

Perhitungan Dot Vektor Dimensi Dua Input 4 bit dengan Rangkaian Digital Berdasarkan Aljabar Boolean

Edward Hendrata
(13505111)

*Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jalan Ganesha 10 Bandung 40132*

E-mail : if15111@students.if.itb.ac.id, edogenius@yahoo.com

Abstrak

Seiring dengan berkembangnya peradaban dunia saat ini, kepraktisan, cepat, konvergensi, dan efisiensi menjadi modal penting untuk *survive* dalam menghadapi persaingan global. Elemen-elemen tersebut dapat terpenuhi dengan menggunakan perangkat digital. Digitalisasi sudah menjadi kebutuhan bagi manusia saat ini. Beberapa perangkat analog sudah digantikan fungsinya menjadi perangkat digital. Oleh karena itu, dibutuhkanlah dasar pengetahuan untuk merancang aplikasi yang bersifat digital. Perangkat digital didasari oleh logika. Susunan logika yang terstruktur dapat dirancang menjadi suatu rangkaian gerbang logika yang menjadi dasar terciptanya suatu perangkat digital

1. Pendahuluan

1.1 Latar Belakang

Pada jaman sekarang ini, teknologi berkembang sangat pesat. Bermacam-macam alat dihasilkan Sekarang, hampir semua peralatan yang bekerja dengan tegangan listrik sudah menggunakan rangkaian digital. Saat ini rangkaian elektronika digital sudah bukan barang asing lagi. Rangkaian digital sudah ada di mana-mana dan bersinergi dengan rangkaian elektronika analog untuk membentuk rangkaian-rangkaian elektronika yang lebih cermat, cepat, dan tepat sasaran

Sebenarnya, sebuah rangkaian digital tidak harus selalu berupa rangkaian rumit dengan banyak komponen kecil seperti yang kita lihat di dalam komputer, handphone, ataupun kalkulator. Sebuah rangkaian dengan kerja sederhana yang menerapkan prinsip-prinsip digital, juga merupakan sebuah rangkaian digital.

Contoh rangkaian digital sederhana adalah rangkaian pengaman yang ditambahkan pada rangkaian kunci kontak sepeda motor atau mobil. Pada rangkaian

pengaman terdapat kontak (berupa relay atau transistor) yang aktivitasnya dikontrol oleh pemilik sepeda motor. Kontak pengaman ini harus dihubungkan seri dengan rangkaian kunci kontak. Akibatnya, walau kunci kontak terhubung, sepeda motor tidak dapat distarter jika kontak pengaman ini masih terbuka. Cara ini cukup manjur untuk menghindari pencurian sepeda motor.

Gerbang (gate) dalam rangkaian logika merupakan fungsi yang menggambarkan hubungan antara masukan dan keluaran. Untuk menyatakan gerbang-gerbang tersebut biasanya digunakan simbol-simbol tertentu. Ada beberapa standar penggambaran simbol. Salah satu standar simbol yang populer adalah MIL-STD-806B yang dikeluarkan oleh Departemen Pertahanan Amerika Serikat untuk keperluan umum pada bulan Februari 1962.

Untuk menunjukkan prinsip kerja tiap gerbang (atau rangkaian logika yang lebih kompleks) dapat digunakan beberapa cara. Cara yang umum dipakai antara lain adalah tabel kebenaran (truth table) dan diagram waktu (timing diagram). Karena merupakan rangkaian digital, tentu saja level kondisi

yang ada dalam tabel atau diagram waktu hanya dua macam, yaitu logika 0 (low, atau false) dan logika 1 (atau high, atau true). Bentuk tabel kebenaran sebenarnya sudah kita kenal di materi logika Matematika dalam mata pelajaran matematika SMA.

Kondisi lain yang mungkin ada adalah kondisi X (level bebas, bisa logika 1 atau 0), dan kondisi high impedance (impedansi tinggi). Kondisi X biasanya ada di masukan gerbang dan menyatakan bahwa apa pun logika masukannya (logika 0 atau 1) tidak akan mempengaruhi logika keluaran yang dihasilkan. Kondisi impedansi tinggi pada suatu titik (point) menunjukkan titik yang bersangkutan diisolasi dari rangkaian lain, sehingga tidak ada logika yang akan mempengaruhi titik tersebut

gerbang dan rangkaian logika juga dapat diimplementasikan dalam bentuk rangkaian dioda, transistor, ataupun rangkaian terpadu yang disebut integrated circuit (IC). Dengan semakin majunya teknologi pembuatan komponen mikro-elektronika, perkembangan komponen IC untuk rangkaian digital menjadi pesat.

