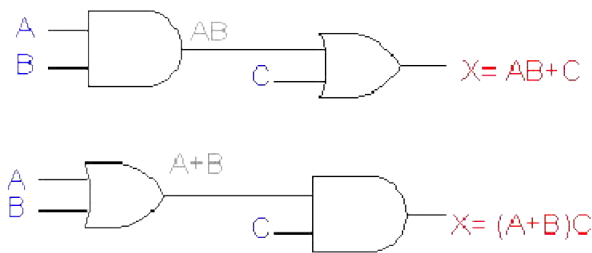


BAB IV ALJABAR BOOLEAN

4.1 Menguraikan Rangkaian-Rangkaian Logika Secara Aljabar

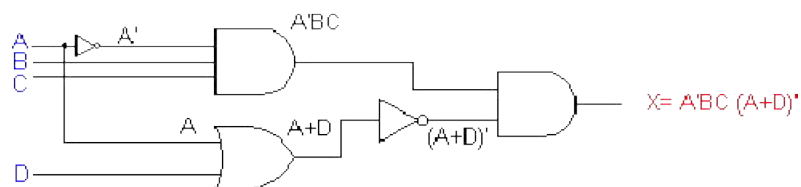
Setiap rangkaian logika, bagaimanapun kompleksnya, dapat diuraikan secara lengkap dengan menggunakan operasi-operasi Boolean yang telah didefinisikan sebelumnya, karena rangkaian OR gate, AND gate, dan NOT gate merupakan blok-blok bangun dasar dari system-sistem digital. Perhatikan rangkaian pada gambar 4.1. Rangkaian ini tiga input, A, B, dan C dan suatu output X. Dengan menggunakan ekspresi Boolean untuk tiap gate dengan mudah dapat ditentukan ekspresi outputnya.



Gambar 4.1 Rangkaian logika dengan ekspresi Booleannya

Rangkaian-rangkaian yang menggunakan Inverter

Jika suatu Inverter ada disuatu rangkaian logika, ekspresi outputnya secara sederhana sama dengan ekspresi input dengan tandan string (') di atasnya.



Gambar 4.2 Rangkaian yang menggunakan Inverter

4.2 Mengevaluasi Output Rangkaian Logika

Sekali ekspresi Boolean untuk suatu output rangkaian diperoleh, level logika dari output dapat ditentukan untuk setiap harga –harga dari input-input rangkaian.

Berikut ini dua buah contoh untuk mengevaluasi output rangkaian logika :

Misal : A=0, B=1, C=1, D=1

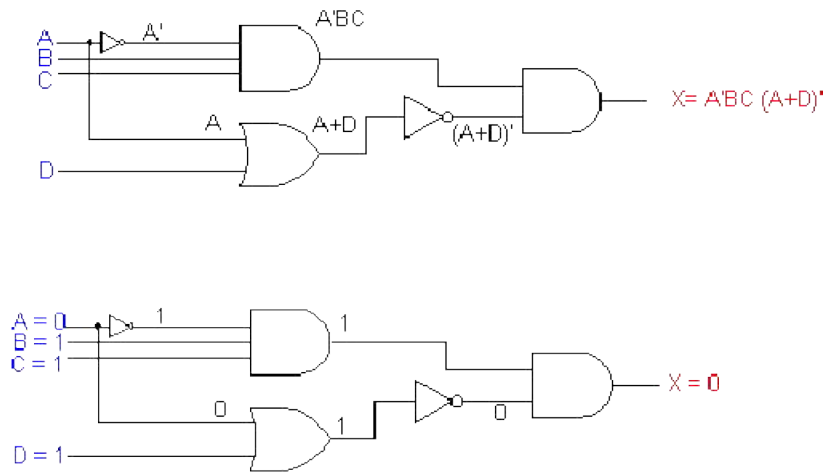
$$\begin{aligned} X &= A'BC (A+D)' \\ &= 0' * 1 * 1 * (0+1)' \\ &= 1 * 1 * 1 * (1)' \\ &= 1 * 1 * 1 * 0 \\ &= 0 \end{aligned}$$

