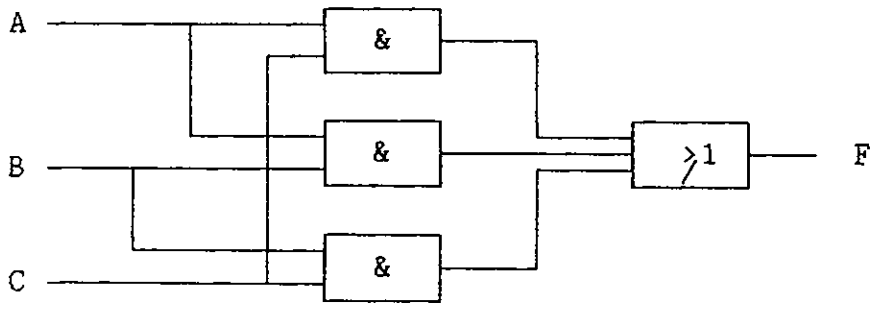
Persamaan Bolean dari pernyataan-pernyataan diatas :

$$F = \overline{A}.B.C + A.\overline{B}.C + A.B.\overline{C} + A.B.C$$

Penyederhanaan atau simplifikasi dari persamaan diatas akan mennghasilkan :

$$F = A.B. + A.C + B.C$$

Sehingga gambar rangkaian dari tabel diatas akan menjadi sebagai berikut :



Gambar 4.7
Tabel dan Simplifikasi

Dari semua uraian , contoh-contoh dan ilustrasi diatas, maka secara umum dapat kita simpulkan langkah-langkah yang dilakukan pada metoda "Sum of Products" yaitu :

- Garis bawahi semua tabel kebenaran yang mempunyai nilai logika 1.
- Susun setiap baris tabel yang berlogika 1 tersebut dalam bentuk persamaan gerbang AND (dalam bentuk perkalian / product).
- Jumlahkan masing-masing bentuk perkalian tersebut (Sum).

Hasil akhir yang didapat setelah memlalui ketiga proses diatas tersebut inilah yang merupakan persamaan akhir dari tabel tersebut. Sebagai catatan adakalanya persamaan yang didapat tersebut masih perlu disederhanakan atau disimplifikasikan lebih dahulu, sebelum didapatkan [persamaan akhir yang lebih representatif.

b. Perkalian dari hasil jumlah-hasil jumlah (Product of Sums)

Perkalian dari hasil jumlah-hasil jumlah pada dasarnya adalah merupakan metoda turunan dari metoda sebelumnya (Sum of Products). Proses penganalisaan ini dimulai dengan menjabarkan atau memodifikasi bentuk persamaan dari penjumlahan dari hasil kali-hasil kali, dengan menggunakan teorema De Morgan dan Aljabar Boolean. Adapun persamaan yang akan kita modifikasi tersebut didapatkan kebenaran juga, hanya saja yang diambil adalah komplemen keluarannya ($\overline{F}$).

Persamaan yang berasal dari komplemen keluaran tersebut, selanjutnya dijabarkan atau dikonversikan

dengan menggunakan aljabar Boolean atau teorema Morgan. Untuk lebih jelasnya dapat kita ikuti contoh berikut ini yang merupakan analisa dari tabel pada gambar 4.5.

Tabel 4.4. Product of Sums

A	B	F	$\overline{F}$
0	0	0	1
0	1	1	0
1	0	0	1
1	1	0	1

Berdasarkan tabel diatas keluaran $\overline{F}$ akan mempunyai persamaan $F = \overline{A}.\overline{B} + A.\overline{B} + A.B$ (untuk analisa menggunakan metoda Sum Of Products). Jika persamaan diatas dijabarkan maka didapat :

$$\overline{F} = \overline{A}.\overline{B} + A.\overline{B} + A.B$$

$$F = \overline{\overline{A}.\overline{B} + A.\overline{B} + A.B}$$

$$F = (\overline{\overline{A}.\overline{B}}) \; . \; (\overline{A.\overline{B}}) \; . \; (\overline{A.B})$$

$$F = \underbrace{(A + B)}_{I} \; . \; \underbrace{(\overline{A} + B)}_{II} \; . \; \underbrace{(\overline{A} + \overline{B})}_{III}$$

Penjabaran dari persamaan diatas tampak menghasilkan tiga suku (term) I, II, dan III, yang mana setiap suku tersebut dinyatakan dalam bentuk penjumlahan-penjumlahan (Sum) atau operasi gerbang OR.. Masing-masing suku yang ada tersebut selanjutnya

UPT PERPUSTAKAAN
P PADANG

digabungkan dalam bentuk perkalian (Products) atau operasi gerbang And. Sehingga bentuk akhir dari persamaan tersebut dapat disimpulkan sebagai "Perkalian dari Hasil Jumlah-Hasil Jumlah" (Product of Sums).

Contoh lain untuk tabel yang memiliki keluaran lebih dari satu kemungkinan (ada beberapa keluaran yang berlogika 0) dapat diamati tabel pada gambar 4.7 terdahulu (dalam hal ini sengaja digunakan tabel yang sama sebagai perbandingan antara proses sum of products dengan product of sums). Namun demikian bukanlah analisanya merupakan duplikasi dari pembahasan sebelumnya, karena disini titik tolak proses didasarkan dari keluaran komplemennya ($\overline{F}$).

Tabel 4.5. Keluaran dengan logika 0 Jamak

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Persamaan keluaran berdasarkan tabel diatas adalah :

$$\overline{F} = \overline{A}.\overline{B}.\overline{C} + \overline{A}.\overline{B}.C + \overline{A}.B.\overline{C} + A.\overline{B}.\overline{C}$$

maka,

$$F = \overline{\overline{A}.\overline{B}.\overline{C} + \overline{A}.\overline{B}.C + \overline{A}.B.\overline{C} + A.\overline{B}.\overline{C}}$$

$$F = (\overline{\overline{A}.\overline{B}.\overline{C}}) . (\overline{\overline{A}.\overline{B}.C}) . (\overline{\overline{A}.B.\overline{C}}) . (\overline{\overline{A}.\overline{B}.C})$$

$$F = (A+B+C) . (A+B+\overline{C}) . (A+\overline{B}+C) . (\overline{A}+B+C)$$

==========

Prosedur yang dilalui untuk mendapatkan bentuk dari Perkalian dari Hasil Jumlah-Hasil Jumlah diatas bukanlah satu-satunya cara yang dapat dilakukan.

Metoda lain yang lazim digunaakan dan sangat praktis pengalisaannya adalah dengan cara langsung mengamati tabel kebenaran, tanpa harus mengkomplemenkan keluarannya. Sebetulnya cara-cara ini hampir mirip dengan pproses penjumlahan dari hasil kali-hasil kali (SoP), hanya saja yang diperhatikan semua keluaran yang berlogika 0 (bukan yang berlogika 1, sebagaimana dalam 'Sum of Products').