IC logika jenis TTL (Transistor-Transistor Logic) dan CMOS (Complementary Metal Oxide Semiconductor) cukup populer di kalangan masyarakat penggemar elektronika, termasuk siswa-siswa SMP dan SMA. Walaupun sudah mulai berkurang, jenis IC tersebut masih banyak digunakan hingga saat ini.

Dalam mengimplementasikan rangkaian digital, kita juga dapat menggunakan FPGA (Field Programmable Gate Array). Komponen yang satu ini mulai populer di kalangan mahasiswa Jurusan Teknik Elektro. FPGA mempunyai banyak kelebihan dibanding jenis-jenis PLD sebelumnya. Di antaranya adalah jenis dan jumlah gerbangnya yang sangat banyak (ribuan hingga ratusan ribu), kecepatannya yang sangat tinggi, mudah diprogram, dan dapat diprogram berkali-kali.

Pembuatan rangkaian digital dengan FPGA cukup menyenangkan dan menarik karena biasanya tersedia software yang multifungsi. Dengan software tersebut, kita dapat merancang dan mensimulasi rangkaian di komputer PC. Setelah itu me-loading rangkaian ke chip FPGA melalui port paralel LPT1. Perancangan rangkaian

dapat kita lakukan dengan cara skematis, yang menggunakan simbol-simbol layaknya menggambar rangkaian digital di kertas. Atau dengan bahasa VHDL (Visual Hardware Description Language) dan Verilog yang lebih sulit.

Kekurangan FPGA adalah sifat program koneksinya yang masih volatile (menguap), sehingga jika aliran suplai terputus, program koneksinya akan hilang. Namun, hal itu dapat diatasi dengan menambahkan ROM pada rangkaiannya. Saat ini sebuah chip FPGA dapat dibeli dengan harga berkisar ratusan ribu, dibandingkan dengan chip IC TTL atau CMOS yang hanya beberapa ribu rupiah. Bisa dikatakan masih mahal untuk ukuran kantong kebanyakan siswa atau mahasiswa. Namun, agar tidak ketinggalan teknologi, rasanya harga itu cukup murah, apalagi mengingat segala kelebihan yang dimilikinya.

1.2 Rumusan Masalah

Bagaimana cara menghitung dot matriks melalui rangkaian digital berdasarkan aljabar boolean?

1.3 Tujuan

Tujuan dari pembuatan makalah ini adalah untuk memenuhi tugas akhir mata kuliah Matematika Diskrit

2. Landasan Teori

Vektor

Vektor di dalam n-Ruang Definisi : Jika n adalah sebuah integer positif, sebuah n -grup topel adalah sekuens dari n bilangan real (a_1, a_2, \dots, a_n) . Set dari semua grup yang terdiri dari n - grup topel dinamakan n -ruang dan dituliskan sebagai R_n .

Jika $n = 2$ atau 3 , sudah menjadi kebiasaan untuk menggunakan istilah grup pasangan dan grup dari tiga secara respektif, daripada 2-grup topel atau 3- grup topel. Ketika $n = 1$, setiap $n -$ grup topel terdiri dari satu bilangan real, sehingga R_1 bisa dilihat sebagai set dari bilangan real. Kita akan menuliskan R daripada R_1 pada set ini.

Mungkin kita telah mempelajari dalam bahan 3-ruang symbol dari (a_1, a_2, a_3) mempunyai dua interpretasi geometris yang berbeda : ini bisa diinterpretasikan sebagai titik, yang dalam kasus ini a_1, a_2, a_3 merupakan koordinat, atau ini bisa

diinterpretasikan sebagai vector, dimana a_1, a_2, a_3 merupakan komponen vector. Selanjutnya kita bisa melihat bahwa n – grup topel (a_1, a_2, \dots, a_n) bisa dilihat sebagai antara sebuah “poin umum” atau “vector umum”- perbedaan antara keduanya tidak penting secara matematis. Dan juga kita bisa menjelaskan 5- topel $(-2, 4, 0, 1, 6)$ antara poin dalam R^5 atau vector pada R^5 .