Misal : A=0, B=0, C=1, D=1, E=1

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

Penentuan Level Output Dari Suatu Diagram

Level logika output untuk level-level input tertentu dapat juga ditentukan secara langsung dari diagram rangkaian tanpa menggunakan operasi Boolean. Teknik ini sering digunakan oleh para teknisi selama mencari kerusakan atau mengetest suatu system logika. Perhatikan gambar 4.3 berikut :



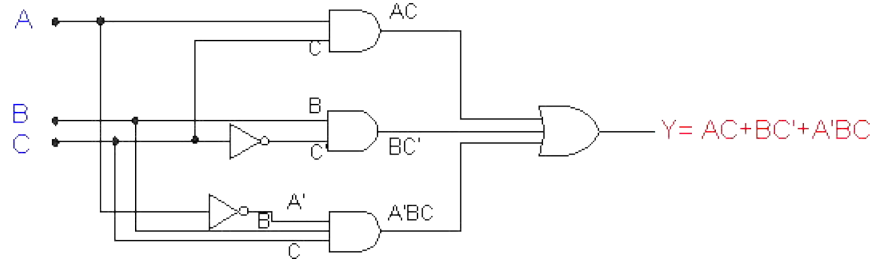
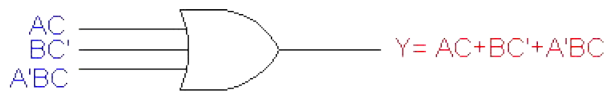
Gambar 4.3 Penentuan Level Output

4.3 Implementasi Rangkaian-Rangkaian dari Ekspresi Boolean

Apabila operasi dari suatu rangkaian didefinisikan oleh suatu ekspresi Boolean, maka suatu diagram rangkaian logika dapat diimplementasikan secara langsung dari ekspresi tersebut.

Misal, apabila diperlukan untuk membuat suatu rangkaian yang didefinisikan oleh output $Y = AC + BC' + A'BC$. Ekspresi Boolean ini mengandung tiga term (AC , BC' , $A'BC$), yang di OR-kan menjadi satu. Ini artinya bahwa ada OR gate tiga input yang masing-masing input sama dengan AC , BC' , dan $A'BC$.

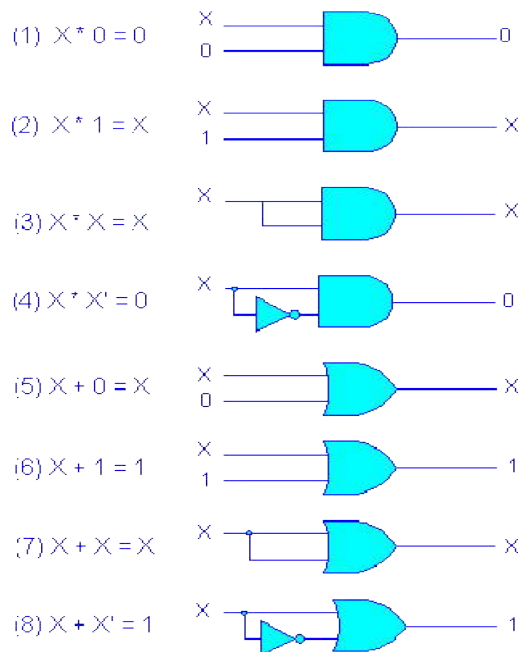
Setiap input OR gate adalah suatu term hasil operasi AND, yang berarti bahwa suatu AND gate dengan multi input yang sesuai dapat digunakan untuk menghasilkan tiap-tiap term ini. Ini ditunjukkan pada gambar 4.4.



Gambar 4.4 Membangun suatu rangkaian logika dari suatu ekspresi Boolean.

4.4 Teorema-teorema Boolean

Teorema-teorema (Hukum) Boolean dapat membantu untuk menyederhanakan ekspresi Boolean dan rangkaian-rangkaian logika.