Kalau pada penjumlahan dari hasil kali-hasil kali adalah merupakan gerabang OR dari beberapa gerbang AND, dalam hal ini yang terjadi adalah sebaliknya, yaitu Gerbang AND dari beberapa gerbang OR.Untuk lebih jelasnya proses tersebut dapat kita sederhanakan sebagai berikut :

- Amati semua keluaran yang berlogika 0, kemudian susun persamaan OR dari setiap baris tabel.(Karena

BAB V

RANGKAIAN ILMU HITUNG

Rangkaian ilmu hitung adalah suatu rangkaian yang digunakan untuk operasi-operasi matematis pada rangkaian logika. Rangkaian ini sangat vital sekali karena merupakan inspirasi dasar dalam penemuan kalkulator elektronik maupun komputer digital sebagaimana yang kita kenal sekarang ini. Pada umumnya rangkaian ilmu hitung tersebut diklasifikasikan atas dua bentuk, yaitu rangkaian Penjumlah (Adder), dan rangkaian Pengurang (Subtractor).

A. Rangkaian Penjumlah (Adder)

Umumnya operasi matematis dapat dikembalikan atau diselesaikan dalam bentuk penjumlahan. Misalnya saja proses perkalian, proses ini sesungguhnya hanyalah proses penjumlahan berulang (3 x 5 = 5 + 5 + 5, 2 x 4 = 4 + 4, 6 x 2 = 2 + 2 + 2 + 2 + 2 + 2,...). Berdasarkan kapasitas atau kemampuan menjumlahkan bilangan, maka rangkaian penjumlah ini juga dikelompokkan lagi atas dua bagian, yaitu Penjumlah Tidak Lengkap (Half Adder) dan Penjumlah Lengkap (Full Adder).

1. Penjumlah Tidak Lengkap (Half Adder)

Rangkaian ini disebut sebagai penjumlah tidak lengkap, karena keterbatasan kemampuan rangkaian ini dalam memproses data biner. Rangkaian ini hanya dapat

...IK UPT PERPUSTAKAAN
...IP PADANG

menjumlahkan dua bilangan biner yang masing-masing terdiri dari satu bit saja.

Disebabkan rangkaian ini hanya dapat menjumlahkan bilngan yang terdiri dari satu bit, dengan demiklian maka rangkaian ini juga hanya terdiri dari dua jalan masuk atau input (satu bit untuk data yang akan dijumlahkan dan satu lainnya untuk penjumlah). Sedangkan keluaran rangkaian ini juga terdiri atas dua keluaran, satu keluaran sebagai tampilan hasil penjumlahan (Sum) dan satu lainnya buat tampilan limpahan hasil penjumlahan (Carry Out).

Analisa terhadap proses pemjulahan data satu bit, akan menghasilkan gambaran sebagai berikut :

0 + 0 = 0 ====> limpahan (carry out) = 0

0 + 1 = 1 ====> limpahan (carry out) = 0

1 + 0 = 1 ====> limpahan (carry out) = 0

1 + 1 = 1 ====> limpahan (carry out) = 1

Pada proses analisa diatas penjumlahan 0 + 0, 0 + 1, ...tersebut merupakan proses yang terjadi pada masukan, sedangkan fungsi keluarannya adalah hasil dari penjumlahan tersebut beserta limpahan yang terjadi. Apabila data diatas ditabulasikan maka akan didapat hasil seperti tabel 5.1.

Jika berdasarkan tabel tersebut dibuatkan persamaannya, kemudian dikonversikan menjadi rangkaian dengan menggunakan metoda 'sum of products' atau 'product of sums', maka akan didapat persamaan-

persamaan keluaran sebagai berikut :

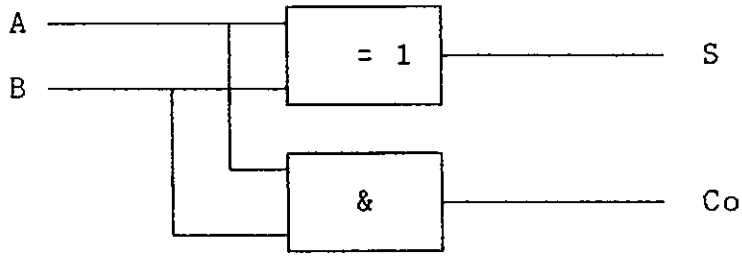
Sum (S) = $\overline{A}.B + A.\overline{B}$ ====> fungsi gerbang XOR

Carry Out (Co) = A.B =====> fungsi gerbang AND

Tabel 5.1. Penjumlahan Biner

A	B	S	Co
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Dari formulasi tabel dan fungsi-fungsi keluaran diatas maka didapat rangkaian logika sebagaimana yang dapat dilihat pada gambar 5.1.



Gambar 5.1. Penjumlah Tidak Lengkap
(M. Morris Mano;1984, h. 120)

2. Penjumlah Lengkap (Full Adder)

Penjumlah tidak lengkap hanya mmampu menjumlahkan bilangan satu bit, sedangkan pada aplikasi umumnya operasi penjumlahan kondisi ini sangat tidak

...LIK UPT PERPUSTAKAAN
IKIP PADANG

mencukupi, karena kebanyakan operasi logika akan melibatkan lebih dari satu bit data. Dengan demikian maka dirancanglah suatu rangkaian yang mampu menjumlahkan multi bit data biner, yang biasa disebut sebagai Rangkaian Penjumlah Lengkap (Full Adder).

Dalam penjumlahan yang melibatkan lebih dari satu bit data, biasanya akan terjadi beberapa tahapan kerja yang berurutan atau beraturan sebagai berikut :

- Proses penjumlahan dimulai dari bit yang terendah bobotnya (Least Significant Bit / LSB).
- Hasil penjumlahan (Sum) pada kolom LSB tersebut jika menghasilkan lebih dari satu bit, maka yang ditulis hanyalah bit yang terakhir saja (LSB), sedangkan bit tinggi (MSB) hasil penjumlahan tersebut akan menjadi limpahan keluaran (Carry Out).
- "Carry Out" yang didapat sewaktu menjumlahkan LSB, merupakan masukan bagi kolom selanjutnya, yang biasa disebut dengan limpahan masukan (Carry In).

Untuk lebih jelasnya dapat diikuti proses penjumlahan bilangan berikut ini yang terdiri dari empat bit :

$A_4\ A_3\ A_2\ A_1$:	1 0 0 1	==>	Bilangan yang ditambah.
$B_4\ B_3\ B_2\ B_1$:	0 1 0 1	==>	Bilangan yang menambah.
$S_4\ S_3\ S_2\ S_1$:	1 1 1 0	==>	Hasil Penjumlahan.
	0 0 0 1	==>	Pindahan Keluaran.
	0 0 1 0	==>	Pindahan Masukan.

Tata urut penjumlahan bilangan diatas, dimulai dari bit terendah yaitu A_1 dengan B_1. Penjumlahan kolom ini akan menghasilkan pindahan keluran (Co) 1, yangmana pindahan keluaran ini selanjutnya akan dijumlahkan pada kolom berikutnya (A_2dan B_2) sebagai pindahan masukan. Dengan demikian maka pada bit ini akan terjadi penjumlahan A_2 + B_2 + Cin. Dengan proses yang sama maka pada penjumlan ini juga akan menghasilkan pindahan bagi bit berikutnya. Dengan demikian maka pada bit-bit berikutnya tersebut akan dilakukan tiga bit penjumlahan, yaitu antara bilangan yang ditambah dengan bilangan penambah serta pindahan dari bit sebelumnya.

