

## MODUL 3

### Operasi Aritmatik

**KELAS** : X TKJ  
**GURU MAPEL** : NurRahmat, S.Kom  
**Materi pokok** : Operasi Aritmatik  
**Alokasi waktu** : Jam Pelajaran

#### A. Tujuan pembelajaran :

- 1 Siswa mengenal jenis operasi aritmatik
- 2 Siswa terampil melakukan percobaan ALU

#### B. Materi Pembelajaran

Operasi Bilangan Biner

#### PENJUMLAHAN BINER

Seperti bilangan desimal, bilangan biner juga dijumlahkan dengan cara yang sama. Pertama-tama yang harus dicermati adalah aturan pasangan digit biner berikut :

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0 \text{ } \square \text{ menyimpan 1}$$

sebagai catatan bahwa jumlah dua yang terakhir adalah :

$$1 + 1 + 1 = 1 \text{ } \square \text{ dengan menyimpan 1}$$

Dengan hanya menggunakan penjumlahan-penjumlahan pada slide sebelumnya, kita dapat melakukan penjumlahan biner seperti ditunjukkan di bawah ini :

1 1111 --> “simpanan 1” ingat kembali aturan di atas

01011011 --> bilangan biner untuk 91

01001110 --> bilangan biner untuk 78

-----+

10101001 --> Jumlah dari  $91 + 78 = 169$

Kita akan menghitung penjumlahan biner yang terdiri dari 5 bilangan:

11101 bilangan <sup>1)</sup>

10110 bilangan <sup>2)</sup>

1100 bilangan <sup>3)</sup>

11011 bilangan <sup>4)</sup>

1001 bilangan <sup>5)</sup>

----- +

Untuk menjumlahkannya, kita hitung berdasarkan aturan yang berlaku, dan untuk lebih mudahnya perhitungan dilakukan bertahap

$$\begin{array}{rcl}
 11101 & \text{bilangan } ^1) & \\
 10110 & \text{bilangan } ^2) & \\
 \hline
 110011 & & \\
 1100 & \text{bilangan } ^3) & \\
 \hline
 111111 & & \\
 11011 & \text{bilangan } ^4) & \\
 \hline
 1011010 & & \\
 1001 & \text{bilangan } ^5) & \\
 \hline
 1100011 & \text{Jumlah Akhir} & 
 \end{array}$$

Apakah benar hasil penjumlahan tersebut?

$$\begin{array}{rcl}
 11101 & \text{bilangan } ^1) & \\
 10110 & \text{bilangan } ^2) & \\
 1100 & \text{bilangan } ^3) & \\
 11011 & \text{bilangan } ^4) & \\
 1001 & \text{bilangan } ^5) & \\
 \hline
 1100011 & \text{Jumlah Akhir} & 
 \end{array}$$

Mari Buktikan dengan merubah biner ke desimal.

$$\begin{array}{rcl}
 11101 & = & 29 \\
 10110 & = & 22 \\
 1100 & = & 12 \\
 11011 & = & 27 \\
 1001 & = & 9 \\
 \hline
 1100011 & = & 99 \text{ Sesuai!}
 \end{array}$$

## PENGURANGAN BINER

Untuk memahami konsep pengurangan biner, kita harus mengingat kembali perhitungan desimal (angka biasa), kita mengurangkan digit desimal dengan digit desimal yang lebih kecil. Jika digit

desimal yang dikurangkan lebih kecil daripada digit desimal yang akan dikurangi, maka terjadi “konsep peminjaman”. Digit tersebut akan meminjam 1 dari digit sebelah kirinya.

Bentuk Umum pengurangan sebagai berikut :

$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

$$0 - 1 = 1 \text{ } \square \text{ meminjam '1' dari digit disebelah kirinya}$$

Contoh :

1111011      desimal 123

101001      desimal 41

----- -

1010010      desimal 82

Pada contoh di atas tidak terjadi “konsep peminjaman”.

