

# RANGKAIAN ARITMETIKA 2

## *Pokok Bahasan :*

1. Sistem Coding
2. Fungsi-fungsi Aritmetika Biner : penjumlahan, pengurangan, perkalian, pembagian
3. Implementasi fungsi Aritmetika pada sistem Bilangan yang lain

## *Tujuan Instruksional Khusus :*

1. Mahasiswa dapat membedakan sistem kode : BCD, ASCII, Grey dan Hamming
2. Mahasiswa dapat melakukan fungsi : penjumlahan, pengurangan, pembagian dan perkalian dengan sistem biner
3. Mahasiswa dapat melakukan fungsi : penjumlahan, pengurangan, perkalian dan pembagian dengan sistem yang lain

# SISTIM CODING

## 1. Kode BCD (Binary Coded Decimal)

Merepresentasikan masing-masing 10 digit desimal menjadi kode 4-digit biner.

Kode ini digunakan untuk meng-outputkan hasil digital ke peralatan yang men-displaykan bilangan numerik (0-9), seperti : jam digital, voltmeter digital

Ada 5 jenis kode BCD :

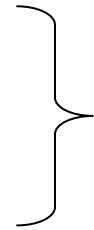
1. Kode 8421

2. Kode 5421

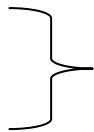
3. Kode 2421

4. Kode Excess-3

5. Kode 2 of 5



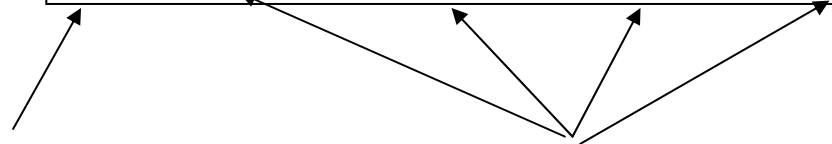
Kode dengan faktor pembobot



Bukan kode pembobot

Kode pembobot direpresentasikan sebagai :

$$d_{10} = 8xa_3 + 4xa_2 + 2xa_1 + 1xa_0$$



*Nilai desimal*

*Nilai bobot (tergantung jenis kode pembobot)*

*Contoh :*

$$1) \quad 7_{10} = \dots\text{BCD (8421)} ?$$

$$7_{10} = 8 \times 0 + 4 \times 1 + 2 \times 1 + 1 \times 1 \rightarrow 7_{10} = 0111_{\text{BCD(8421)}}$$

$$2) \quad 18_{10} = \dots\text{BCD (5421)} ?$$

$$\begin{aligned} 18_{10} &= 5 \times 0 + 4 \times 0 + 2 \times 0 + 1 \times 1 & 5 \times 1 + 4 \times 0 + 2 \times 1 + 1 \times 1 \\ &= 0001 \quad 1011_{\text{BCD(5421)}} \end{aligned}$$

$$3) \quad 48_{10} = \dots\text{BCD (2421)} ?$$

$$\begin{aligned} 48_{10} &= 2 \times 0 + 4 \times 1 + 2 \times 0 + 1 \times 0 & 2 \times 1 + 4 \times 1 + 2 \times 1 + 1 \times 0 \\ &= 0100 \quad 1110_{\text{BCD(2421)}} \end{aligned}$$

Dari ke-tiga jenis kode BCD dengan bobot, yang paling banyak digunakan adalah kode 8421

## *Kode Excess-3*

Kode ini memiliki kelebihan nilai 3 dari digit asalnya.

Contoh :

$0_{10}$  disimpan sebagai  $(0+3) = 0011_{\text{Excess-3}}$

Nilai tertinggi untuk BCD Excess-3 adalah  $(9+3) = 1100_{\text{Excess-3}}$

## *Kode 2 of 5*

Kode ini memiliki 2 nilai bit "1" dari 5 bit yang tersedia. Penempatan bit "1" dimulai dari MSB, sedang bit "1" untuk digit berikutnya mengikuti posisi di sebelahnya.

