

BAB VI RANGKAIAN-RANGKAIAN ARITMETIK

Fungsi terpenting dari hampir semua computer dan kalkulator adalah melakukan operasi-operasi aritmetik. Operasi-operasi ini semuanya dilaksanakan di dalam unit aritmetik computer, di dalam unit ini gate-gate logika dan flip-flop dikombinasikan sedemikian rupa sehingga dapat melakukan operasi penjumlahan, pengurangan, perkalian dan pembagian bilangan-bilangan biner.

Pada bab ini akan dipelajari beberapa rangkaian aritmetik dasar yang digunakan untuk melaksanakan operasi-operasi aritmetik yang dibahas pada BAB V. dalam beberapa hal akan dibahas sampai proses desain yang sebenarnya, meskipun rangkaian-rangkaian tersebut mungkin tersedia dipasaran dalam bentuk rangkaian terintegrasi (IC), tujuan pembahasan ini adalah untuk memberikan lebih banyak latihan dalam menggunakan teknik-teknik yang dipelajari pada BAB III.

6.1. Unit Aritmetik

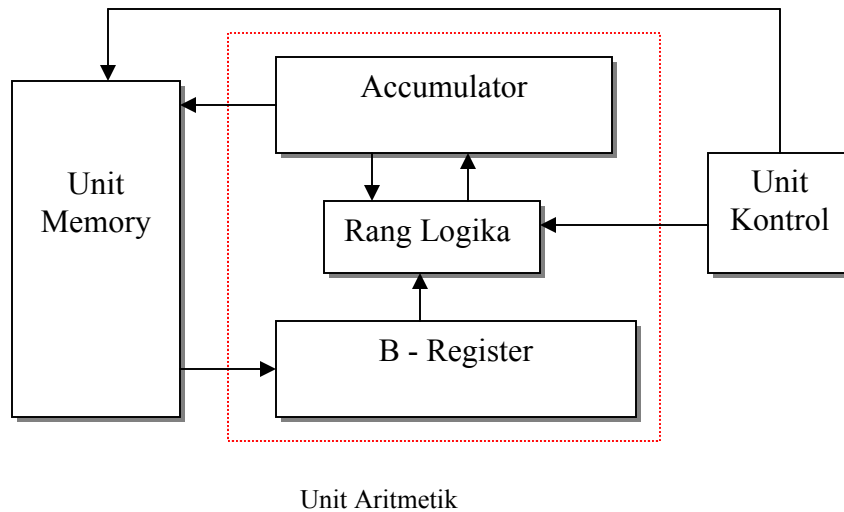
Semua operasi aritmetik terjadi di dalam unit aritmetik computer. Gambar 6.1. adalah diagram blok yang menunjukkan elemen-elemen utama yang terdapat di dalam unit aritmetik. Kegunaan utama dari unit aritmetik adalah menerima data biner yang disimpan di dalam memori dan melaksanakan operasi-operasi aritmetik atas data tersebut sesuai dengan instruksi-instruksi dari unit kontrol.

Unit aritmetik paling sedikit mengandung dua buah flip-flop register : B register dan accumulator register atau register pengumpul. Unit aritmetik juga mengandung logika kombinatorial yang melaksanakan operasi-operasi aritmetik atas bilangan-bilangan biner yang disimpan di dalam B register dan accumulator register.

Urutan operasi berjalan sebagai berikut :

1. Unit kontrol menerima suatu instruksi (dari unit memori) yang menghendaki agar suatu bilangan yang disimpan di dalam suatu tempat memori tertentu harus ditambahkan pada bilangan yang pada saat itu tersimpan di dalam accumulator register.