Perhatikan contoh berikut!

0      kolom ke-3 menjadi ‘0’, sudah dipinjam

111101      desimal 61

10010      desimal 18

----- -

101011      Hasil pengurangan akhir 43

Pada soal yang kedua ini kita pinjam ‘1’ dari kolom 3, karena ada selisih 0-1 pada kolom ke-2  
Lalu bagaimana jika saya tidak dapat meminjam 1 dari kolom berikutnya karena kolom tersebut berupa bilangan ‘0’?

Untuk membahas hal itu mari kita beri bandingkan jika hal ini terjadi pada bilangan desimal. Mari kita hitung desimal  $800046 - 397261$ !

7999

8000<sup>1</sup>46

3972 61

----- -

4027 05

Perhatikan bahwa kita meminjam 1 dari kolom keenam untuk kolom kedua, karena kolom ketiga, keempat dan kolom kelima adalah nol. Setelah meminjam, kolom ketiga, keempat, dan kelima menjadi:  
 **$10 - 9 = 1$**

Hal ini juga berlaku dalam pengurangan biner, kecuali bahwa setelah meminjam kolom nol akan mengandung:  **$10 - 1 = 1$**

Sebagai contoh pengurangan bilangan biner  $110001 - 1010$  akan diperoleh hasil sebagai berikut:

$$\begin{array}{r}
 1100^101 \\
 10\ 10 \\
 \hline
 1001\ 11
 \end{array}$$

## PERKALIAN BINER

Metode yang digunakan dalam perkalian biner juga pada dasarnya sama dengan perkalian desimal, akan terjadi pergeseran ke kiri setiap dikalikan 1 bit pengali. Setelah proses perkalian masing-masing bit pengali selesai, dilakukan penjumlahan masing-masing kolom bit hasil.

Contoh :

$$\begin{array}{r}
 1101 \\
 1011 \\
 \hline
 1101 \\
 1101 \\
 0000 \\
 1101 \\
 \hline
 10001111
 \end{array}$$

Perkalian bilangan biner dapat dilakukan seperti pada perkalian bilangan desimal. Sebagai contoh, untuk mengalikan  $1110_2 = 14_{10}$  dengan  $1101_2 = 13_{10}$  langkah-langkah yang harus ditempuh adalah:

Biner	Desimal
1 1 1 0	1 4
1 1 0 1	1 3
-----x	----x
1 1 1 0	4 2
0 0 0 0	1 4
1 1 1 0	
1 1 1 0	
-----+	-----+
1 0 1 1 0 1 1 0	1 8 2

Perkalian juga bisa dilakukan dengan menambahkan bilangan yang dikalikan ke bilangan itu sendiri sebanyak bilangan pengali.

Contoh barusan, hasilnya akan sama dengan jika kita menambahkan  $1112$  ke bilangan itu sendiri sebanyak  $1101$  atau 13 kali.

## PEMBAGIAN BINER

Serupa dengan perkalian, pembagian pada bilangan biner juga menggunakan metode yang sama dengan pembagian desimal. Bit-bit yang dibagi diambil bit per bit dari sebelah kiri. Apabila nilainya lebih dari bit pembagi, maka bagilah bit-bit tersebut, tetapi jika setelah bergeser 1 bit nilainya masih dibawah nilai pembagi maka hasilnya adalah 0.

Contoh :

$$\begin{array}{r}
 \text{Biner} \\
 011 \overline{) 1001} \\
 \underline{- 011} \phantom{0} \\
 0011 \\
 \underline{- 011} \\
 0
 \end{array}
 = 3_{10} \leftarrow \text{hasil bagi}$$

yang dibagi

pembagi

Pembagian pada sistem bilangan biner dapat dilakukan sama seperti contoh pembagian sistem bilangan desimal. Sebagai contoh, untuk membagi 110011 (disebut bilangan yang dibagi) dengan 1001 (disebut pembagi), langkah-langkah berikut yang perlu dilakukan.

$$\begin{array}{r}
 101 \text{ Hasil} \\
 \hline
 1001 \overline{) 110011} \\
 \underline{1001} \phantom{00} \\
 001111 \\
 \underline{1001} \phantom{00} \\
 110 \text{ sisa}
 \end{array}$$

Sehingga hasilnya adalah 101, dan sisa pembagian adalah 110.