Contoh :

$2_{10}$  disimpan sebagai  $10010_{2 \text{ of } 5}$

# Ringkasan Kode BCD

Digit desimal	Kode 8421	Kode 5421	Kode 2421	Kode Excess-3	Kode of 5
0	0000	0000	0000	0011	11000
1	0001	0001	0001	0100	10100
2	0010	0010	0010	0101	10010
3	0011	0011	0011	0110	10001
4	0100	0100	0100	0111	01100
5	0101	1000	1011	1000	01010
6	0110	1001	1100	1001	01001
7	0111	1010	1101	1010	00110
8	1000	1011	1110	1011	00101
9	1001	1100	1111	1100	00011
tidak digunakan	1010	0101	0101	0000	sembarang pola yg lain
	1011	0110	0110	0001	
	1100	0111	0111	0010	
	1101	1101	1000	1101	
	1110	1110	1001	1110	
	1111	1111	1010	1111	

## 2. Kode ASCII (American Standard Code for Information Interchange)

Merepresentasikan nilai alphanumeric (huruf, bilangan dan simbol) menjadi nilai-nilai biner

Nilai-nilai ini akan dibaca dan diproses oleh peralatan digital (misal : komputer, microprocessor) dalam bentuk biner

ASCII Code terdiri dari 7 bit biner  $\rightarrow 2^7 = 128$  kombinasi kode

7 bit  $\rightarrow$  3 bit MSB dan 4 bit LSB

Contoh :

$\underbrace{100}_{\text{Grup 3 bit (MSB)}} \underbrace{0111}_{\text{Grup 4 bit (LSB)}} = G$

# Tabel ASCII

LSB \ MSB	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M	]	m	}
1110	SOH	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL



**Definisi kelas kontrol :**

ACK	Acknowledge	GS	Group Separator
BEL	Bell	HT	Horizontal Tag
BS	Backspace	LF	Line Feed
CAN	Cancel	NAK	Negative Acknowledge
CR	Carriage Return	NUL	Null
DC1-DC4	Direct Control	RS	Record Separator
DEL	Delete idle	SI	Shift In
DLE	Data Link Escape	SO	Shift Out
EM	End of Medium	SOH	Start of Heading
ENQ	Enquiry	STX	Start of Text
EOT	End of Transmission	SUB	Substitute
ESC	Escape	SYN	Synchronous Idle
ETB	End of Transmission Block	US	Unit Separator
ETX	End Text	VT	Vertical Tab
FF	Form Feed		
FS	Form Separator		

Contoh :

Dengan menggunakan Tabel ASCII, tentukan kode ASCII untuk 65-M

Jawab :