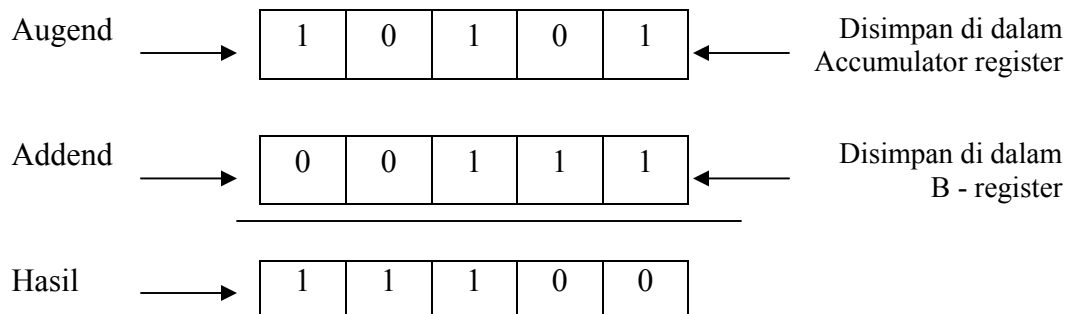
2. Bilangan yang ditambahkan tersebut ditransfer dari memori menuju B register
3. Bilangan di dalam B register dan bilangan di dalam accumulator register dijumlahkan menjadi satu di dalam rangkaian-rangkaian logika. Jumlah yang diperoleh kemudian dikirimkan ke accumulator untuk disimpan.
4. Bilangan baru di dalam accumulator tersebut dapat tetap tinggal di sana sehingga bilangan lain dapat ditambahkan kepadanya, atau apabila proses aritmetik tersebut telah selesai, hasilnya dapat di transfer ke memori untuk di simpan.



Gambar 6.1. Diagram bagian-bagian fungsional dari unit aritmetik

6.2. Penjumlahan Biner Paralel

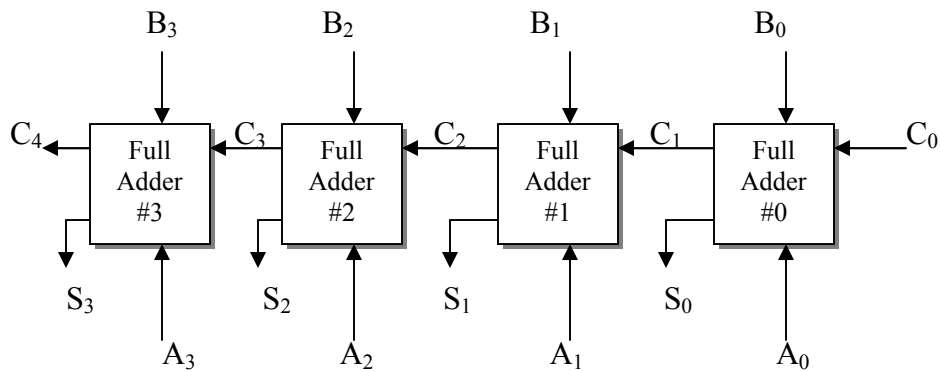
Komputer dan kalkulator melakukan operasi penjumlahan atas dua bilangan biner secara berturut-turut digit demi digit yang mana tiap-tiap bilangan biner dapat memiliki beberapa digit biner. Gambar 6.2. menunjukkan penjumlahan dari dua bilangan 4-bit.



Gambar 6.2 Contoh Proses penjumlahan Biner

Proses penjumlahan dimulai dengan menjumlahkan LSB dari augend (bilangan yang ditambah) dan addend (bilangan penambah). Jadi , $1 + 1 = 10$, yang berarti bahwa sum untuk posisi ini sama dengan 0 dengan carry 1. Carry ini harus ditambahkan pada posisi berikutnya, yaitu dijumlahkan menjadi satu dengan augend dan addend pada posisi tersebut. Jadi, pada posisi kedua, $1 + 1 + 1 = 10$, sum nya sama dengan 0 dan carry 1. Carry ini ditambahkan pada posisi berikutnya bersama-sama dengan bit-bit augend dan addend pada posisi itu, dan demikian seterusnya untuk posisi-posisi yang lainnya.

Dengan demikian dapat dirancang suatu rangkaian logika yang dapat melakukan proses ini hanya dengan menggunakan rangkaian-rangkaian identik untuk tiap-tiap posisi bit. Ini ditunjukkan pada gambar 6.3



Jumlah muncul pada output-output S_3 , S_2 , S_1 , dan S_0 ; serta carry C_4 .