Pembagian bisa juga dilakukan dengan cara menjumlahkan secara berulang kali dengan bilangan pembagi dengan bilangan itu sendiri sampai jumlahnya sama dengan bilangan yang dibagi atau setelah sisa pembagian yang diperoleh lebih kecil dari bilangan pembagi.

## MODUL 4

### Operasi Aritmatik

**KELAS** : X TKJ  
**GURU MAPEL** : NurRahmat, S.Kom  
**Materi pokok** : Operasi Aritmatik  
**Alokasi waktu** : Jam Pelajaran

#### A. Tujuan Pembelajaran :

- 1 Siswa mengenal rangkaian half adder dan full adder
- 2 Siswa mampu membedakan rangkaian ripple carry adder








#### B. Materi Pembelajaran

#### Gerbang Logika

#### HALF ADDER & FULL ADDER

Gerbang logika atau sering juga disebut gerbang logika Boolean merupakan sebuah sistem pemrosesan dasar yang dapat memproses input-input yang berupa bilangan biner menjadi sebuah output yang berkondisi yang akhirnya digunakan untuk proses selanjutnya. Gerbang logika dapat mengkondisikan input-input yang masuk kemudian menjadikannya sebuah output yang sesuai dengan apa yang ditentukan olehnya. Terdapat tiga gerbang logika dasar, yaitu : gerbang AND, gerbang OR, gerbang NOT. Ketiga gerbang ini menghasilkan empat gerbang berikutnya, yaitu : gerbang NAND, gerbang NOR, gerbang XNOR.

**Tabel 4.1 : Kebenaran Gerbang Logika.**

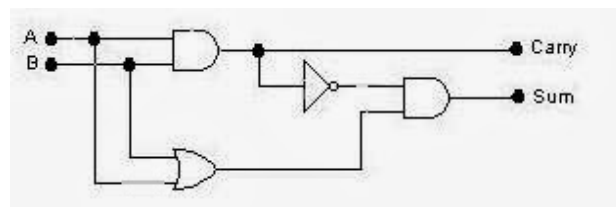
AND (7408)		NAND (7400)		XOR(7486)	
0 0=0	<div><div>A</div><div>B</div><div><math>A \cdot B = C</math></div></div>	0 0=1	<div><div>A</div><div>B</div><div><math>\overline{A \cdot B} = C</math></div></div>	0 0=0	<div><div>A</div><div>B</div><div><math>A \oplus B = C</math></div></div>
0 1=0		0 1=1		0 1=1	
1 0=0		1 0=1		1 0=1	
1 1=1		1 1=0		1 1=0	
OR (7432)		NOR (7402)		XAND	
0 0=0	<div><div>A</div><div>B</div><div><math>A+B = C</math></div></div>	0 0=1	<div><div>A</div><div>B</div><div><math>\overline{A+B} = C</math></div></div>	0 0=1	<div><div>A</div><div>B</div><div><math>\overline{A \oplus B} = C</math></div></div>
0 1=1		0 1=0		0 1=0	
1 0=1		1 0=0		1 0=0	
1 1=1		1 1=0		1 1=1	
		NOT (7404)			
		0 = 1			
		1 = 0			
		<div><div>A</div><div></div><div>C</div></div>			

Rangkaian aritmatika dasar termasuk ke dalam rangkaian kombinasional yaitu suatu rangkaian yang output-nya tidak tergantung pada kondisi output sebelumnya, hanya tergantung pada present state dari input.