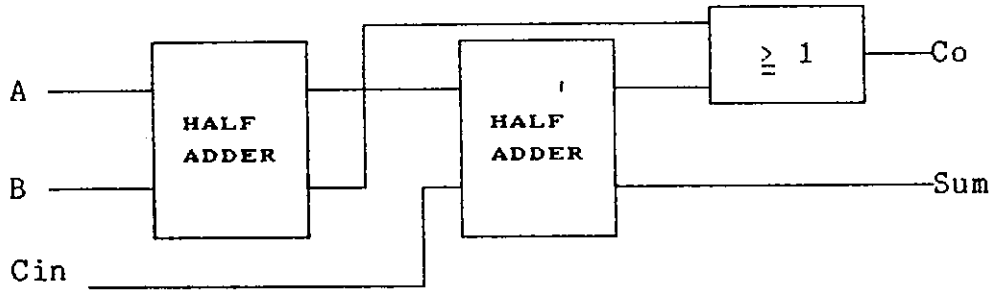
Berdasarkan penganalisaan pada prosees penjumlahan diatas, maka jelaslah bahwa pada rangkaian Penjumlah Lengkap ini akan membutuhkan tiga masukan (A, B, dan Cin), dan dua keluaran yaitu Sum dan Co. Dengan adanya tiga masukan maka akan diperoleh 2^3 (8) fariasi masukan, atau secara tabulasinya dapat dilihat sebagaimana tabel 5.2.

Blok diagram dari rangkaian penjumlah ini merupakan gabungan dari dua penjumlah tidak lengkap, yang dilengkapi dengan gerbang OR untuk limpahan keluaran (Co). Lebih jelasnya tabel serta diagram

lengkap rangkaian ini dapat dilihat pada gambar 5.2 berikut ini :

A	B	Cin	S	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

a. Tabel



b. Rangkaian

Gambar 5.2. Blok diagram Penjumlah Lengkap
(G. Sudarto;1980, h. 162)

Jika gambar 5.2 diatas diuraikan dengan mentransformasikan data pada tabel serta menggunakan

penyelesauian 'jumlah dari hasil kali-hasil kali' (sum of products), maka akan didapat persamaan sebagai berikut :

$$* \text{ Sum } = \bar{A}.\bar{B}.C_{in} + \bar{A}.B.\overline{C_{in}} + A.\bar{B}.\bar{C}_{in} + A.B.C_{in}$$

$$= (\bar{A}.B + A.\bar{B}).\bar{C}_{in} + (A.B + \bar{A}.\bar{B}).C_{in}$$

$$= X . \overline{C_{in}} + Y . C_{in}$$

$$* \; C_{out} = \bar{A}.B.C_{in} + A.\bar{B}.C_{in} + A.B.\overline{C_{in}} + A.B.C_{in}$$

$$= (\bar{A}.B + A.\bar{B}).\bar{C}_{in} + A.B .(\bar{C}_{in} + C_{in})$$

$$= X . C_{in} + A.B$$

X = Fungsi Keluaran gerbang EXOR

Y = Fungsi keluaran Komparator/EXNOR

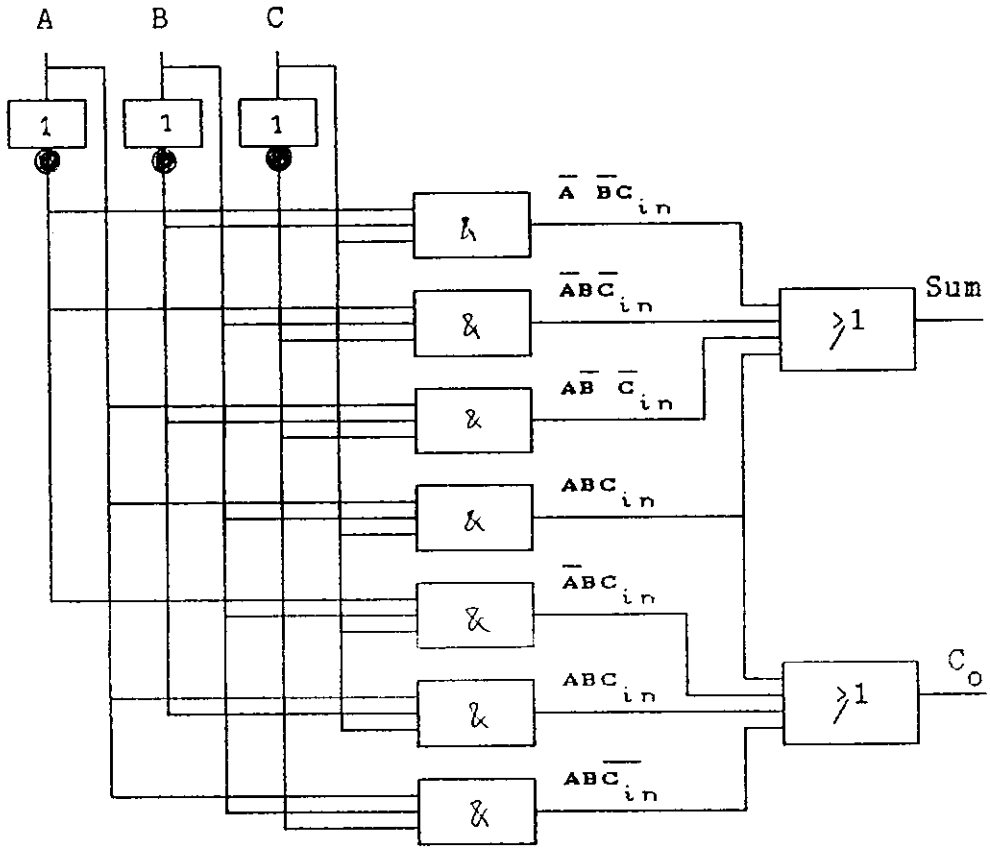
(Sudarto;1980, h.99)

Ekspresi rangkaian logika dari persamaan diatas (Persamaan pada baris pertama) adalah sebagaimana yang terdapat pada gambar 5.3.

Secara logika rangkaian diatas hanya akan menjumlahkan bilangan biner secara berurutan, yaitu A + B + Cin. Tentu saja penjumlahan tersebut haruslah dilakukan mulai dari bit yang mempunyai bobot paling rendah (LSB), setelah itu baru dilanjutkan untuk bit-bit berikutnya.

Sebagaiman ulasan pada awal bab ini, dinyatakan bahwa "Full Adder" tersebut mampu menjumlahkan

bilangan biner dengan bit tidak terbatas, namun perlu digaris bawahi bahwa untuk rangkaian sebagaimana pada gambar 5.2 tersebut hanya dapat dilakukan secara deret atau serial.