Gambar 6.3. Diagram blok rangkaian adder paralel 4 bit yang menggunakan full adder

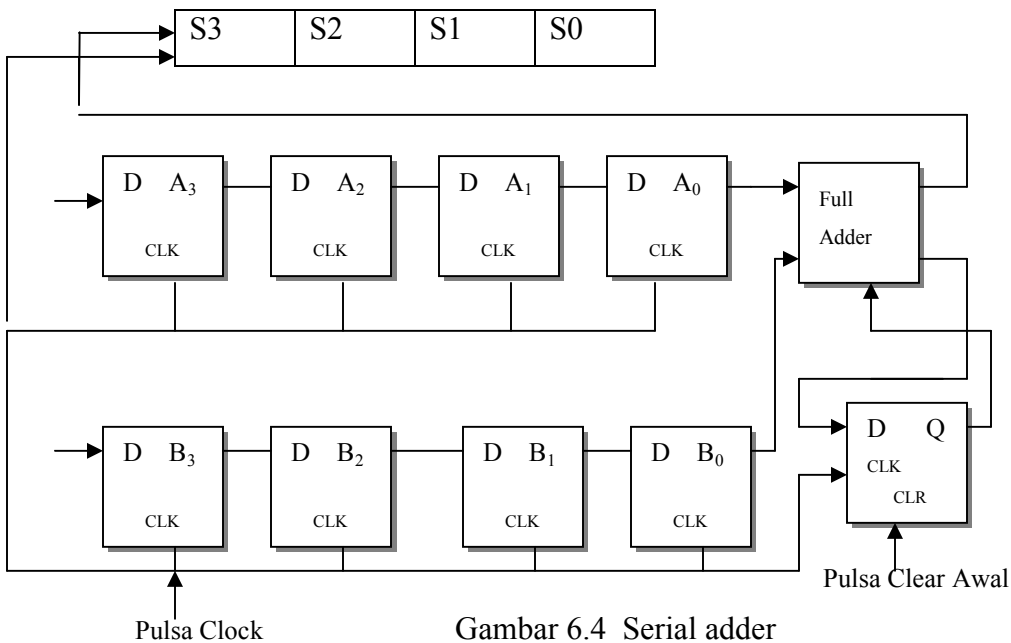
Pada diagram ini variabel-variabel A_3 , A_2 , A_1 , dan A_0 menyatakan bit-bit augend yang disimpan dalam accumulator register. Variabel-variabel B_3 , B_2 , B_1 , dan B_0 menyatakan bit-bit addend yang disimpan dalam B register. Variabel-variabel C_3 , C_2 , C_1 , dan C_0 menyatakan bit-bit Carry untuk posisi-posisi yang sesuai. Variabel-variabel S_3 , S_2 , S_1 , dan S_0 adalah bit-bit sum untuk tiap-tiap posisi.

Rangkaian Full Adder yang digunakan pada tiap-tiap posisi mempunyai tiga input: bit A, bit B, dan bit C, dan menghasilkan dua output : bit sum dan carry bit.

6.3 Penjumlah Biner Serial

Adder paralel yang telah dipelajari mengerjakan penjumlahan dua bilangan biner pada kecepatan yang relatif cepat karena semua posisi-posisi bit dioperasikan secara serentak. Kecepatannya dibatasi oleh waktu perambatan carry, kerugian utama dari proses penjumlahan paralel adalah memerlukan sejumlah besar rangkaian logika, yang semakin bertambah sebanding dengan jumlah bit-bit bilangan yang akan dijumlahkan. Pada penjumlahan seri proses penjumlahan dilakukan dengan cara yang kurang lebih sama dengan cara yang dilakukan pada saat mengerjakan penjumlahan di atas kertas, yaitu satu posisi demi satu posisi. Ini menghasilkan rangkaian yang jauh lebih sederhana dibandingkan dengan penjumlahan paralel tetapi menghasilkan kecepatan operasi yang jauh lebih lambat.