### HALF ADDER

Merupakan rangkaian elektronik yang bekerja melakukan perhitungan penjumlahan dari dua buah bilangan binary, yang masing-masing terdiri dari satu bit. Rangkaian ini memiliki dua input dan dua buah output, salah satu outputnya dipakai sebagai tempat nilai pindahan dan yang lain sebagai hasil dari penjumlahan.

Rangkaian ini bisa dibangun dengan menggunakan IC 7400 dan IC 7408. Seperti yang terlihat pada gambar dibawah ini, rangkaian half adder merupakan gabungan beberapa gerbang NAND dan satu gerbang AND. Karakter utama sebuah gerbang NAND dalah bahwa ia membalikkan hasil dari sebuah gerbang AND yang karakternya hanya akan menghasilkan nilai satu ketika kedua inputnya bernilai satu, jadi gerbang NAND hanya akan menghasilkan nilai nol ketika semua inputnya bernilai satu.



**Gambar 4.1 : Skema Diagram HALF ADDER**

Ketika salah satu atau lebih input bernilai nol maka keluaran pada gerbang NAND pertama akan bernilai satu. Karenanya kemudian input di gerbang kedua dan ketiga akan bernilai satu dan mendapat input lain yang salah satunya bernilai nol sehingga PASTI gerbang NAND yang masukannya nol tadi menghasilkan nilai satu. Sedangkan gerbang lain akan bernilai nol karena mendapat input satu dan satu maka keluaran di gerbang NAND terakhir akan bernilai satu, karena salah satu inputnya bernilai nol.

Untuk menghitung carry digunakan sebuah gerbang AND yang karakter utamanya adalah bahwa ia hanya akan menghasilkan nilai satu ketika kedua masukannya bernilai satu. Jadi carry satu hanya akan dihasilkan dari penjumlahan dua digit bilangan biner sama-sama bernilai satu, yang dalam penjumlahan utamanya akan menghasilkan nilai nol.

**Tabel 4.2 : Kebenaran HALF ADDER**

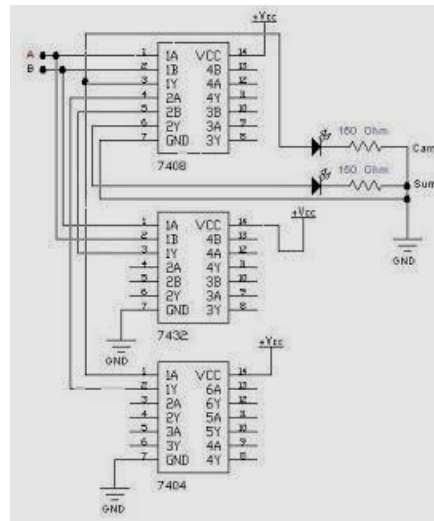
INPUT		OUTPUT	
A	B	CARRY	SUM
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Ket :

1 = Benar

0 = Salah

Jika setiap elemen yang dihubungkan salah satu ada yang Salah/(0) maka pernyataan pada percobaan Rangkaian Half Adder ini menunjukkan Salah/(0).



**Gambar 4.2 : Skema Pengkabelan HALF ADDER**

## KESIMPULAN :

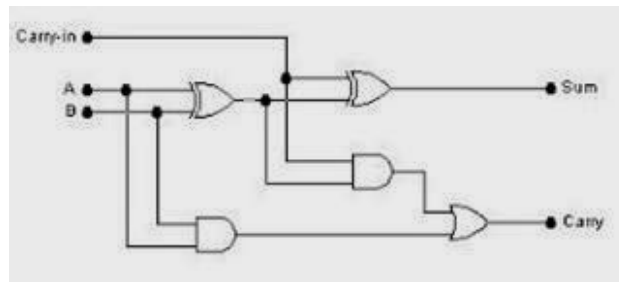
*Half Adder* adalah suatu rangkaian penjumlahan sistem bilangan biner yang paling sederhana. Rangkaian ini hanya dapat digunakan untuk operasi penjumlahan data bilangan biner sampai 1bit saja. Rangkaian Half Adder memiliki 2 terminal input untuk 2 variabel bilangan biner dan 2 terminal output, yaitu SUMMARY OUT (SUM) dan CARRY OUT (CARRY).