Gambar 5.3. Rangkaian Gerbang Penjumlah Lengkap (Floyd;1986, h.203)

a. Penjumlahan Secara Paralel

Penjumlahan secara paralel sangat banyak diterapkan, karena pada proses ini semua bit data tersebut diolah secara bersamaan atau serentak. Dengan

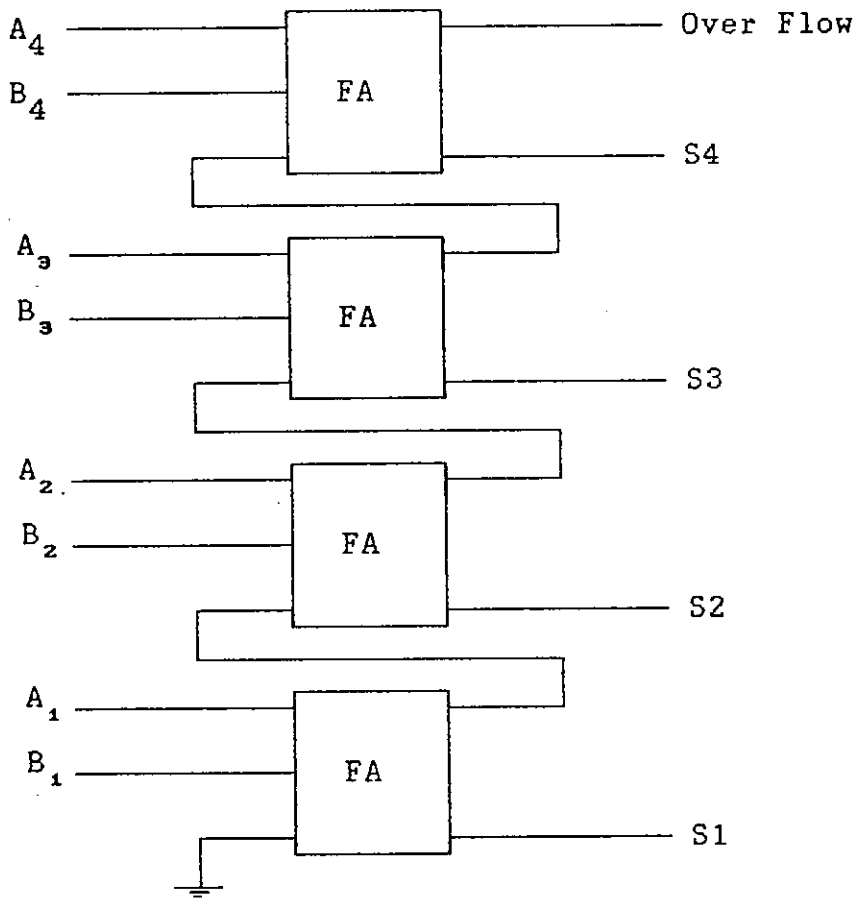
demikian maka proses ini dapat berlangsung dalam tempo yang lebih cepat . Sedangkan pada gambar 5.2 terdahulu proses tersebut dilakukan atau berlangsung bit demi bit sehingga membutuhkan waktu yang relatif lebih lama.

Dalam penjumlahan secara paralel tersebut, jumlah bit bilangan yang akan dijumlahkan sangat menentukan akan jumlah 'Full Adder' yang diperlukan pada rangkaian. Jika ingin menjumlahkan empat bit bilangan biner, maka untuk ini harus disediakan empat buah penjumlah lengkap yang dirangkaikan sedemikian rupa, sehingga penjumlahan akan berlangsung pada saat yang bersamaan untuk ke-empat bit tersebut.

Lebih jelasnya konstruksi lengkap dari penjumlah paralel 4-bit (Hanya sebagai contoh digunakan penjumlah 4-bit) dapat dilihat pada gambar 5.3.

Pada gambar 5.3 terdapat empat buah penjumlah lengkap (FA), yang mana masing-masing penjumlah tersebut memiliki tiga jalan masukan dan keluaran. Sebagomana kita ketahui bahwa pada setiap awal atau pertama kali proses penjumlahan dilakukan (pada bit terendah atau LSB), maka dalam hal ini proses tersebut hanya berlangsung antara bilangan yang akan dijumlahkan dengan bilangan yang penambah (A : Addend = Bilangan yang ditambah dengan B : Augend = bilangan penambah), tanpa adanya penjumlahan limpahan. Dengan demikian penjumlah lengkap pada orde bit terendah

(LSB) tersebut sesungguhnya dapat saja digantikan dengan penjumlahan tidak lengkap, karena dalam hal ini hanya melibatkan dua proses penjumlahan masukan. Akan tetapi pada gambar 5.3 sengaja hal ini tidak dilakukan, karena untuk penggandengan beberapa modul penjumlah lengkap 4-bit menjadi penjumlah 8-bit, 12-bit dan seterusnya hal ini dirasakan kurang praktis.



Gambar 5.4, Penjumlah Paralel 4 Bit

(Millman Halkias;1985, h.598)

Selanjutnya pada gambar 5.3 juga dapat kita lihat bahwa pengawatan antara limpahan keluaran bit terendah selanjutnya akan dikopelkan ke bit berikutnya. Model ini seterusnya akan berlanjut hingga ke bit tertinggi rangkaian (MSB). Perbedaan yang ada disini adalah antara limpahan keluaran (Co) pada bit tertinggi tersebut (C3), disini C 3 tetap dibiarkan tidak terhubung sedemikian rupa, dan biasanya dikenal dengan istilah 'Over Flow', atau *limpahan berlebih*, yang mana hal ini nantinya diperlukan untuk mengamati hasil penjumlahan sesungguhnya, untuk proses penjumlahan yang tidak tertampung oleh keluaran 'sum'. misalnya pada penjumlahan berikut ini :

```
      1   1   0   1
      1   0   1   1
  ------------------  +
  1   1   0   0   0
  ---  -------------
  :        :
  :        :.........> Sum
  :
  :.......................> Over Flow/limpahan
```

Pada proses penjumlahan diatas kalau hanya diamati keluaran sum saja, maka hasilnya adalah 1 0 0 0. Hasil ini tentu saja keliru, untuk itu maka hasil akhir setiap proses selalu harus diamati limpahannya sehingga hasil sesungguhnya adalah dengan menempatkan hasil limpahan tersebut menjadi MSB. Berdasarkan data

diatas sum adalah 1000 dan limpahan 1, maka hasil akhir menjadi 11000.

B. Rangkaian Pengurang (subtraktor)

Rangkaian pengurang atau populernya dikenal dengan nama rangkaian subtraktor, adalah rangkaian logika yang berfungsi untuk melakukan proses pengurangan bilangan biner.

Proses pengurangan bilangan biner tersebut dapat diklasifikasikan berdasarkan jenis bilangan biner yang digunakan. Adakalanya dalam satu proses menggunakan bilangan *biner murni* yaitu bilangan biner sebagaimana yang kita kenal sejauh ini (yang digunakan dalam pembahasan pada bab-bab terdahulu). Proses lainnya yaitu menggunakan *bilangan biner bertanda* (Sign binary digit), sebagaimana yang akan kita bahas pada bagaian mendatang.

1. Pengurangan dengan Bilangan Biner Murni

Pola yang digunakan pada proses pengurangan bilangan biner, sesungguhnya sama saja dengan pola yang digunakan pada bilangan desimal. Dalam hal ini akan didapatkan dua bentuk atau dua kemungkinan hasil proses yaitu : proses akan menghasilkan bilangan positif atau nol, jika bilangan yang dikurangi (minuend) lebih besar atau sama dengan bilangan pengurang (subtrahend). Kemungkinan kedua yaitu akan didapat hasil negatif apabila bilangan yang dikurangi tersebut lebih kecil dari bilangan pengurangnya.