Teorema Multivariabel

- (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) $x + xy = x$ [proof see below]
(15) $x + x'y = x + y$
-

Pembuktian teorema (14)

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

4.5 Teorema DeMorgan's

Dua teorema paling penting dari aljabar Boolean ditemukan oleh seorang matematikawan bernama DeMorgan. Teorema-teorema DeMorgan sangat berguna dalam menyederhanakan ekspresi-ekspresi aljabar Boolean. Dua teorema tersebut adalah :

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

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

Teorema (16) menunjukkan bahwa apabila jumlah OR dari dua variabel ($X + Y$) dikomplementasikan atau diinverskan, ini sama dengan apabila dua variable tersebut diinverskan sendiri-sendiri dan kemudian di-AND-kan menjadi satu. Dengan cara

lain dinyatakan, komplement dari suatu penjumlahan OR sama dengan perkalian AND dari komplement-komplementnya.

Teorema (17) menunjukkan bahwa apabila hasil kali dua variable dikomplementasikan, hasilnya adalah ekuivalen dengan mengkomplementasikan variable-variabel tersebut secara individual dan kemudian hasilnya di-OR-kan.

Contoh

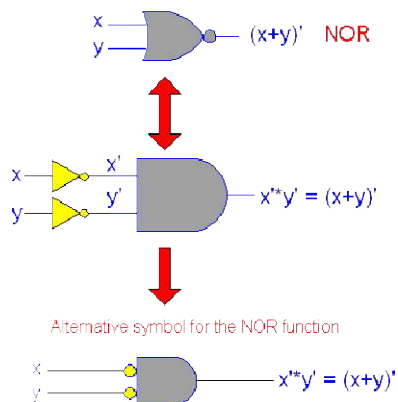
$$\begin{aligned}
 X &= [(A'+C) * (B+D)]' \\
 &= (A'+C)' + (B+D)' \text{ [dengan teorema (17)]} \\
 &= (A''*C') + (B'+D'') \text{ [dengan teorema (16)]} \\
 &= AC' + B'D
 \end{aligned}$$

Teorema DeMorgan's Tiga Variabel

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

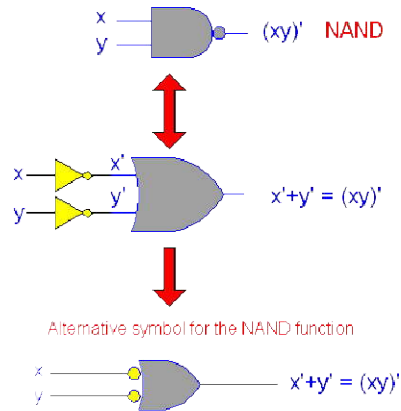
$$(19) (xyz)' = x' + y' + z'$$

Implikasi Teorema DeMorgan's



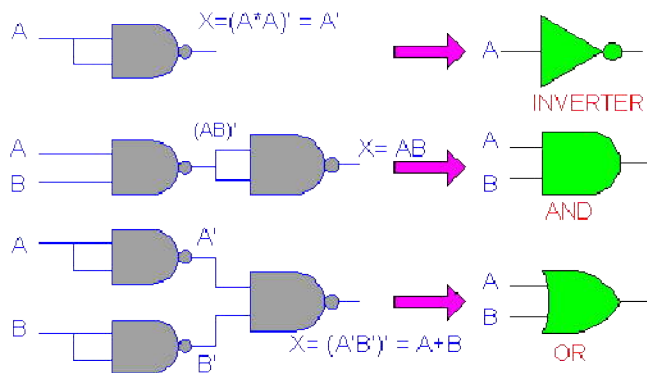
Untuk (16): $(x+y)' = x' * y'$

untuk (17): $(x*y)' = x' + y'$



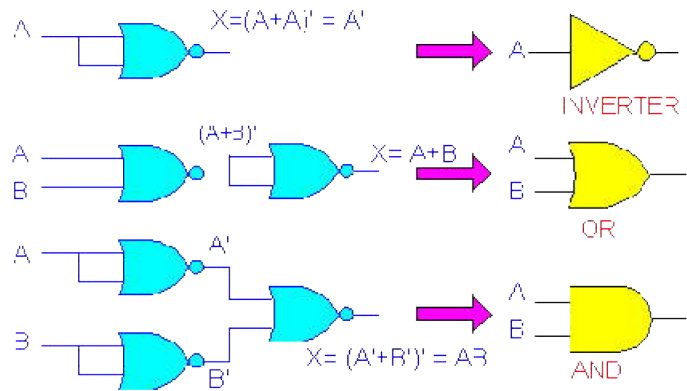
4.6 Keuniversalan dari NAND gate dan NOR gate