Gambar 6.4 menunjukkan diagram dari sebuah adder serial 4-bit. Register A dan B digunakan untuk menyimpan bilangan-bilangan yang akan dijumlahkan. Pada adder seri register-register tersebut merupakan shift register yang harga-harga binernya menggeser dari kiri ke kanan atas komando tiap pulsa clock. Output-output A_0 dan B_0 (LSB) diberikan ke dalam suatu rangkaian full adder tunggal bersama-sama dengan output Q dari carry FF. Carry FF adalah FF tersendiri yang digunakan untuk menyimpan carry out dari FA sehingga dapat dijumlahkan dengan posisi signifikan berikutnya dari bilangan-bilangan di dalam register-register.

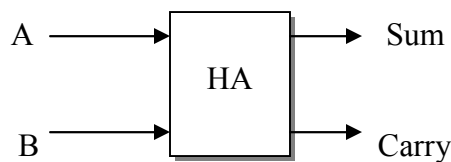


Gambar 6.4 Serial adder

6.4 Rangkaian Penjumlah Biner

6.4.1 Half Adder (HA)

Half adder adalah suatu rangkaian penjumlah system bilangan biner yang paling sederhana. Rangkaian ini hanya dapat digunakan untuk operasi penjumlahan data bilangan biner sampai 1 bit saja. Rangkaian half adder mempunyai 2 masukan dan 2 keluaran yaitu Summary out (Sum) dan Carry out (Carry). Secara blok diagram dapat digambar sebagai berikut :



Gambar 6.5 Blok Diagram Half Adder

Masukan : A = 1
 B = 0

Keluaran : ----- +
 0 1
Carry ↑
Sum ↑

Masukan : A = 1
 B = 1

Keluaran : ----- +
 1 0
Carry ↑
Sum ↑

Dimana A dan B merupakan data-data Input.

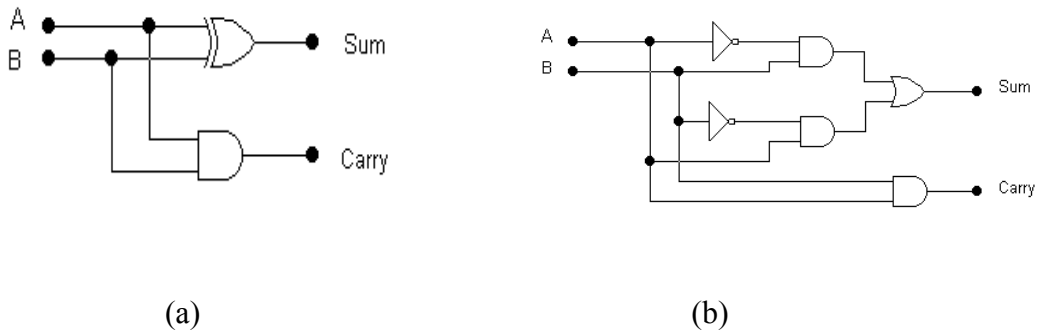
Operasi dari HA dapat ditunjukkan pada tabel kebenaran berikut :

Input		Output	
A	B	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Persamaan logikanya adalah :

$$\text{Sum} = (A' \cdot B) + (A \cdot B') \quad \text{serta} \quad \text{Carry} = A \cdot B$$

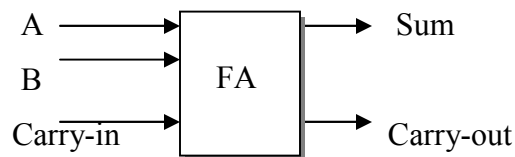
Half Adder dapat diimplementasikan dengan menggunakan EX-OR gate atau susunan gate yang membentuk fungsi EX-OR untuk sum dan AND gate untuk carry. Rangkaian HA dapat ditunjukkan pada gambar 6.6.a. untuk HA yang disusun dari EX-OR dan AND dan gambar 6.6.b. untuk HA yang disusun dari AND, OR dan NOT gate