Sebagai kesimpulan dasar pada prosesa pengurangan bialangan biner akan didapat empat kemungkianan sebagai berikut :

* 0 - 0 = 0 pinjaman masukan 0
* 0 - 1 = 1 pinjaman masukan 1
* 1 - 0 = 1 pinjaman masukan 0
* 1 - 1 = 0 pinjaman masukan 0

Berdasarkan empat kemungkinan di atas kelihatan bahwa disini akan terjadi suatu proses yang dikenal dengan istilah pinjaman (borrow). Pinjaman ini juga dikelompokan jadi dua, yaitu pinjaman masukan (Borrow input) dan pinjaman keluaran (Borrow Output).

Pinjaman masukan terjadi apabila pada proses pengurangan, bilangan yang dikurangi (misalkan X) bernilai lebih kecil dari bilangan pengurangnya (misalkan Y). Dari empat kemungkinan di atas kondisi ini terjadi pada saat 0 - 1, disisni terjadi proses peminjaman ke-bit yang lebih tinggi 1 bit. nilai satu bit ini selanjutnya dimasukan ke bit dibawahnya yang mana nilainya akan berharga 2 (untuk bilangan desimal proses peminjaman ini akan bernilai 10 untuk bit dibawahnya).

Hasil akhir dari proses pengurangan tersebut biasa disebut dengan Selisih (Difference/D). Secara

MILIK UPT PERPUSTAKAAN
IKIP PADANG

lengkap porses subtraksi tersebut dapat ditabulasikan seperti terlihat pada tabel 5.3.

Tabel 5.3. Subtraksi

Yang diku-rangi (X)	Pengurang (Y)	Selisih (D)	Pinjaman (B)
0	0	0	0
0	1	1	1
1	0	1	0
1	1	1	0

a. Rangkaian Pengurang Tidak Lengkap

Sama halnya dengan rangkaian penjumlah, pada rangkaian pengurang juga ada rangkaian yang disebut dengan rangkaian tidak lengkap. Tidak lengkapnya rangkaian ini juga ditandai dari kemampuan rangkaian yang hanya dapat dioperasikan untuk mengurangi satu bit bilangan biner saja.

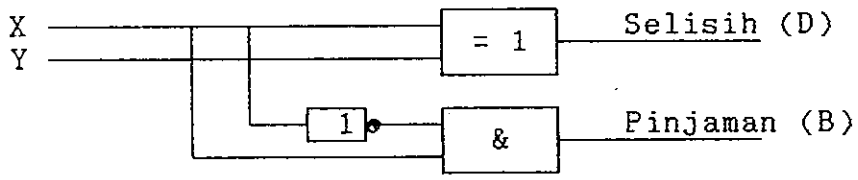
Jika tabel subtraksi di atas diselesaikan dengan menggunakan metoda 'sum of products', maka akan kita dapat persamaan sebagai berikut :

@ Selisih (D) : $\overline{X}.Y + X.\overline{Y}$

@ Pinjaman (B) : $\overline{X}.Y$

Persamaan-persamaan di atas menunjukan bahwa untuk keluaran selisih (D) tersebut adalah keluaran gerbang EXOR (persis sama dengan keluaran 'Sum'pada rangkaian penjumlah). Sedangkan keluaran

(B) merupakan keluaran gerbang AND yang mana salah satu masukannya yaitu bilangan yang dikurangi dikomplemenkan (diberi Inverter). Dengan demikian maka konstruksi dasar dari pengurang tidak lengkap ini dapat di amati pada gambar 5.5.



Gambar 5.5. Pengurang Tidak Lengkap
(Wasito;1983, h.261)

b. Pengurang Lengkap

Pengurang lengkap merupakan penyempurnaan dari pengurang tidak lengkap. Kesempurnaan ini antara lain ditandai dari kemampuan rangkaian, yang mana pada pengurang lengkap ini dapat dioperasikan untuk lebih dari 1 bit data.

Kalau kita perhatikan proses maupun tabel kebenaran pengurangan, jelas terlihat adanya kesamaan yang erat antara proses atau data pengurangan dengan penjumlahan. Perbedaan yang nyata dalam hal ini hanya terjadi antara proses peminjaman (borrow) pada rangkaian pengurang dengan pindahan (carry) pada rangkaian penjumlah.

Oleh karena itu model tabulasi data pada pengurang lengkap juga tidak jauh berbeda dengan

penjumlah lengkap, hanya saja disini tentu saja dalam bentuk pengurangan. Atau lebih sederhananya proses pada tabel tersebut dinotasikan sebagai berikut :

$$X - Y - B_{in} = D \text{ dengan pinjaman } B_o$$

Bentuk lengkap dari tabulasi pengurangan tersebut dapat di lihat tabel berikut ini :

Tabel 5.4. Pengurang Lengkap

X	Y	Bin	D	Bo
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Dengan cara yang sama dengan rangkaian penjumlah lengkap (menggunakan metoda penjumlahan dari hasilkali-hasil kali) maka tabel diatas dapat dinotasikan menjadi :

$$* \quad D = \overline{X}.\overline{Y}.B_{in} + \overline{X}.Y.\overline{B_{in}} + X.\overline{Y}.\overline{B}_{in} + X.Y.B_{in}$$

$$= (\overline{X}.Y + X.\overline{Y}).\overline{B}_{in} + (X.Y + \overline{X}.\overline{Y}).B_{in}$$

$$= P \ . \ \overline{B_{in}} + Q \ . \ B_{in}$$

$$* \ B_{out} = \overline{X}.\overline{Y}.B_{in} \quad \overline{X}.Y.\overline{B}_{in} + \overline{X}.Y.B_{in} + A.B.C_{in}$$

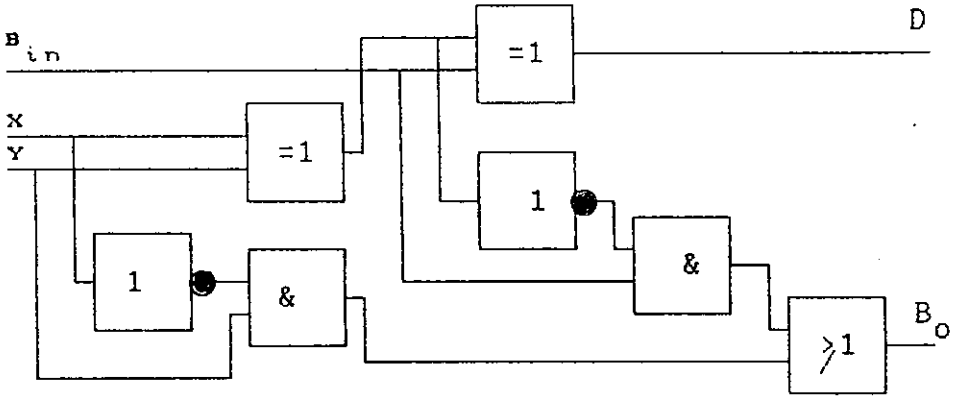
$$= (\overline{X}.\overline{Y} + X.Y).B_{in} + \overline{X}.Y$$

$$= Q \ . \ B_{in} + X.Y$$

P = Fungsi Keluaran gerbang EXOR

Q = Fungsi keluaran Komparator/EXNOR

Namun karena keterbatasan (jarang digunakan) gerbang Exor yang terdiri dari tiga masukan, maka biasanya rangkaian pengurang lengkap ini direalisir dengan menggunakan gerbang-gerbang logika yang terdiri dari dua masukan, sebagaiman yang terlihat pada gambar 5.6.