Adalah mungkin untuk mengimplementasikan suatu ekspresi Boolean hanya menggunakan NAND gate dan tanpa jenis gate lainnya. Ini dikarenakan NAND gate, dalam kombinasi yang benar dapat digunakan untuk melaksanakan setiap operasi Boolean yaitu OR, AND, dan INVERT. Ini ditunjukkan pada gambar 4.5



Gambar 4.5 Universal NAND gate

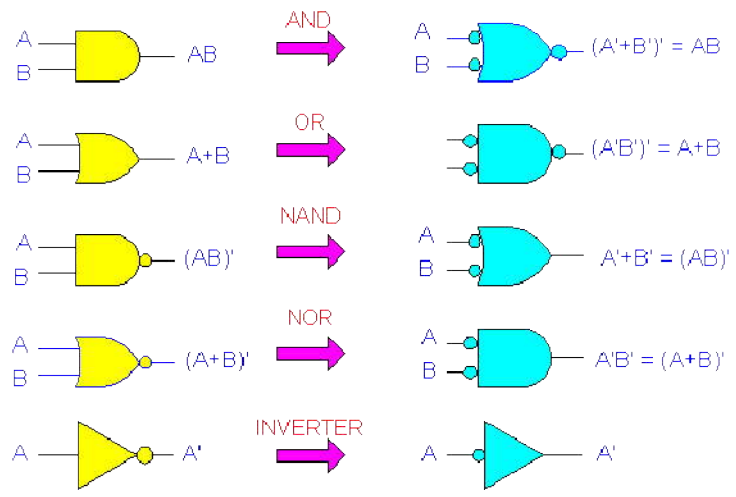
Dengan cara yang sama, dapat ditunjukkan bahwa NOR gate dapat disusun untuk melakukan setiap operasi Boolean.



Gambar 4.6 Universal NOR gate

4.7 Representasi Alternative Gate Logika

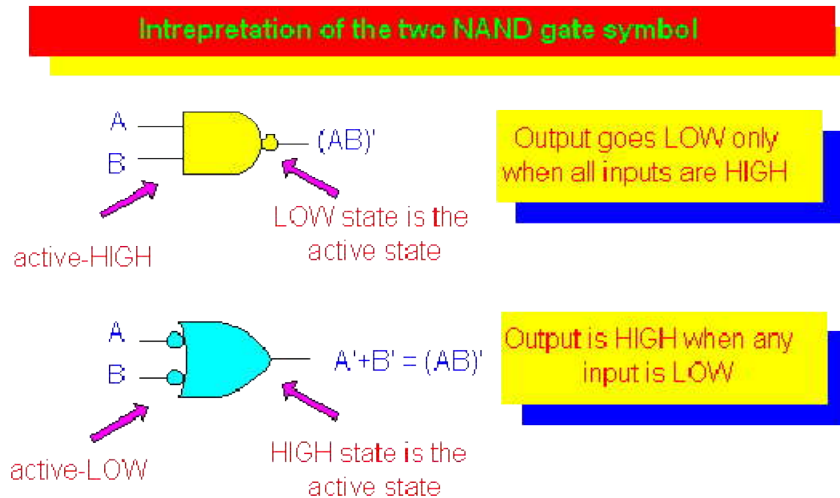
Sisi kiri dari ilustrasi di bawah ini merupakan symbol standar untuk setiap gate logika, dan sisi kanan menunjukkan symbol alternatif.



Gambar 4.7 Simbol Gate Logika Standar dan Simbol Ekuivalennya.