## FULL ADDER

Merupakan rangkaian elektronik yang bekerja melakukan perhitungan penjumlahan sepenuhnya dari dua buah bilangan binary, yang masing-masing terdiri dari satu bit. Rangkaian ini memiliki tiga input dan dua buah output, salah satu input merupakan nilai dari pindahan penjumlahan, kemudian sama seperti pada half adder salah satu outputnya dipakai sebagai tempat nilai pindahan dan yang lain sebagai hasil dari penjumlahan.

Rangkaian ini dibuat dengan gabungan dua buah half adder dan sebuah gerbang OR. Logika utama rangkaian gerbang full adder adalah bahwa ketika menjumlahkan dua bilangan biner maka ada sebuah carry yang juga mempengaruhi hasil dari penjumlahan tersebut, karenanya rangkaian ini bisa melakukan penjumlahan secara sepenuhnya.





**Gambar 4.3 : Skema Diagram FULL ADDER**

Ketika dua masukan menghasilkan nilai satu pada half adder atau paruh dari full adder pertama, hasilnya akan kembali dijumlahkan dengan carry yang ada. Jika carry bernilai satu maka ia akan menghasilkan keluaran akhir bernilai nol, namun menghasilkan carry out yang bernilai satu, dan jika carry in bernilai nol maka ia akan menghasilkan keluaran akhir satu dengan carry out bernilai nol.

Lain halnya ketika kedua masukan pada paruh full adder pertama menghasilkan nilai nol karena inputnya sama-sama satu, maka carry out untuk paruh pertama half adder adalah satu, penjumlahan paruh pertama yang menghasilkan nol akan kembali dijumlahkan dengan carry in yang ada, yang jika bernilai satu maka hasil penjumlahannya adalah satu dan memiliki carry out satu dari penjumlahan input pertama.

Untuk menghitung carry out pada full adder digunakan sebuah gerbang OR yang menghubungkan penghitung carry out dari half adder pertama dan kedua. Maksudnya bahwa entah paruh pertama atau kedua yang menghasilkan carry out maka akan dianggap sebagai carry out, dan dianggap satu meski kedua gerbang AND yang digunakan untuk menghitung carry out sama-sama bernilai satu.

**Tabel 4.3 : Kebenaran FULL ADDER**

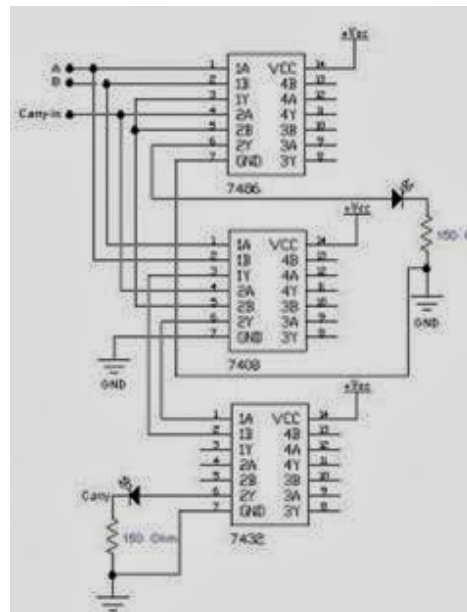
INPUT			OUTPUT	
A	B	CARRY IN	CARRY	SUM
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Ket :

1 = Benar

0 = Salah

Jika setiap elemen yang dihubungkan salah satu ada yang Benar/(1) maka pernyataan pada percobaan Rangkaian Full Adder ini menunjukkan pernyataan Benar/(1)