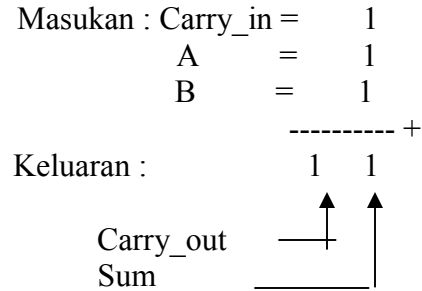
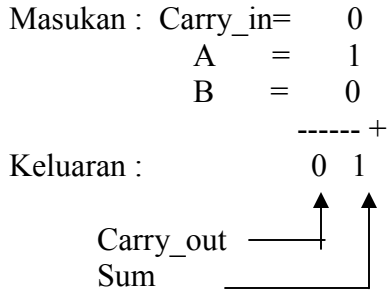
Gambar 6.6. Rangkaian Half Adder

6.4.2 Full Adder (FA)

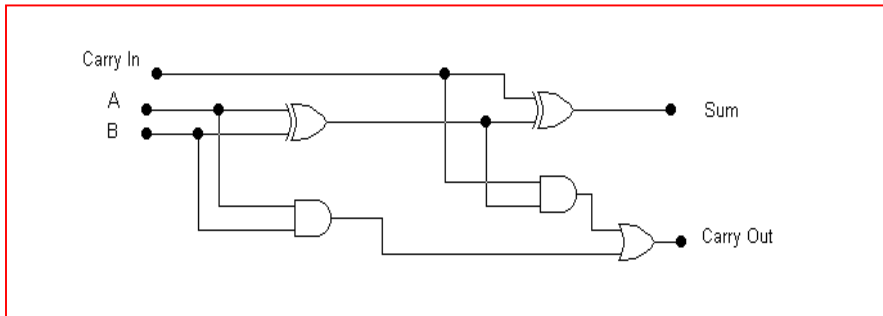
Rangkaian full adder dapat digunakan untuk menjumlahkan bilangan biner yang lebih dari 1 bit. Ciri pokok dari Full adder dibandingkan dengan half adder terletak pada jenis/jumlah masukan. Pada Full adder terdapat tambahan satu masukan, yaitu Carry_in.



Gambar 6.7 Blok Diagram FA



Rangkaian FA dapat disusun oleh dua buah HA yang di-OR-kan. (Gb.6.8)



Gambar 6.8 Full Adder yang disusun dari 2 buah HA

Tabel Kebenaran Full Adder

Masukan			Keluaran	
A	B	Carry In	Carry out	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

Dari tabel kebenaran dapat dituliskan ekspresi logika untuk Sum dan carry :

$$\text{Sum} = A'B'C_{in} + A'B'C_{in}' + A B'C_{in}' + A B C_{in}$$

$$\text{Carry} = A'B C_{in} + A B' C_{in} + A B C_{in}' + A B C_{in}$$

Persamaan di atas disederhanakan dengan menggunakan K-Map :

		B A			
		00	01	11	10
C	0	0	1	0	1
	1	1	0	1	0

Karnaugh map untuk Sum

		BA			
		00	01	11	10
C	0	0	0	1	0
	1	0	1	1	1

Karnaugh map untuk Carry-out

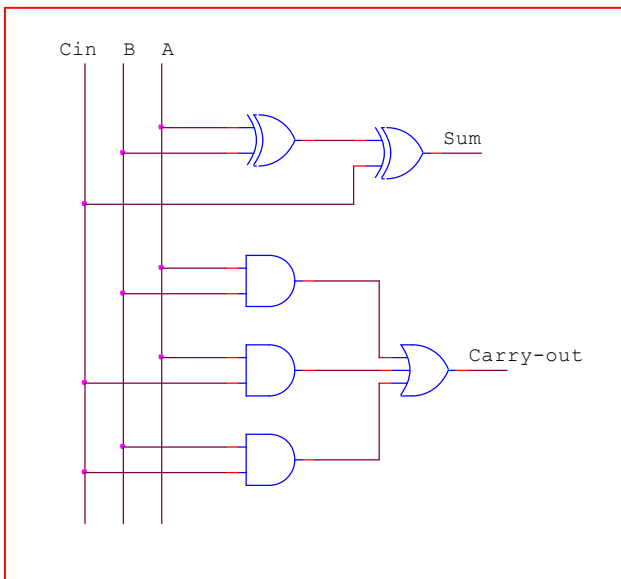
Persamaan logikanya adalah :