4.8 Konsep dari Level-Level Logika Aktif

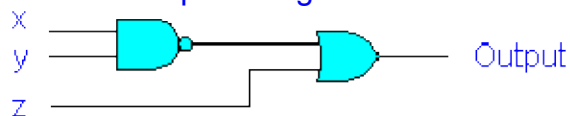
Apabila lintasan input atau output dari symbol rangkaian logika tidak ada invert-nya, maka lintasan tersebut dinyatakan sebagai aktif Tinggi. Apabila lintasan input atau output ada invert-nya, maka lintasan tersebut dinyatakan aktif Rendah.



Gambar 4.8 Interpretasi dari dua symbol NAND gate

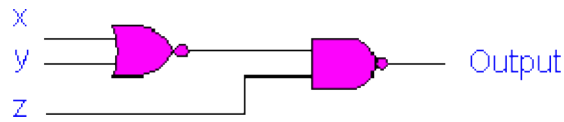
Soal Latihan

1. Persamaan output dari gambar berikut :



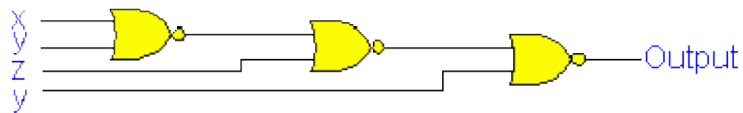
- $x'y'+z$
- $(x'+y')z$
- $x'y'z$
- $x'+y'+z$
- NA

2. Apa persamaan output gambar berikut :



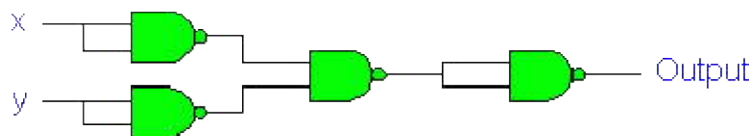
- $x+y+z$
- $x+y+z'$
- $x'y'z$
- $x'+y'+z'$
- NA

3. Apa persamaan output gambar berikut :



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

4. Rangkaian berikut ekivalen dengan gate ?



- AND
- OR
- NAND
- NOR
- None of the above

5. Fungsi di bawah ini yang sama dengan fungsi: $f=x+yz'$?

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

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

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

- NA

6. Beberapa fungsi logic biner dapat diimplementasikan hanya dengan :

- AND

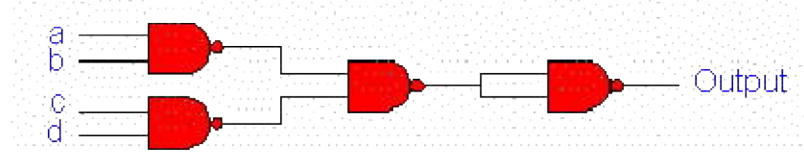
- OR

- NOT

- AA (anyone is sufficient)

- NAND

7. Fungsi dari rangkaian berikut adalah :



- $abcd$

- $ab+cd$

- $(a+b)(c+d)$

- $a+b+c+d$

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

8. $F=A'B+(C'+E)(D+F)$, gunakan de Morgan's theorem untuk mendapatkan F' .

- $ACE'+BCE'+D'F$

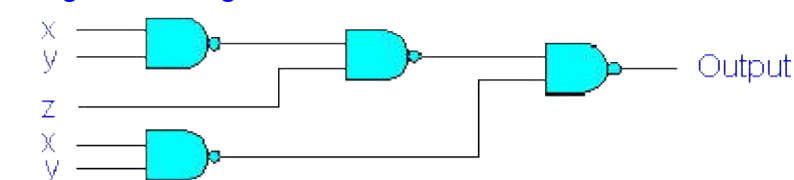
- $(A+B')(CE'D'F)$

- $A+B+CE'D'F$

- $ACE'+AD'F+B'CE'+B'D'F$

- NA

9. Fungsi dari rangkaian berikut adalah :



- $x'+y'+z'$

- $x+y+z$
- $x'z'+y'z'$
- $xy+z$
- z

10. Sederhanakan persamaan berikut : $\{[(AB)'C]'D\}'$

- $(A'+B')C+D'$
 - $(A+B')C'+D'$
 - $A'+(B'+C')D$
 - $A'+B'+C'+D'$
 - $A+B+C+D$
-