**Gambar 4.4 : Skema Pengkabelan FULL ADDER**

#### KESIMPULAN :

*Full Adder* dapat digunakan untuk menjumlahkan bilangan-bilangan biner yang lebih dari 1bit. Penjumlahan bilangan-bilangan biner sama halnya dengan penjumlahan bilangan decimal dimana hasil penjumlahan tersebut terbagi menjadi 2 bagian, yaitu SUMMARY (SUM) dan CARRY, apabila hasil penjumlahan pada suatu tingkat atau kolom melebihi nilai maksimumnya maka output CARRY akan berada pada keadaan logika 1.

#### Pengertian dan Cara Kerja Arithmetic Logical Unit (ALU)

##### Pengenalan ALU

Arithmetic Logical Unit (ALU), adalah salah satu bagian/komponen dalam sistem di dalam sistem komputer yang berfungsi melakukan operasi/perhitungan aritmatika dan logika (Contoh operasi aritmatika adalah operasi penjumlahan dan pengurangan, sedangkan contoh operasi logika adalah logika AND dan OR. ALU bekerja bersama-sama memori, di mana hasil dari perhitungan di dalam ALU di simpan ke dalam memori

Perhitungan dalam ALU menggunakan kode biner, yang merepresentasikan instruksi yang akan dieksekusi (opcode) dan data yang diolah (operand). ALU biasanya menggunakan sistem bilangan biner *two's complement*. ALU mendapat data dari register. Kemudian data tersebut diproses dan hasilnya akan disimpan dalam register tersendiri yaitu ALU output register, sebelum disimpan dalam memori.

Pada saat sekarang ini sebuah chip/IC dapat mempunyai beberapa ALU sekaligus yang memungkinkan untuk melakukan kalkulasi secara paralel. Salah satu chip ALU yang sederhana

(terdiri dari 1 buah ALU) adalah IC 74LS382/HC382ALU (TTL). IC ini terdiri dari 20 kaki dan beroperasi dengan 4×2 pin data input (pinA dan pinB) dengan 4 pin keluaran (pinF).

### Kegunaan / Fungsi ALU

Arithmetic Logical Unit (ALU), fungsi unit ini adalah untuk melakukan suatu proses data yang berbentuk angka dan logika, seperti data matematika dan statistika. ALU terdiri dari register-register untuk menyimpan informasi. Tugas utama dari ALU adalah melakukan perhitungan aritmatika (matematika) yang terjadi sesuai dengan instruksi program. Rangkaian pada ALU (Arithmetic and Logic Unit) yang digunakan untuk menjumlahkan bilangan dinamakan dengan Adder. Adder digunakan untuk memproses operasi aritmetika, Adder juga disebut rangkaian kombinasional aritmatika.

### Berikut contoh sederhana dari rangkaian ALU dan cara kerjanya

Rangkaian utama yang digunakan untuk melakukan perhitungan ALU adalah Adder. Adder adalah rangkaian ALU (Arithmetic and Logic Unit) yang digunakan untuk menjumlahkan bilangan dinamakan dengan Adder. Karena Adder digunakan untuk memproses operasi aritmetika, maka Adder juga sering disebut rangkaian kombinasional aritmetika.

Ada 3 jenis adder:

1. Rangkaian Adder dengan menjumlahkan dua bit disebut Half Adder.
2. Rangkaian Adder dengan menjumlahkan tiga bit disebut Full Adder.
3. Rangkaian Adder dengan menjumlahkan banyak bit disebut Paralel Adder.