$$\begin{aligned} \text{Sum} &= A'B'C_{in} + A'B C_{in}' + A B'C_{in}' + A B C_{in} \\ &= (A'B' + AB) C_{in} + (A'B + AB') C_{in}' \end{aligned}$$

$$\text{Sum} = (A \oplus B) \oplus C_{in}$$

$$C_{out} = (A.B) + (A.C_{in}) + (B.C_{in})$$

Berdasarkan tabel kebenaran dapat disusun juga FA seperti ditunjukkan pada gambar 6.9



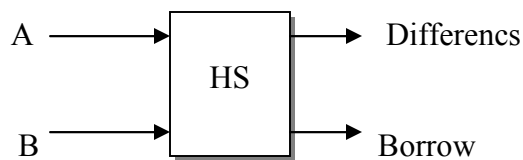
Gambar. 6.9 Rangkaian FA

6.5 Rangkaian Pengurang Biner

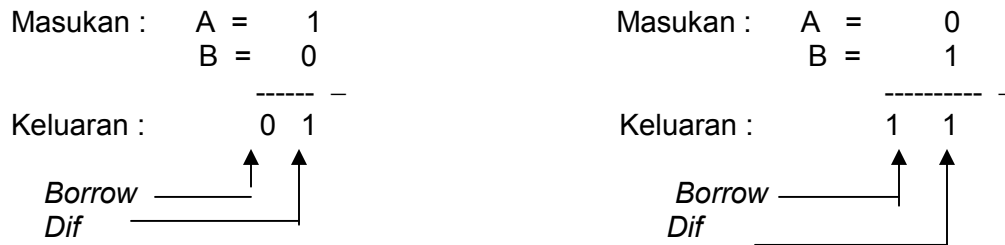
6.5.1 Half Subtractor (HS)

Half subtractor adalah suatu rangkaian yang dapat digunakan untuk melakukan operasi pengurangan data-data bilangan biner hingga 1 bit saja. Half subtractor mempunyai karakteristik : 2 masukan yaitu input A dan B serta 2 keluaran yaitu Difference (Dif) dan Borrow (Br). Pada contoh berikut, input B sebagai bilangan pengurangnya dan input A sebagai bilangan yang dikurang.

Secara blok diagram dapat digambar sebagai berikut :



Gambar 6.10 Blok Diagram Half Subtractor



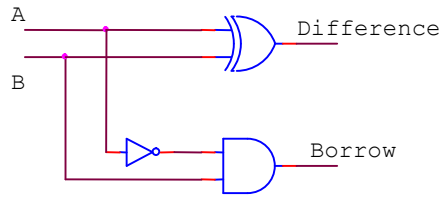
Input		Output	
A	B	Borrow	Difference
0	0	0	0
0	1	1	0
1	0	0	1
1	1	0	0

Tabel Kebenaran Half Subtractor

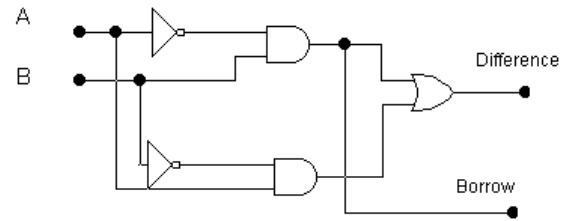
Persamaan Logikanya adalah :

Difference = (A' . B) + (A . B') serta **Borrow = A' . B**

Berdasarkan persamaan logika tersebut di atas dapat digambarkan rangkaian HS :



(a)

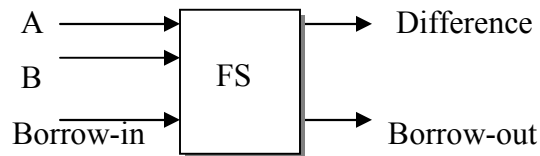


(b)

Gambar 6.11. a Rangkaian HS tersusun dari EX-OR, AND dan NOT
b Rangkaian HS dari AND, OR dan NOT