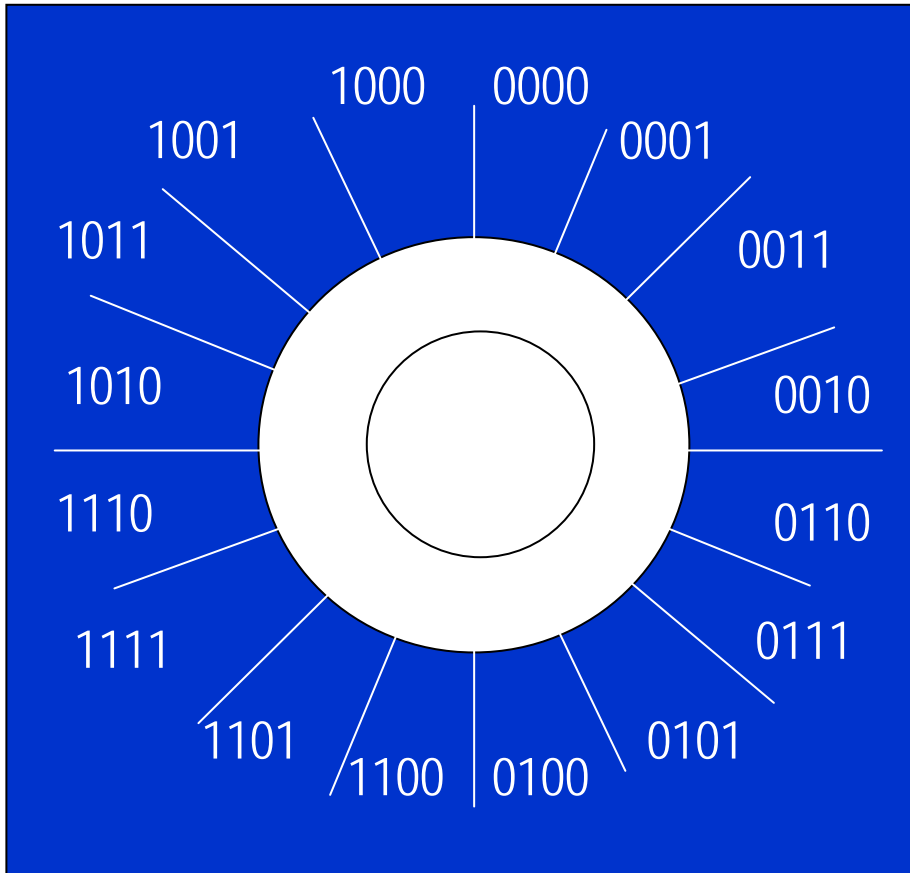
6	=	011 0110
5	=	011 0101
-	=	010 1101
M	=	100 1101

### 3. Gray Code

Digunakan dalam pengkodean posisi sudut dari peralatan yang bergerak secara berputar, seperti motor stepper, mesin bubut otomatis, gerinda

Kode ini terdiri dari 4 bit biner, dengan  $2^4 \rightarrow 16$  kombinasi untuk total putaran  $360^\circ$ .

Masing-masing kode digunakan untuk perbedaan sudut  $22,5^\circ$   
(=  $360^\circ/16$ )



Roda Gray Code

Bilangan	Gray Code	Biner 4-bit
0	0000	0000
1	0001	0001
2	0011	0010
3	0010	0011
4	0110	0100
5	0111	0101
6	0101	0110
7	0100	0111
8	1100	1000
9	1101	1001
10	1111	1010
11	1110	1011
12	1010	1100
13	1011	1101
14	1001	1110
15	1000	1111

Tabel Gray Code dan Biner

## 4. Hamming Code

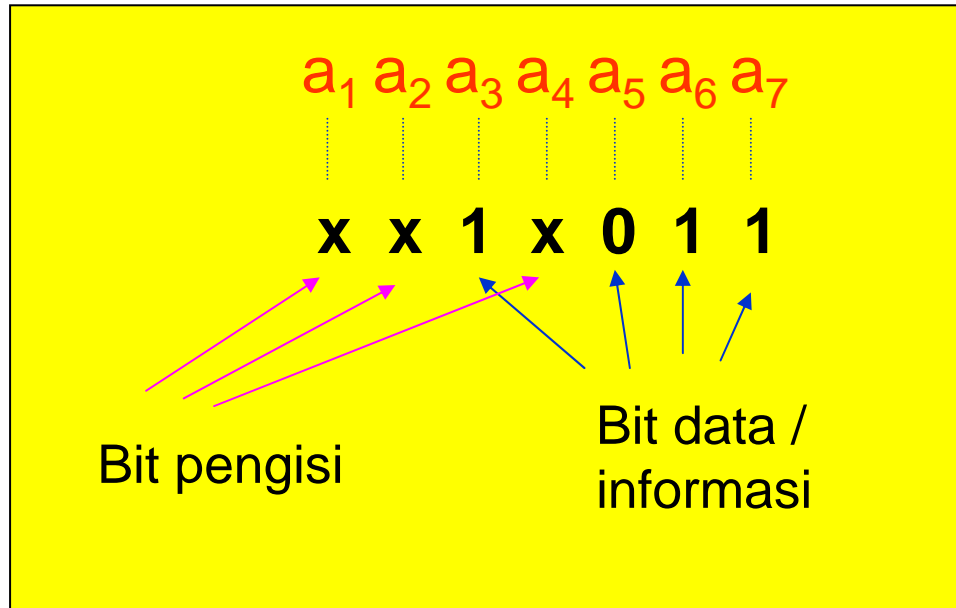
Kode ini dikenalkan oleh Richard Hamming (1950) sebagai kode tunggal pengoreksi kesalahan (*single error-correcting code*).