$\mathbf{u}_1 = \mathbf{v}_1 \quad \mathbf{u}_2 = \mathbf{v}_2 \quad \mathbf{u}_n = \mathbf{v}_n$
 Penjumlahan $\mathbf{u} + \mathbf{v}$ didefinisikan oleh

$\mathbf{u} + \mathbf{v} = (u_1 + v_1, u_2 + v_2, \dots, u_n + v_n)$
 Dan jika k adalah konstanta scalar, maka perkalian scalar $k\mathbf{u}$ didefinisikan oleh

$k\mathbf{u} = (k u_1, k u_2, \dots, k u_n)$
 Operasi dari penambahan dan perkalian scalar dalam definisi ini disebut operasi standar untuk R^n Vektor nol dalam R^n didenotasikan oleh $\mathbf{0}$ dan difeniskan ke vektor

$\mathbf{0} = (0, 0, \dots, 0)$
 Jika $\mathbf{u} = (u_1, u_2, \dots, u_n)$ dalam setiap vector dalam R^n , maka negative (atau invers aditif) dari \mathbf{u} dituliskan oleh $-\mathbf{u}$ dan dijelaskan oleh

$-\mathbf{u} = (-u_1, -u_2, \dots, -u_n)$
 Perbedaan dari vector dalam R^n dijelaskan oleh

$\mathbf{v} - \mathbf{u} = \mathbf{v} + (-\mathbf{u})$
 atau, dalam istilah komponen,

$\mathbf{v} - \mathbf{u} = (v_1 - u_1, v_2 - u_2, \dots, v_n - u_n)$
Sifat-sifat dari vektor dalam R^n
 jika $\mathbf{u} = (u_1, u_2, \dots, u_n)$,
 $\mathbf{v} = (v_1, v_2, \dots, v_n)$, dan
 $\mathbf{w} = (w_1, w_2, \dots, w_n)$ adalah vektor dalam R^n sedangkan k dan m adalah skalar, maka :

- (a) $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$
- (b) $\mathbf{u} + \mathbf{0} = \mathbf{0} + \mathbf{u} = \mathbf{u}$
- (c) $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$
- (d) $\mathbf{u} + (-\mathbf{u}) = \mathbf{0}$; berarti, $\mathbf{u} - \mathbf{u} = \mathbf{0}$
- (e) $k(m\mathbf{u}) = (km)\mathbf{u}$
- (f) $k(\mathbf{u} + \mathbf{v}) = k\mathbf{u} + k\mathbf{v}$
- (g) $(k + m)\mathbf{u} = k\mathbf{u} + m\mathbf{u}$
- (h) $1\mathbf{u} = \mathbf{u}$

Perkalian *dot product* $\mathbf{u} \cdot \mathbf{v}$ didefinisikan sebagai

$$\mathbf{u} \cdot \mathbf{v} = u_1v_1 + u_2v_2 + \dots + u_nv_n$$

Gerbang Logika

Gerbang logika atau gerbang logik adalah suatu entitas dalam elektronika dan matematika boolean yang mengubah satu atau beberapa masukan logik menjadi sebuah sinyal keluaran logik. Gerbang logika terutama diimplementasikan secara elektronis menggunakan dioda atau transistor, akan tetapi dapat pula dibangun menggunakan susunan komponen-komponen yang memanfaatkan sifat-sifat elektromagnetik (*relay*), cairan, optik dan bahkan mekanik.

FPGA

field programmable gate array (FPGA) berbentuk chip IC. FPGA tergolong dalam jenis ASIC (application-specific IC). Jenis lain yang termasuk ASIC adalah PLD (programmable logic device). Sementara IC TTL, CMOS, mikroprosesor, dan mikrokontroler termasuk dalam jenis IC standar.

Sebuah chip FPGA mengandung puluhan hingga puluhan ribu sel logika. Tiap sel dapat dilihat sebagai komponen standar. Tiap-tiap sel logika mempunyai beberapa jumlah masukan. Biasanya berjumlah dua hingga 10 masukan. Sel-sel logika juga mempunyai keluaran yang berjumlah satu atau dua, tergantung dari jenis fungsinya.

FPGA juga dapat dipandang sebagai kumpulan blok, di mana tiga elemen penyusunnya adalah CLB (combinational logic block), IOB (input/output block), dan interkoneksi. CLB menghasilkan elemen fungsional untuk menyusun rangkaian logika yang diinginkan. IOB berfungsi sebagai antarmuka (interface) antara pin-pin terminal chip dengan kawat penghubung dalam chip IC. Blok interkoneksi, yang berupa kawat-kawat dan sakelar penghubung, mengelilingi blok-blok CLB.

Sebelum diprogram, tiap blok tidak terikat satu dengan lain. Keterikatan hanya terjadi jika program dimasukkan untuk membuat koneksi antara satu blok dan blok lainnya. Koneksi dilakukan dengan menggunakan interkoneksi berupa sambungan-sambungan atau sakelar-sakelar (dalam FPGA) yang disusun dalam bentuk matriks.

FPGA tidak hanya mengakomodasi kebutuhan akan gerbang dasar, tetapi juga rangkaian logika kombinasi sederhana seperti decoder, multiplexer, adder atau rangkaian kombinasi lain yang lebih rumit. Rangkaian logika kombinasi dapat disusun dari sekumpulan gerbang dan multiplexer atau dalam bentuk memori look-up table (LUT) yang rangkaianannya menjadi lebih sederhana.