#### 1. HALF ADDER

Rangkaian Half Adder merupakan dasar penjumlahan bilangan Biner yang terdiri dari satu bit, oleh karena itu dinamai Penjumlah Tak Lengkap.

a. jika  $A = 0$  dan  $B = 0$  dijumlahkan, hasilnya  $S$  (Sum) = 0.

b. jika  $A = 0$  dan  $B = 1$  dijumlahkan, hasilnya  $S$  (Sum) = 1.

c. jika  $A = 1$  dan  $B = 1$  dijumlahkan, hasilnya  $S$  (Sum) = 0

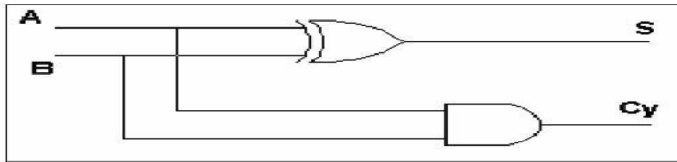
jika  $A = 1$  dan  $B = 1$  dijumlahkan, hasilnya  $S$  (Sum) = 0. dengan nilai pindahan  $cy$  (Carry Out) = 1

Dengan demikian, half adder memiliki 2 masukan (A dan B) dan dua keluaran (S dan Cy).

A	B	S	Cy
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Dari tabel diatas, terlihat bahwa nilai logika dari Sum sama dengan nilai logika dari gerbang XOR, sedangkan nilai logika Cy sama dengan nilai dari gerbang logika AND.

Dari table tersebut, dapat dibuat rangkaian half adder seperti pada gambar berikut:



## 2. FULL ADDER

Sebuah Full Adder menjumlahkan dua bilangan yang telah dikonversikan menjadi bilangan-bilangan biner. Masing-masing bit pada posisi yang sama saling dijumlahkan. Full Adder sebagai penjumlah pada bit-bit selain yang terendah. Full Adder menjumlahkan dua bit input ditambah dengan nilai Carry-Out dari penjumlahan bit sebelumnya. Output dari Full Adder adalah hasil penjumlahan (Sum) dan bit kelebihannya (carry-out).

Full adder mengolah penjumlahan untuk 3 bit bilangan atau lebih (bit tidak terbatas), olehkarena itu dinamakan rangkaian penjumlah lengkap. Perhatikan tabel kebenaran dari Full adder berikut :

A	B	C	S	Cy
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

Dari tabel diatas dapat dibuat persamaan boolean sebagai berikut :

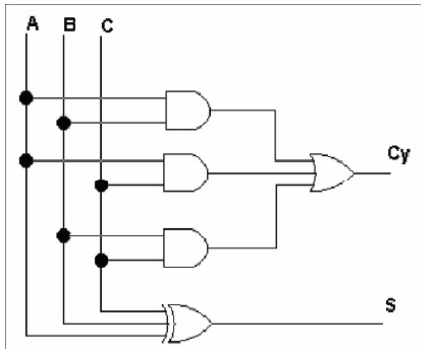
$$S = A B C + A B \bar{C} + A \bar{B} C + A \bar{B} \bar{C}$$

$$S = A \oplus B \oplus C$$

$$Cy = A B C + A B \bar{C} + A \bar{B} C + A \bar{B} \bar{C}$$

Dengan menggunakan peta karnaugh, Cy dapat disederhanakan menjadi :

$$Cy = AB + AC + BC$$



### 3. PARALEL ADDER

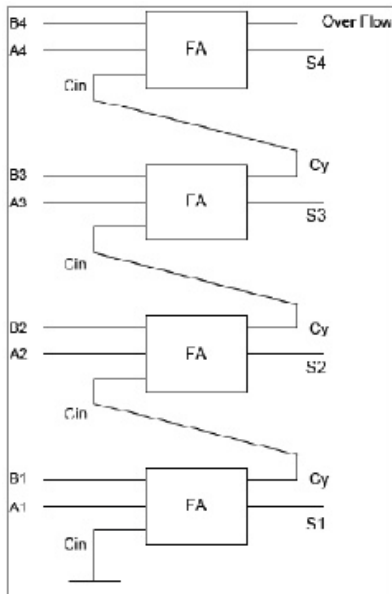
Rangkaian Parallel Adder adalah rangkaian penjumlah dari dua bilangan yang telah dikonversikan ke dalam bentuk biner. Anggap ada dua buah register A dan B, masing-masing register terdiri dari 4 bit biner : A3A2A1A0 dan B3B2B1B0.