Gambar 5.6. Pengurang Lengkap
(Wasito ;1983, h. 262)

2. Pengurangan dengan Menggunakan Biner Bertanda

Sebagaimana yang telah diulas pada awal bab ini, bahwa pada dasarnya semua proses rangkaian aritmatika dapat diselesaikan atau dimanipulasi dengan menggunakan rangkaian penjumlah. Dengan demikian maka proses pengurangan otomatis juga dapat dilakukan dengan menggunakan rangkaian penjumlah.

Berdasarkan uraian pada sub bab rangkaian pengurang, jelas terlihat adanya banyak kesamaan antara rangkaian pengurang dengan rangkaian penjumlah. Similaritas ini tampak pada keluaran serta penggunaan gerbangnya. Secara matematispun pengurangan tersebut ssesungguhnya sama saja dengan menambahkan bilangan yang dikurangi dengan bilangan negatif dari bilangan pengurang.

misalnya :

$$\begin{array}{r} 14 \\ 3 \\ \hline 11 \end{array} - \quad \Longrightarrow \quad \begin{array}{r} 14 \\ (-3) \\ \hline 11 \end{array} +$$

Untuk menuliskan bilangan negatif pada sistem bilangan biner dilakukan dengan memberikan tanda. Tanda tersebut dinyatakan dalam bentuk angka 1 dan 0, yang ditempatkan pada bit terdepan (MSB). Sehingga pada sistem bilangan biner bertanda (Binary Sign bit) bit terdepan tersebut yang biasanya mempunyai bobot paling tinggi (pada sistem bilangan biner murni), maka

disini tidak mempunyai nilai (biasanya di tulis dalam tanda kurung). Lebih jelasnya dapat dilihat contoh berikut ini :

- *Angka 0 pada bit terdepan* menunjukan bahwa bilangan tersebut *positif*.

  misal : + 1 1 0 ditulis (0) 1 1 0

  + 1 0 1 ditulis (0) 1 0 1

- *Angka 1 pada bit terdepan* menunjukan bahwa bilangan tersebut *negatif*.

  misal : - 1 1 0 ditulis (1) 1 1 0

  - 1 0 1 ditulis (1) 1 0 1

a. Komplemen

Untuk mengekspresikan bilangan biner negatif secara penulisan tidak menjadi masalah. Permasalahan baru muncul saat kita akan mengekspresikannya dalam bentuk level tegangan.

Ditinjau dari sisi level tegangan, bilangan biner tersebut hanya diklasifikasikan berdasarkan ada tidaknya tegangan, yang mana logika 1 berarti ada tegangan dan logika 0 untuk tidak ada tegangan. Sedangkan bilangan negatif adalah bilangan yang mempunyai nilai dibawah atau lebih kecil dari 0.

Bentuk komplemen merupakan salah satu cara yang dilakukan dalam mengekpresikan bilangan negatif tersebut. Ada beberapa metoda penulisan komplemen, namun yang menyangkut bilangan biner

cuma dikenal komplemen-1 (1's complement), dan komplemen-2 (2's complement).

1) Komplemen-1

Pada awal-awal buku ini sudah disinggung-singgung istilah komplemen, yaitu sesuatu yang menyangkut bentuk inverse atau lawan dari suatu logika. Pada sistem komplemen-1 juga terjadi hal yang sedemikian yang mana pada sistem ini semua bilangan yang ada dikonversikan ke bentuk lawannya.

Sebagaimana yang kita ketahui bahwa pada bilangan biner tersebut hanya ada dua bilangan, yaitu 1 dan 0. Dengan kata lain dapat ditarik suatu kesimpulan bahwa, penulisan bentuk komplemen-1 tersebut dilakukan dengan cara merubah semua angka 0 menjadi 1 sebaliknya semua angka 1 dirubah menjadi 0.

Perubahan logika ini tidak hanya dilakukan untuk bilangan biner murni, namun juga berlaku pada bilangan biner bertanda. Khusus untuk bilangan biner bertanda ini, tanda bilanganpun ikut juga dikonversikan.

misalnya ;

-	1 0 1 0	komplemen-1nya	0 1 0 1
-	(0) 1 0 0 1	komplemen-1nya	(1) 0 1 1 0
-	(1) 1 1 0 1	komplemen-1nya	(0) 0 0 1 0

Berdasarkan bilangan komplemen-1 ini maka selanjutnya proses subtraksi dapat dilakukan dengan cara menjumlahkan.

Pengurangan yanhg menggunakan azas penjumlahan dengan komplemen-1 ini, pada dasarnya berlangsung atas dua tahap. Tahap pertama yaitu menjumlahkan bilangan yang dikurangi dengan bilangan yang mengurangi atau bilangan komplemen-1. Tahap kedua yaitu menambahkan hasil penjumlahan pada tahap pertama tadi dengan pindahan (Carry) yang terjadi. Penambahan ini dilakukan pada bagian LSB hasil penjumlahan.

Langkah pada tahap kedua ini saangat penting sekali, khususnya jika proses penjumlahan tersebut menghasilkan bilangan positif (tanda bilangan 0). Pindahan yang dijumlahkan ini populer disebut dengan 'End Around Carry' (EAC), yaitu pindahan dari hasil penjumlahan digit terakhir yang harus ditambahkan pada hasil penjumlahan tersebut

Hal lain yang perlu diperhatikan ialah cara pembacaan hasil akhir proses. Apabila proses meenghasilkan bilangan dengan tanda 0 (positif), maka hasil akhir dapat langsung dibaca setelah dilakukan penjumlahan dengan pindahannya (proses tahap dua di atas). Sebaliknya jika proses

menghasilkan bilangan dengan tanda 1 (bilangan negatif), maka hasil tersebut masih saja dalam bentuk komplemen-1, sedangkan hasil akhirnya harus dikonversikan terlebih dahulu kebentuk bilangan biner murni atau bertanda.