Selain rangkaian logika kombinasi, FPGA juga dapat mengakomodasi kebutuhan akan rangkaian logika sekuensial seperti flip-flop (JK, RS, T, D) hingga counter, shift register. Biasanya, rangkaian logika kombinasi atau sekuensial standar telah disediakan dalam file library sehingga memudahkan perancang dalam membangun rangkaianannya.

Decoder

Decoder adalah suatu rangkaian dengan n buah input dan 2^n buah output. Output decoder hanya dapat aktif salah satu, ditentukan oleh kombinasi input yang terjadi.

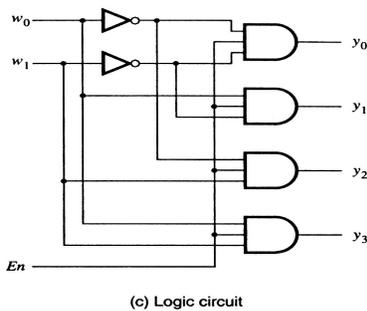
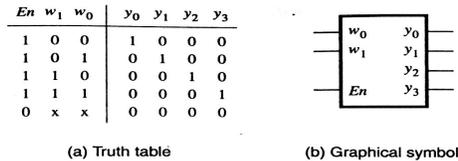
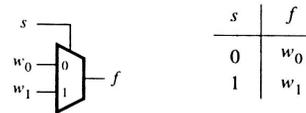


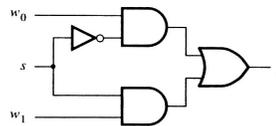
Figure 6.16 A 2-to-4 decoder.

sehingga muncul di saluran keluarannya.

- λ Untuk menyeleksi saluran masukan mana yang akan diambil, sebuah MUX memerlukan saluran pemilih yang terdiri dari n saluran.

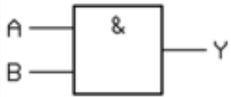
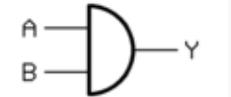
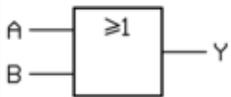
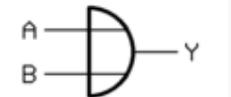
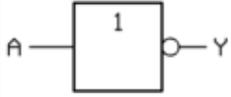
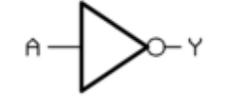
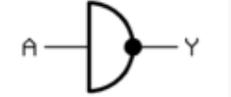


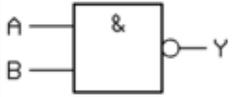
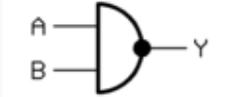
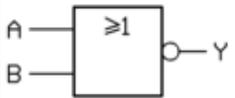
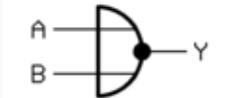
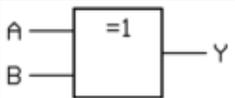
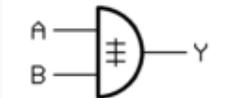
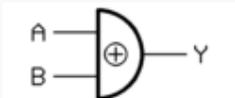
(a) Graphical symbol (b) Truth table



(c) Sum-of-products circuit

- λ Multiplexer (MUX) adalah suatu rangkaian yang berfungsi untuk menyeleksi satu buah masukan dari kemungkinan 2^n buah masukan

Nama	Fungsi	Lambang dalam rangkaian			Tabel kebenaran															
		IEC 60617-12	US-Norm	DIN 40700 (sebelum 1976)																
Gerbang-AND (AND)	$Y = A \wedge B$ $Y = A \cdot B$ $Y = AB$				<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	Y	0	0	0	0	1	0	1	0	0	1	1	1
A	B	Y																		
0	0	0																		
0	1	0																		
1	0	0																		
1	1	1																		
Gerbang-OR (OR)	$Y = A \vee B$ $Y = A + B$				<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	1
A	B	Y																		
0	0	0																		
0	1	1																		
1	0	1																		
1	1	1																		
Gerbang-NOT (NOT, Gerbang-komplemen, Pembalik(<i>Inverter</i>))	$Y = \bar{A}$ $Y = \neg A$				<p style="text-align: center;">\</p> <table border="1"> <thead> <tr> <th>A</th> <th>Y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	Y	0	1	1	0									
A	Y																			
0	1																			
1	0																			