Rangkaian Parallel Adder terdiri dari Sebuah Half Adder (HA) pada Least Significant Bit (LSB) dari masing-masing input dan beberapa Full Adder pada bit-bit berikutnya. Prinsip kerja dari Parallel Adder adalah sebagai berikut : penjumlahan dilakukan mulai dari LSB-nya. Jika hasil penjumlahan adalah bilangan desimal “2” atau lebih, maka bit kelebihannya disimpan pada Cout, sedangkan bit di bawahnya akan dikeluarkan pada  $\Sigma$ . Begitu seterusnya menuju ke Most Significant Bit (MSB)nya.

Tugas lain dari ALU adalah melakukan keputusan dari operasi sesuai dengan instruksi program yaitu operasi logika (logical operation). Operasi logika meliputi perbandingan dua buah elemen logika dengan menggunakan operator logika.

Parallel Adder adalah rangkaian Full Adder yang disusun secara parallel dan berfungsi untuk menjumlah bilangan biner berapapun bitnya, tergantung jumlah Full Adder yang diparalelkan. Gambar berikut menunjukkan Parallel Adder yang terdiri dari 4 buah Full Adder yang tersusun parallel sehingga membentuk sebuah penjumlah 4 bit.

Contoh:



### Tugas Arithmetic Logical Unit (ALU)

Bertugas membentuk fungsi – fungsi pengolahan data komputer.

ALU sering disebut mesin bahasa (machine language) karena bagian ini mengerjakan instruksi – instruksi bahasa mesin yang diberikan padanya.

ALU terdiri dari dua bagian, yaitu unit aritmetika dan unit logika boolean, yang masing – masing memiliki spesifikasi dan tugas tersendiri.

Fungsi-fungsi yang didefinisikan pada ALU adalah Add(penjumlahan), Addu (penjumlahan tidak bertanda), Sub(pengurangan), Subu (pengurangan tidak bertanda), and, or, xor, sll (shift left logical), srl (shift right logical), sra (shift right arithmetic), dan lain-lain.

Arithmetic Logical Unit (ALU) merupakan unit penalaran secara logic. ALU ini merupakan Sirkuit CPU berkecepatan tinggi yang bertugas menghitung dan membandingkan. Angka-angka dikirim dari memori ke ALU untuk dikalkulasi dan kemudian dikirim kembali ke memori. Jika CPU diasumsikan sebagai otaknya komputer, maka ada suatu alat lain di dalam CPU tersebut yang kenal dengan nama Arithmetic Logical Unit (ALU), ALU inilah yang berfikir untuk menjalankan perintah yang diberikan kepada CPU tersebut.

ALU sendiri merupakan suatu kesatuan alat yang terdiri dari berbagai komponen perangkat elektronika termasuk di dalamnya sekelompok transistor, yang dikenal dengan nama logic gate, dimana logic gate ini berfungsi untuk melaksanakan perintah dasar matematika dan operasi logika. Kumpulan susunan dari logic gate inilah yang dapat melakukan perintah perhitungan matematika yang lebih komplit seperti perintah “add” untuk menambahkan bilangan, atau “devide” atau pembagian dari suatu bilangan. Selain perintah matematika yang lebih komplit, kumpulan dari logic gate ini juga mampu untuk melaksanakan perintah yang berhubungan dengan logika, seperti hasil perbandingan dua buah bilangan.

Instruksi yang dapat dilaksanakan oleh ALU disebut dengan *instruction set*. Perintah yang ada pada masing-masing CPU belum tentu sama, terutama CPU yang dibuat oleh pembuat yang berbeda,