Untuk lebih jelasnya dapat diamati contoh-contoh berikut ini :

1. Misalnya bilangan desimal 13 - 8
   - Bilangan pengurang = $(8)_{10}$ = (0) 1 0 0 0
   - Bentuk komplemen-1 bilangan pengurang adalah (1) 0 1 1 1

   maka :

```
      ( 0 ) 1 1 0 1
      ( 1 ) 0 1 1 1
      ---------------  +
1     ( 0 ) 0 1 0 0
 |
 |..................> 1  ......> E A C
      ---------------  +
      ( 0 ) 0 1 0 1
       |        |.....> Hasil akhir
       |                ( 5 )10
       |.................> Tanda Bilangan
                          (Bil.Positif)
```

2. Bilangan desimal 5 - 9
   - Bilangan yang dikurang ; (0) 0 1 0 1
   - Bilangan pengurang ; (0) 1 0 0 1
   - Komplemen-1 bilangan pengurang ;
     (1) 0 1 1 0

maka :

```
        ( 0 ) 0 1 0 1
        ( 1 ) 0 1 1 0
        ---------------  +
      0 ( 1 ) 1 0 1 1
```

→ Tidak ada E A C (Hasil akhir dalam bentuk dalam komplemen-1)

atau :

1 0 1 1 (1's comp.) = $(0\ 1\ 0\ 0)_2 = (4)_{10}$

Aplikasi pengubahan bilangan biner kebentuk komplemen ini dilakukann dengan suatu rangkaian khusus yang berfungsi sebagai inverter sekaligus sebagai kontrol.

Dengan adanya masukan kontrol ini maka efisisensi rangkaian dapat ditingkatkan. Efekktifitas rangkaian tersebut disebabkan karena melalui masukan kontrol ini rangkaian dapat dikendalikan apakah akan difungsikan sebagai rangkaian pengurang (subtraktor) ataukah akan berfungsi sebagai rangkaian penjumlah (adder). Rangkaian kontrol ini lazim dikenal dengan nama rangkaian 'True/Complement'.

Pada dasarnya rangkaian kontrol ini dibentuk dari kombinasi gerbang XOR, And dan Inverter. Secara keseluruhannya rangkaian ini terdiri dari tiga masukan (dua untuk kontrol dan

MILIK UPT PERPUSTAKAAN
IKIP PADANG

satu untuk data) dan satu keluaran.

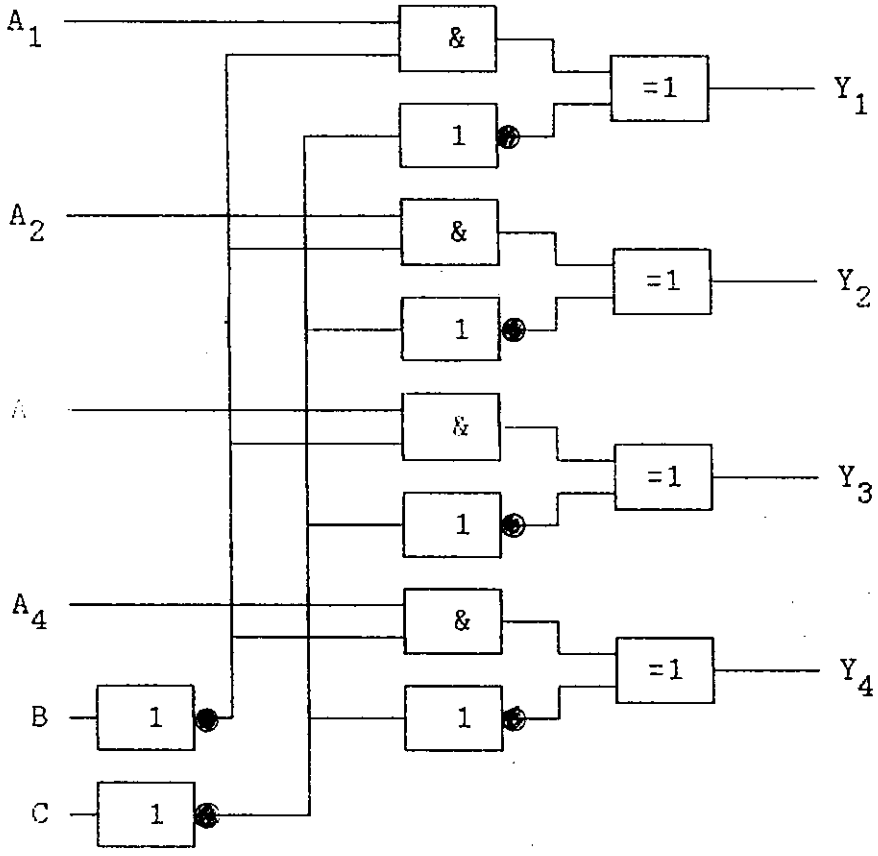
Akan tetapi pada umumnya rangkaian pengontrol tersebut sudah dikemas dalam bentuk rangkaian terpadu (IC). Untuk lebih memahami prinsip kerja rangkaiannya dapat diamati konstruksi dasar dari rangkaian 'True Complement' yang terdapat pada IC SN74H87, yaitu rangkaian kontrol/true complement yang terdiri dari 4 bit.

Analisa data pada gambar 5.7 khususnya tabel kebenarannya menunjukan bahwa :

- Baris pertama, rangkaian akan menghasilkan komplemen masukan.
- Baris kedua, keluaran rangkaian akan sama dengan masukan atau rangkaian ahanya akan meneruskan masukan.
- Baris ketiga, keluaran rangkaian akan selalu logika 1 (Kondisi set).
- Baris keempat, keluaran rangkaian akan selalu logika 0 (kondisi reset).

Berdasarkan kemungkinan-kemungkinan di atas, maka dapat ditarik kesimpulan bahwa rangkaian akan menjadi rangkaian pengurang (subtraktor), apabila terjadi penjumlahan dengan bilangan komplemen-1. Yang mana dalam hal ini terjadi pada saat kedua masukan kontrol (B & C) berlogika 0. Sebaliknya rangkaian akan menjadi

rangkaian penjumlah (adder) apabila masukan kontrol berada pada kondisi B = 0 (Low/L), dan C = 1 (High/H).

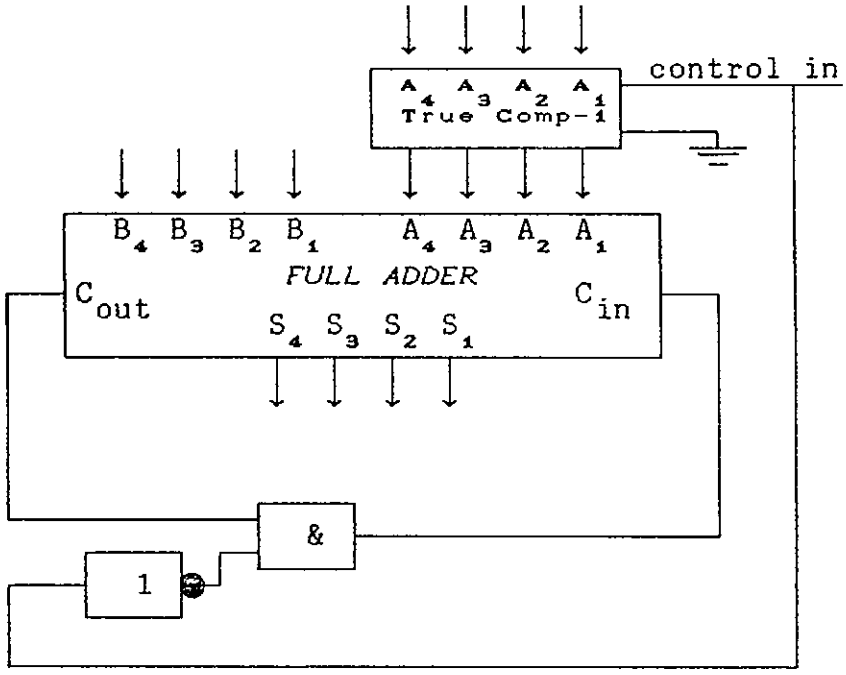


Control Input		OUTPUT			
B	C	Y_1	Y_2	Y_3	Y_4
L	L	$\overline{A_1}$	$\overline{A_2}$	$\overline{A_3}$	$\overline{A_4}$
L	H	A_1	A_2	A_3	A_4
H	L	H	H	H	H
H	H	L	L	L	L

Gambar 5.7. 4 Bit True/Complement-1

(Carlson ; 1981, h. 191)

Proses penjumlahan dan pengurangan tersebut dapat baru direalisasikan rangkaiannya dengan cara menggabungkan antara rangkaian penjumlah dan rangkaian 'true/complement'. Lengkapnya blok diagram rangkaian pengurang yang sekaligus juga dapat dijadikan rangkaian penjumlah tersebut dapat di lihat pada gambar 5.6.