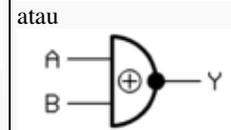
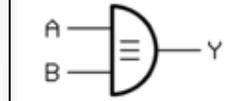
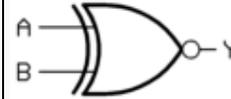
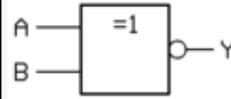
<p>Gerbang-NAND (Not-AND)</p>	$Y = \overline{A \wedge B}$ $Y = A \overline{B}$ $Y = \overline{A} B$				<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	Y	0	0	1	0	1	1	1	0	1	1	1	0
A	B	Y																		
0	0	1																		
0	1	1																		
1	0	1																		
1	1	0																		
<p>Gerbang-NOR (Not-OR)</p>	$Y = \overline{A \vee B}$ $Y = A \overline{B}$ $Y = \overline{A} B$				<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	0
A	B	Y																		
0	0	1																		
0	1	0																		
1	0	0																		
1	1	0																		
<p>Gerbang-XOR (Antivalen, Exclusive-OR)</p>	$Y = A \underline{\vee} B$ $Y = A \oplus B$			<p>atau</p>  	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	0
A	B	Y																		
0	0	0																		
0	1	1																		
1	0	1																		
1	1	0																		

Gerbang-XNOR
(Ekuivalen, Not-Exclusive-OR)

$$Y = \overline{A \vee B}$$

$$Y = A \overline{\vee} B$$

$$Y = \overline{A \oplus B}$$



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

3. Pembahasan

Persoalan

Masalah yang dihadapi pada makalah ini adalah menghitung dot product dari dua buah vector berdimensi 2 4 bit. Misalkan ada vektor $U = (3, 5)$ dan $V = (2,1)$. Untuk perhitungan secara langsung, berdasarkan landasan teori dapat dilakukan dengan cara $(3 \times 2) + (5 \times 1) = 11$. Hasil dot kedua vector tersebut adalah 11. Pada makalah ini, hasil dot product tersebut akan didapatkan melalui rangkaian gerbang logika.

Batasan Persoalan

Komponen dari vector persoalan harus bernilai ≥ 0 . Input negative tidak dapat diproses karena rangkaian yang digunakan adalah rangkaian untuk memproses bentuk unsigned magnitude.

Input Rangkaian

Input berupa biner dari kedua vector tersebut. Oleh karena itu, akan ada empat input yaitu u_1, u_2, v_1, v_2 , dalam bentuk biner sehingga menghasilkan keseluruhan total 16 input dalam bentuk bit.

Output Rangkaian

Output yang akan dihasilkan pada rangkaian gerbang logika tersebut adalah hasil dot product dari kedua vector tersebut. Hasil akan dioutputkan ke dalam seven segmen sebanyak 3 kali. Output pertama, kedua, dan ketiga akan digabungkan dan membentuk 3 digit bilangan. Bilangan tersebut merupakan hasil dari dot product kedua vektor dalam bentuk hexadesimal.

Prinsip dasar

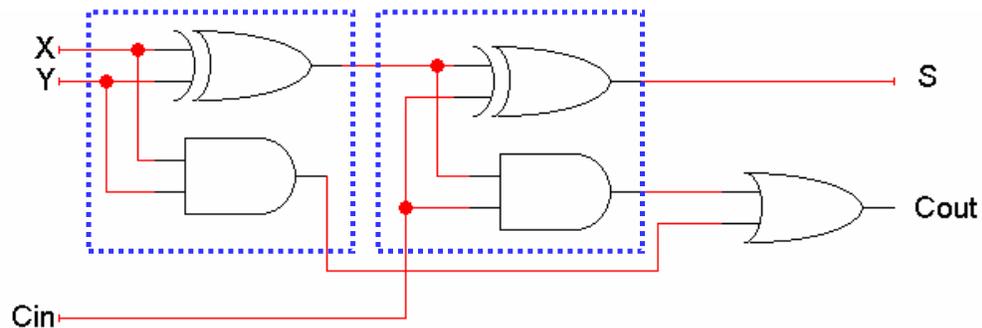
Prinsip dasar dari pembuatan rangkaian adalah menerjemahkan operasi matematika yang diperlukan ke dalam bentuk rangkaian gerbang logika. Input yang diterima diproses ke dalam rangkaian operasi matematika kemudian hasil yang diperoleh ditampilkan/divisualkan ke dalam seven segment.