6.5.2 Full Subtractor (FS)

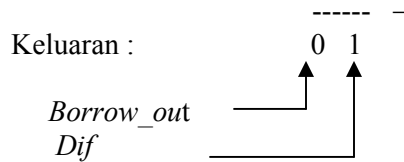
Rangkaian full subtractor digunakan untuk melakukan operasi pengurangan bilangan biner yang lebih dari 1 bit. Dengan 3 terminal input yang dimilikinya yaitu A, B, serta terminal Borrow input dan 2 terminal output yaitu Dif dan Borrow out. Secara blok diagram dapat digambarkan sebagai berikut :



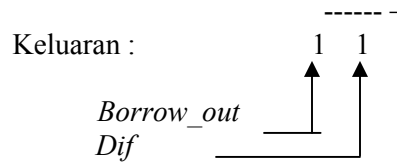
Gambar 6.12 Blok Diagram FS

Masukan :

A = 1
B = 0
Borrow_in = 0



A = 1
B = 1
Borrow_in = 1



Tabel Kebenaran Full Subtractor

A	Masukan		Keluaran	
	B	Borrow In	Borrow- out	Difference
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Dari tabel kebenaran dapat dituliskan ekspresi logika untuk Dif dan Br-out :

$$\text{Dif} = A'B'Br\text{-in} + A'B Br\text{-in}' + A B'Br\text{-in}' + A B Br\text{-in}$$

$$\text{Borrow} = A'B' Br\text{-in} + A' B Br\text{-in}' + A' B Br\text{-in} + A B Br\text{-in}$$

Persamaan di atas disederhanakan dengan menggunakan K-Map :

		B A			
		00	01	11	10
Br-in	0	0	1	0	1
	1	1	0	1	0

Karnaugh map untuk Dif

		B A			
		00	01	11	10
Br-in	0	0	1	0	0
	1	1	1	1	0

Karnaugh map untuk Borrow-out

Persamaan logikanya adalah :

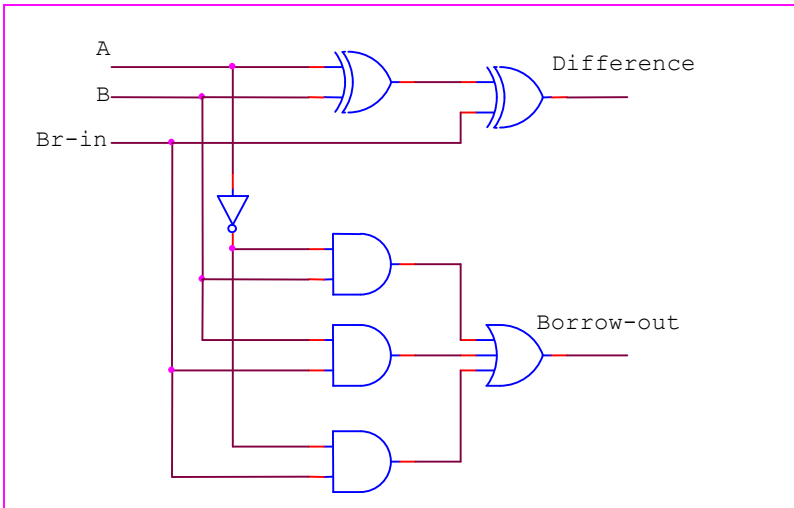
$$\text{Dif} = A'B'Br\text{-in} + A'B Br\text{-in}' + A B'Br\text{-in}' + A B Br\text{-in}$$

$$= (A'B' + AB) Br\text{-in} + (A'B + AB') Br\text{-in}'$$

$$\text{Dif} = (A \oplus B) \oplus Br\text{-in}$$

$$\text{Borrow_out} = A' \cdot B + B Br\text{-in} + A' Br\text{-in}$$

Rangkaian Full Subtractor dapat digambarkan sebagai berikut :