Bit pengecek ditambahkan ke dalam bit-bit informasi, jika suatu saat ada perubahan bit-bit data ketika proses transmisi, maka bit-bit informasi asli masih bisa diselamatkan.

Kode ini dikenal pula sebagai *parity code*

Bit pengecek tambahan diberikan pada bit-bit informasi sebelum ditransmisikan, sedangkan pada sisi penerima dilakukan pengecekan dengan algoritma yang sama dengan pembangkitan bit pengecek tambahan

# Cara pengisian bit tambahan pada bit-bit informasi



Untuk bit data 4-bit, bit-bit data terletak pada posisi 3, 5, 6 dan 7  
 Bit pengisi terletak pada posisi 1, 2, 4 ( $2^K$ )  $\rightarrow$   $K = \text{jumlah bit data} - 1$

Jumlah bit informasi = $2^n - n - 1$ ( $n \rightarrow$ jumlah bit cek)	$\left\{ \begin{array}{l} \Sigma \text{ Bit pengisi/cek} \\ \Sigma \text{ bit informasi} \end{array} \right.$	2	1
		3	4
		4	11
		5	26

Nilai bit pengisi/cek : (untuk informasi 4-bit)

$$a_1 = a_3 \oplus a_5 \oplus a_7$$

$$a_2 = a_3 \oplus a_6 \oplus a_7$$

$$a_4 = a_5 \oplus a_6 \oplus a_7$$

Untuk informasi n-bit, nilai bit pengisi / cek adalah :

$$a_1 = 3, 5, 7, 9, 11, 13, 15, \dots$$

$$a_2 = 3, 6, 7, 10, 11, 14, 15, \dots$$

$$a_4 = 5, 6, 7, 12, 13, 14, 15, 20, 21, 22, 23, \dots$$

$$a_8 = 9-15, 24-31, 40-47, \dots$$

$$a_{16} = 17-31, 48-63, 80-95, \dots$$

$$a_{32} = 33-63, 96-127, 160-191, \dots$$

dst.

Bit-bit masing-masing posisi yang disertakan di Ex-OR kan

# Tabel Hamming untuk informasi 4-bit

Data/bit	a <sub>1</sub>	a <sub>2</sub>	a <sub>3</sub>	a <sub>4</sub>	a <sub>5</sub>	a <sub>6</sub>	a <sub>7</sub>
0000	0	0	0	0	0	0	0
0001	1	1	0	1	0	0	1
0010	0	1	0	1	0	1	0
0011	1	0	0	0	0	1	1
0100	1	0	0	1	1	0	0
0101	0	1	0	0	1	0	1
0110	1	1	0	0	1	1	0
0111	0	0	0	1	1	1	1
1000	1	1	1	0	0	0	0
1001	0	0	1	1	0	0	1
1010	1	0	1	1	0	1	0
1011	0	1	1	0	0	1	1
1100	0	1	1	1	1	0	0
1101	1	0	1	0	1	0	1
1110	0	0	1	0	1	1	0
1111	1	1	1	1	1	1	1

*Contoh :*

Bagaimana bentuk data yang ditransmisikan dengan kode Hamming, jika diketahui bit data = 1010 ?