Proses Rangkaian

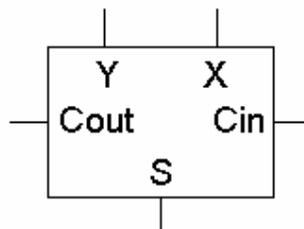
Karena perhitungan dilakukan dengan menggunakan rangkaian operasi matematika, maka berikut ini adalah beberapa rangkaian operasi matematika yang diperlukan :

Operasi Penjumlahan

Rangkaian Full adder untuk 1 bit :



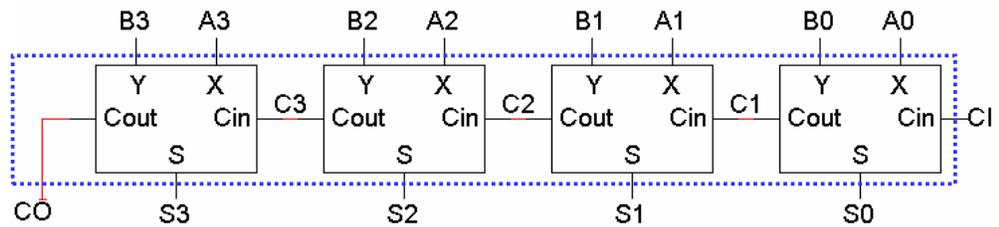
untuk memudahkan penggambaran maka untuk selanjutnya, rangkaian full adder 1 bit akan disimbolkan seperti ini:



S merupakan hasil penambahan, sedangkan carry dioutputkan menjadi Cout. Karena input yang digunakan pada persoalan adalah 4 bit, maka diperlukan full adder

untuk 4 bit. Berikut simbol rangkaian full adder 4 bit yang dibuat berdasarkan susunan beberapa full adder 1 bit :

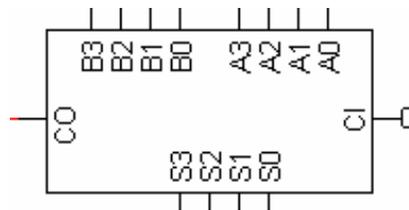
Rangkaian Full Adder untuk 4 bit :



4 bit dari input pertama adalah A0, A1, A2, A3, dan untuk input kedua adalah B0, B1, B2, B3. C pada rangkaian merupakan carry dari penambahan bit sebelumnya. Output yang dihasilkan berupa 4 bit yaitu,

S0, S1, S2, S3. Untuk memudahkan penggambaran, maka untuk selanjutnya rangkaian Full Adder untuk penjumlahan disimbolkan sebagai berikut :

Simbol Full Adder 4 bit :



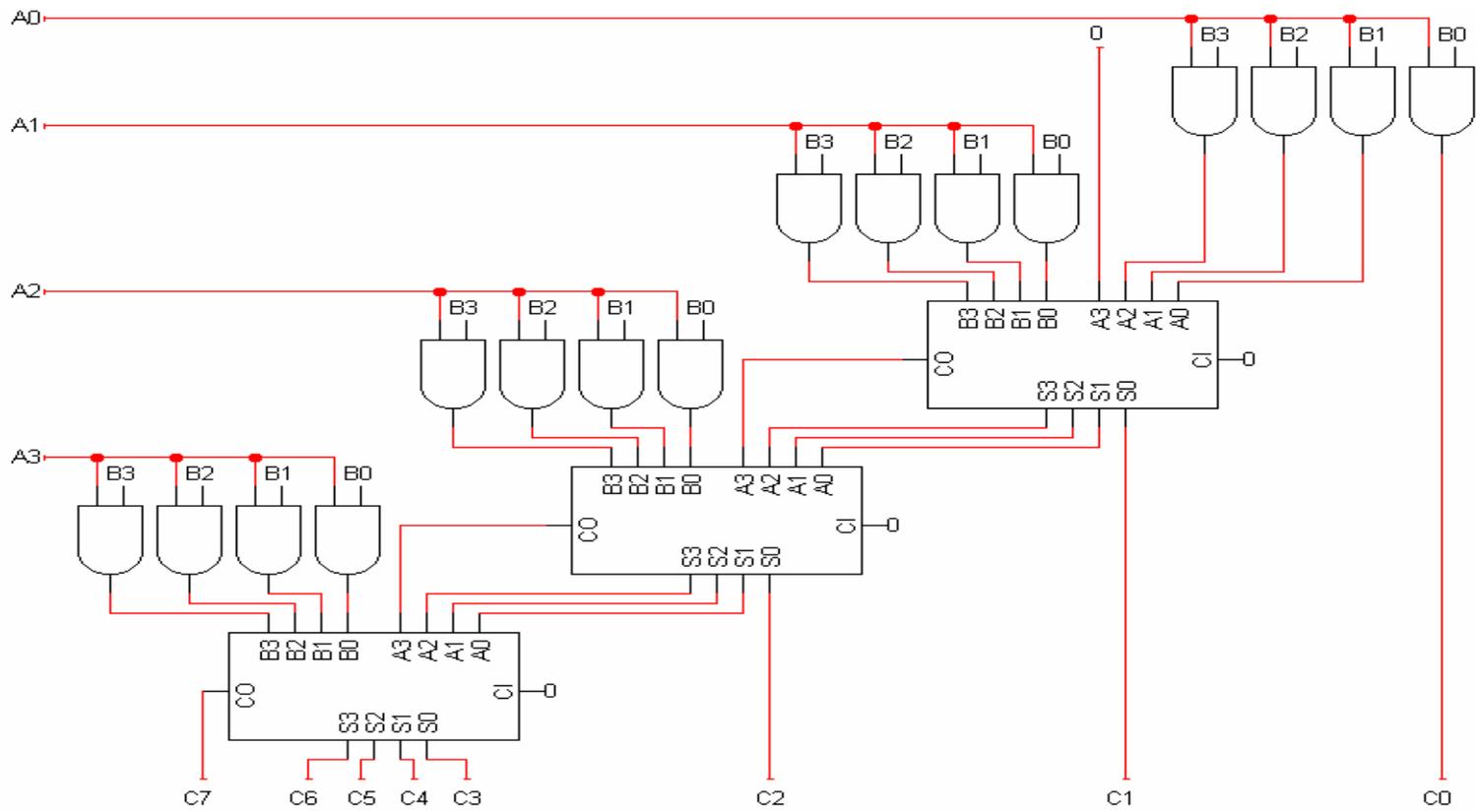
Simbol Full Adder tersebut dapat digeneralisir untuk operasi penjumlahan dengan jumlah bit yang berbeda