Gambar 6.13 Rangkaian Full Subtractor

6.6 Rangkaian Penjumlah/Pengurang

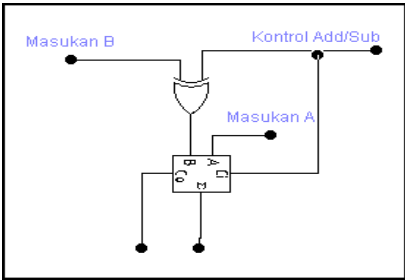
Penjumlahan dan pengurangan 4 bit sesungguhnya didasari oleh penjumlahan dan pengurangan 1 bit . Untuk penjumlahan 4 bit diperlukan 4 buah full adder yang inputnya dipasang secara parallel dan outputnya diambil dari masing-masing “sum” . C-out dimasukkan ke C_in dari full adder berikutnya.

Seperti diketahui bahwa persamaan **Sum** pada penjumlahan 1 bit (FA) sesungguhnya sama dengan **Difference** pada pengurangan 1 bit (FS),

$$\text{Sum (Difference)} = A'.B'.C + A'.B.C' + A.B'.C' + A.B.C$$

Akan tetapi persamaan untuk *carry_out* dan *Borrow_out* adalah berlainan. Oleh karena itu untuk membuat rangkaian dapat berfungsi sebagai penjumlah dan pengurang perlu ditambah kontrol *Add/Sub*. Hal ini sebagai pemilih kapan rangkaian tersebut memilih *Carry_out* atau *Borrow_out*.

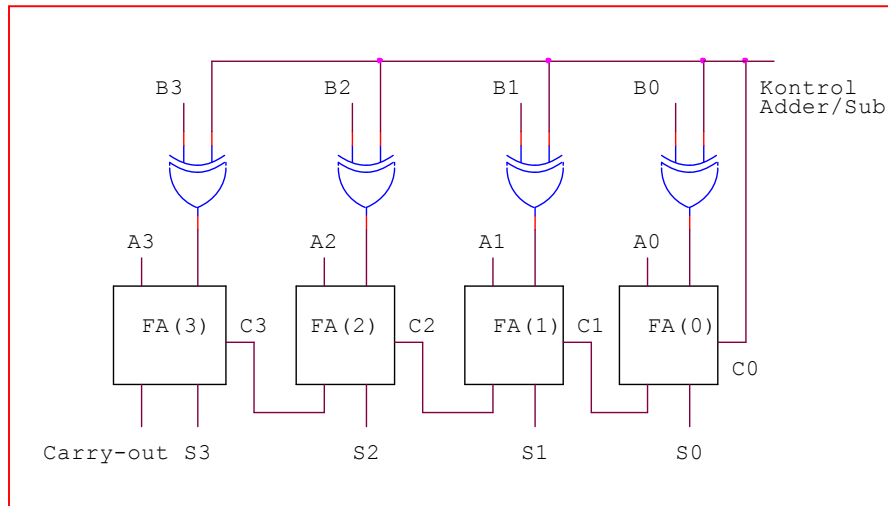
Kontrol Add/Sub ditunjukkan sebagai berikut :



Gambar 6.14. Penjumlahan dan Pengurangan dengan Kontrol Add/Sub

Sehingga dari kontrol add/sub tersebut rangkaian dapat berfungsi sebagai penjumlahan, jika kontrol add/sub berlogika rendah, dan berfungsi sebagai pengurangan, bilamana kontrol add/sub berlogika tinggi.

Gambar 6.15 memperlihatkan rangkaian adder/subtractor 4 bit



Gambar 6.15 Rangkaian adder/subtractor 4 bit

Soal :

1. Apa fungsi dari sebuah Full Adder ? tuliskan tabel kebenarannya.
2. Ubahlah rangkaian FA dari gambar 6.9 seluruhnya dari NAND gate.
3. Apakah fungsi dari sebuah Half Adder ? apakah bedanya dengan Full Adder ?
4. Apa fungsi dari sebuah Full Subtractor ? tuliskan tabel kebenarannya
5. Ubahlah rangkaian FS dari gambar 6.13 seluruhnya dari NAND gate
6. Apakah fungsi dari sebuah Half Subtractor ? apakah bedanya dengan Full Subtractor ?