Jawab :

$$a1 = a3 \oplus a5 \oplus a7 \rightarrow a1 = 1 \oplus 0 \oplus 0 = 1$$

$$a2 = a3 \oplus a6 \oplus a7 \rightarrow a2 = 1 \oplus 1 \oplus 0 = 0$$

$$a4 = a5 \oplus a6 \oplus a7 \rightarrow a3 = 0 \oplus 1 \oplus 0 = 1$$

Sehingga bentuk data yang ditransmisikan menjadi : 1011010



Cara pengecekan di sisi terima : (untuk informasi 4-bit)

$$e_1 = a_1 \oplus a_3 \oplus a_5 \oplus a_7$$

$$e_2 = a_2 \oplus a_3 \oplus a_6 \oplus a_7$$

$$e_3 = a_4 \oplus a_5 \oplus a_6 \oplus a_7$$

Jika nilai  $\mathbf{e} = \mathbf{0}$ , maka seluruh data yang diterima adalah benar

## Untuk informasi n-bit, cara pengecekan adalah :

1. Tanda semua posisi bit yang merupakan pangkat dua sebagai bit pengecek (posisi 1, 2, 4, 8, 16, 32, 64, ...).
2. Posisi yang lain digunakan sebagai bit data yang akan dikodekan (posisi 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17, ...)
3. Masing-masing bit pengecek menghitung bit setiap posisi dengan cara mengecek dan melewati, sebagai berikut :  
*Posisi 1* : cek 1 bit, lewat 1 bit, cek 1 bit, lewat 1 bit dsb  
(1,3,5,7,9,11, 13, 15...)  
*Posisi 2* : cek 2 bit, lewat 2 bit, cek 2 bit, lewat 2 bit dsb  
(2,3,6,7,10,11, 14, 15,...)

..... *Next page*

.....*cont'd*

*Posisi 4* : cek 4 bit, lewat 4 bit, cek 4 bit, lewat 4 bit dsb  
(4,5,6,7,12,13,14,15,20,21,22,23, ...)

*Posisi 8*: cek 8 bit, lewat 8 bit, cek 8 bit, lewat 8 bit dsb  
(8-15,24-31,40-47,...)

*Posisi 32*: cek 32 bit, lewat 32 bit, cek 32 bit, lewat 32  
bit, dsb. (32-63,96-127,160-191,...)

Beri nilai bit penge-cek = 1 jika total bit "1" di posisi  
yang di cek adalah ganjil (*Odd*)  
dan beri nilai 0 jika total bit "1" adalah genap (*Even*)

*Contoh :*

Sebuah urutan data diterima : 0010011

Dengan :  $e_1 = 0$     $e_2 = 1$     $e_4 = 0$

Tentukan bit di posisi mana yang salah ? Berapa nilai data asli (sebelum ditambah bit pengecek) ?

Jawab :

$$e_1 = a_1 + a_3 + a_5 + a_7 = 0 + 1 + 0 + 1 = 0 \rightarrow \text{benar}$$

$$e_2 = a_2 + a_3 + a_6 + a_7 = 0 + 1 + 1 + 1 = 1 \rightarrow \text{salah}$$

$$e_3 = a_4 + a_5 + a_6 + a_7 = 0 + 0 + 1 + 1 = 0 \rightarrow \text{benar}$$

$$a_1 = a_3 + a_5 + a_7 = 1 + 0 + 1 = 0 \rightarrow \text{sama dengan yang dikirim}$$

$$a_2 = a_3 + a_6 + a_7 = 1 + 1 + 1 = 1 \rightarrow \text{tidak sama dengan yang dikirim}$$

$$a_3 = a_5 + a_6 + a_7 = 0 + 1 + 1 = 0 \rightarrow \text{sama dengan yang dikirim}$$