Operasi Perkalian

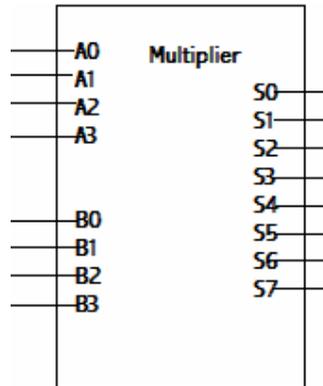
Pada rangkaian operasi perkalian 4 bit dibutuhkan juga rangkaian full adder.

Perkalian 4 bit dengan 4 bit akan menghasilkan output 8 bit. Berikut ini adalah rangkaian dari operasi perkalian 4 bit dengan input pertama A0, A1, A2, A3, dan input kedua B0, B1, B2, B3 akan dihasilkan output 8 bit yaitu C0, C1, C2, C3, C4, C5, C6, C7.

Rangkaian Multiplier untuk 4 bit :



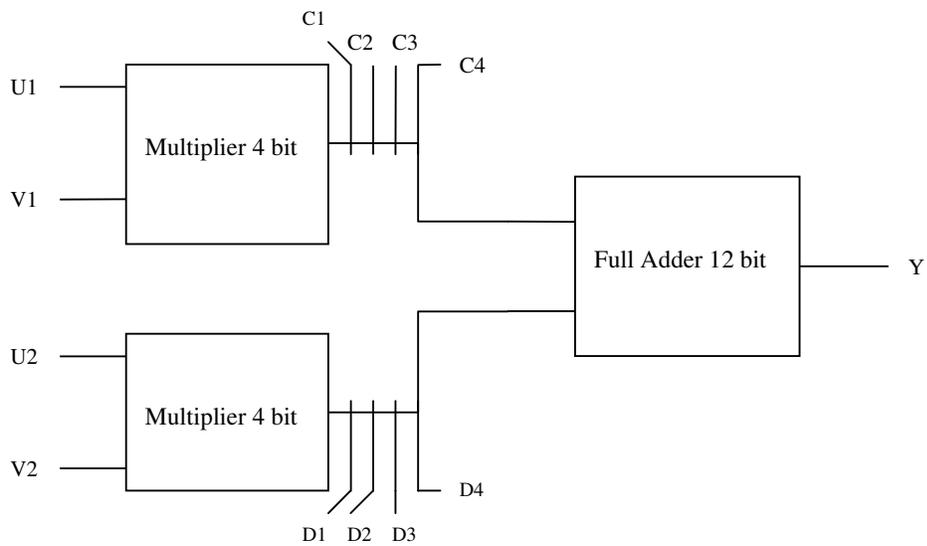
Untuk selanjutnya rangkaian multiplier 4 bit akan dilambangkan sebagai berikut :



Dengan input pertama A0, A1, A2, A3, dan input kedua B0, B1, B2, B3, dan menghasilkan output S0, S1, S2, S3.