Gambar 5.8. Pengurang/Penjumlah Komplemen-1
(Sudarto;1980, h.185)

2) **Komplemen-2**

Komplemen-2 adalah pengembangan lebih lanjut dari komplemen-1. Lebih tepatnya dapat kita simpulkan bahwa komplemen-2 tersebut adalah komplemen-1 yang ditambahkan dengan 1 pada

bagian LSBnya.

Dengan demikian proses pengurangan dengan cara penjumlahan juga dapat dilakukan dengan menggunakan komplemen-2 ini. Bahkan cara pengurangan yang lebih umum/banyak dilakukan adalah dengan menggunakan komplemen-2. Hal ini disebabkan karena dengan menggunakan komplenmen-2 tersebut, maka rangkaian yang diperlukan dapat lebih disederhanakan lagi.

Berikut ini dapat dilihat cara-cara pengurangan yang dilakukan dengan cara menjumlahkan komplemen-2,

Misalnya :

- Desimal : 13 - 9
- Biner : (0)1 1 0 1 - (0)1 0 0 1
- Bilangan Pengurang : (0)1 0 0 1 ==> (positif)
- Bentuk Komp-1 : (1)0 1 1 0
- Bentuk Komp-2 : 1 +
  (1)0 1 1 1 ==> (negatif)

Penjumlahan :

```
    ( 0 ) 1 1 0 1
    ( 1 ) 0 1 1 1
  ---------------- +
1   ( 0 ) 0 1 0 0
```

→ Hasil akhir

→ Tanda bilangan positif, pindahan diabaikan.

Perbedaan mendasar antara komplemen-1 dengan komplemen-2 antara lain pada saat pembacaan hasil akhir proses.

Jika proses menghasilkan bilangan positif (bertanda 0), pada komplemen-2 pindahan yang didapat diabaikan saja, sehingga tidak terjadi 'end around carry' sebagaimana yang terjadi pada komplemen-1. Dengan demikianmaka hasil akhir proses tersebut dapat langsung ditentukan (pada contoh diatas $(0)0100_2 = 4_{10}$).

Apabila proses menghasilkan bilangan dengan tanda negatif (1), maka hasil akhir proses terlebih dahulu harus dikonversikan kedalam bentuk komplemen-1. Selanjutnya komplemen-1 inipun juga ditransfer menjadi bentuk komplemen-2. Bentuk komplemen-2 inilah yang menjadi hasil akhir proses yang dapat diindikasi.

Misalnya :

- Desimal : 6 - 10
- Biner : (0)0 1 1 0 - (0)1 0 1 0
- Bilangan Pengurang : (0)1 0 1 0 ==> (positif)
- Bentuk Komp-1 : (1)0 1 0 1
- Bentuk Komp-2 1 +

(1)0 1 1 0 ==> (negatif)

Penjumlahan :

```
( 0 ) 0 1 1 0
( 1 ) 0 1 1 0
-------------- +
( 1 ) 1 1 0 0
```

→ Tanda bilangan negatif, hasil harus di proses terlebih dahulu.

Hasil penjumlahan = (1) 1 1 0 0

Bentuk komp-1 : (1) 1 1 0 0

Bentuk komp-2 1

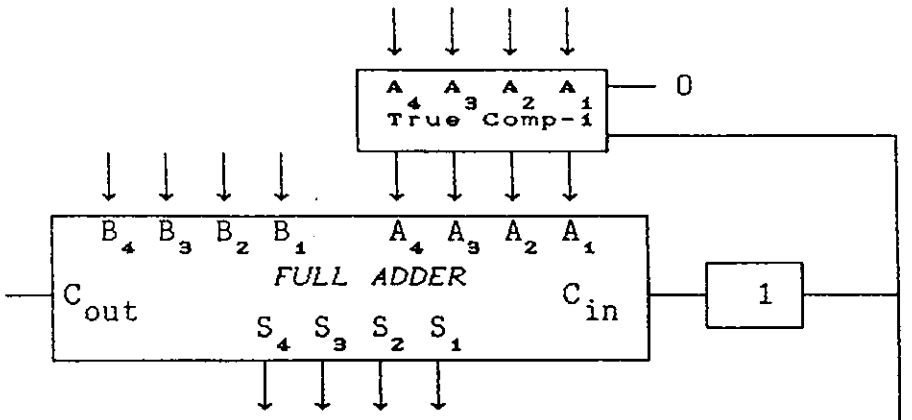
---------- +

(0) 0 1 0 0

↓

Hasil akhir

Diagram aplikasi dari rangkaian pengurang/ penjumlah komplemen-2 ini dapat dilihat pada gambar 5.7 berikut :



1 = Penjumlah

0 = Pengurang

Gambar 5.9. Penjumlah/Pengurang Komplemen-2 (Carlson ; 1981, h. 191)

DAFTAR BACAAN

Carlson, Per & Staffan Johanson, Digital Technology, third edition, Liber Tryck, Stockholm, 1981.

Floyd, Thomas L., Digital Fundamentals, third edition, A Bell & Howell Co., Columbus, Ohio 43216, 1986.

Lindell, Börje, Digital and Pulse Circuits, Kristianstads Boktrykeri A B, Stockholm, 1981.

Malvino, Albert Paul, Elektronika Komputer Digital, edisi kedua, Terjemahan oleh : Tjia May On, Erlangga, Jakarta, 1989.

Mano, Morris M., Digital Design, Prentice-Hall Inc., New Jersey, 1984.

Millman, Jacob & Christos C. Haalkias, Integrated Electronic : Analog and Digital Circuits and Systems, 27th Printing, MacGraw Hill Book Co., Singapore, 1985.

Sudarto, Gatot, Dasar-dasar Sistem Digital, Usaha Nasional, Surabaya, 1980.

Suryatmo, F., Teknik Digital, cetakan pertama, Bina Aksara, Jakarta, 1986.

Wasito, S. & B. Hernawan, Teknik Digit, cetakan keempat, Karya Utama, Jakarta, 1986.

Williams, Arthur B., Designer's Handbook of Integrated Circuits, McGraw-Hill Book Co., New York, 1984.

Tocci, Ronald J., Digital System "Principles and Applications", revised and enlarge, Prentice-Hall Inc., New Jersey, 1980.