Dengan sudah adanya rangkaian gerbang logika untuk operasi matematika dasar yang diperlukan, maka operasi dot vector 2 dimensi 4 bit dapat dirangkai sebagai berikut :

Rangkaian dot vector 2 dimensi 4 bit



Proses Perhitungan :

- Input U1, U2, V1, dan V2 merupakan input biner dengan panjang 4 bit
- Input U1 dikalikan dengan V1 dengan menggunakan multiplier 4 bit yang akan menghasilkan output 8 bit
- Input U2 dikalikan dengan V2 dengan menggunakan multiplier 4 bit yang akan menghasilkan output 8 bit
- Kedua output tersebut ditambahkan jumlah bitnya dengan input tambahan masing-masing 4 bit yaitu C1, C2, C3, C4 dan D1, D2, D3, D4 sehingga output yang semula sebesar 8 bit akan menjadi 12 bit.
- Kedua output 12 bit tersebut kemudian dijumlahkan dengan menggunakan Full Adder 12 bit yang akan menghasilkan satu output 12 bit.
- Output Y merupakan hasil dot vektor dari U dengan V dengan panjang 12 bit.

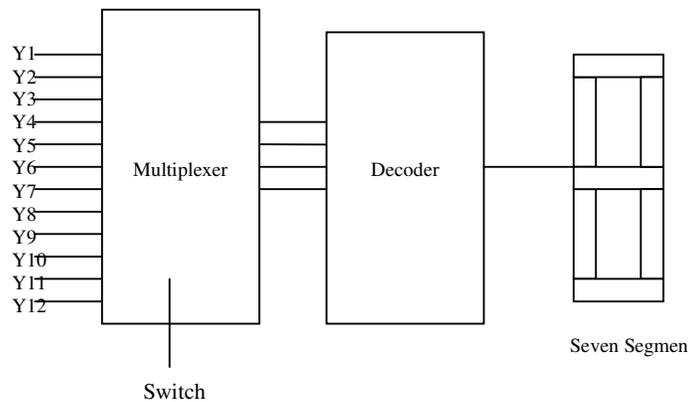
Visualisasi Output

Output 12 bit yang dihasilkan akan ditampilkan ke dalam seven segmen. Bilangan yang ditampilkan pada seven segmen merupakan bilangan hexadesimal.

Proses Pengoutputan

Output 12 bit yang dihasilkan dari proses perhitungan kemudian dihubungkan pada multiplexer yang dapat menyeleksi suatu input menjadi beberapa bagian kecil dari input tersebut. Pada kasus ini akan dapat dihasilkan 3 output dengan masing-masing sebesar 4 bit melalui switch. Masing-masing output 4 bit tersebut kemudian dihubungkan pada decoder untuk kemudian diterjemahkan ke dalam seven segmen. Decoder merupakan suatu rangkaian yang berfungsi untuk mengubah input 4 bit menjadi 7 bit untuk kemudian ditampilkan ke dalam seven segmen. Tiga angka yang dihasilkan masing-masing decoder kemudian disatukan sehingga menjadi suatu bilangan. Berikut merupakan gambaran dari proses pengoutputan :

Proses Pengoutputan



4. Kesimpulan

Aljabar boolean merupakan dasar dari teknologi digital. Dengan aljabar boolean, rangkaian logika dapat digunakan untuk menyelesaikan masalah baik dasar maupun kompleks secara digital. Salah satunya adalah untuk menyelesaikan perhitungan dot vektor berdimensi 2 input 4 bit secara digital. Berdasarkan kajian dan analisis dari persoalan tersebut, beberapa kesimpulan dapat diambil yaitu :

1. Operasi matematika dan logika dapat diselesaikan dengan rangkaian gerbang logika
2. Rangkaian digital melakukan proses dengan input berupa 0 atau 1.

3. Keluaran yang dihasilkan juga berupa 0 atau 1.
4. Untuk memvisualisasikan hasil yang masih berupa 0 atau 1, dapat digunakan seven segmen.
5. Aljabar boolean dapat digunakan dalam perancangan rangkaian digital
6. Proses logika dilakukan secara bit per bit
7. Perhitungan dot product vektor dimensi 2 input 2 bit dapat diselesaikan dengan menggunakan rangkaian logika.

5. Daftar Pustaka

Brown, Stephen & Zvonko Vranesic. 2005. *Fundamental of Digital Logic with VHDL Design*, 4th. McGraw-Hill.

Munir, Rinaldi. 2006. *Diktat Kuliah IF2152 Matematika Diskrit*, edisi keempat. Teknik Informatika ITB.

Partasubita, Santika. 2006. *Slide Kuliah Organisasi dan Arsitektur Komputer 1*. Teknik Informatika ITB.

http://id.wikipedia.org/wiki/Aljabar_linier

http://id.wikipedia.org/wiki/Gerbang_logika