Berarti bit di posisi 2 yang salah, seharusnya yang diterima adalah : 0110011

Nilai data asli =  $a_3a_5a_6a_7 = 1011$

# FUNGSI-FUNGSI ARITMETIKA BINER

## 1. PENJUMLAHAN

- Penjumlahan dasar (pada kolom LSB)

$$A_0 + B_0 = \Sigma_0 + C_{out}$$

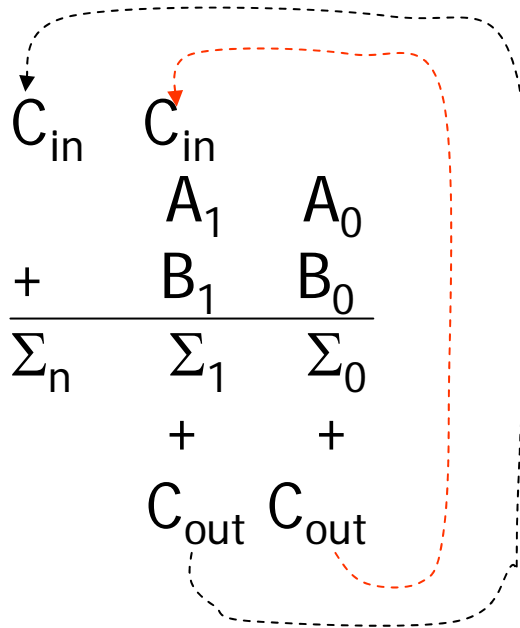
$0 + 0 = 0$	carry 0
$0 + 1 = 1$	carry 0
$1 + 0 = 1$	carry 0
$1 + 1 = 0$	carry 1

Tabel Kebenaran untuk  
Penjumlahan 2 bit biner (LSB)

$A_0$	$B_0$	$\Sigma_0$	$C_{out}$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- Penjumlahan lanjut (selain kolom LSB)

$$A_i + B_i + C_{in} = \Sigma_i + C_{out} \quad i = 2, 3, 4, \dots$$



Tabel Kebenaran untuk Penjumlahan 2 bit biner (lanjut)

$A_1$	$B_1$	$C_{in}$	$\Sigma_1$	$C_{out}$
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

*Contoh :*

$$\begin{array}{r} 1. \quad 5 \\ + 4 \\ \hline 9 \end{array} \qquad \begin{array}{r} 0101 \\ + 0100 \\ \hline 1001 = 9_{10} \end{array}$$

$$\begin{array}{r} 2. \quad 18 \\ + 2 \\ \hline 20 \end{array} \qquad \begin{array}{r} 10010 \\ + 00010 \\ \hline 10100 = 20_{10} \end{array}$$

$$\begin{array}{r} 3. \quad 147 \\ + 75 \\ \hline 222 \end{array} \qquad \begin{array}{r} 10010011 \\ + 01001011 \\ \hline 11011110 = 222_{10} \end{array}$$

## 2. PENGURANGAN

- Pengurangan dasar (pada kolom LSB)

$$A_0 - B_0 = R_0 + B_{out}$$

0 - 0 = 0	borrow 0
0 - 1 = 1	borrow 1
1 - 0 = 1	borrow 0
1 - 1 = 0	borrow 0

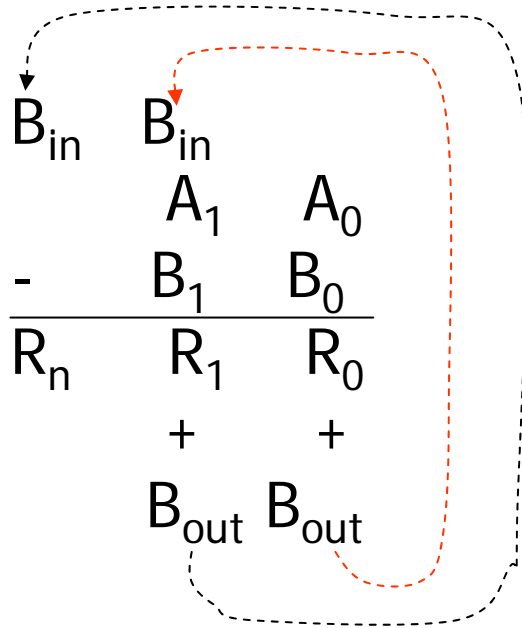
Tabel Kebenaran untuk  
Pengurangan 2 bit biner (LSB)

$A_0$	$B_0$	$R_0$	$B_{out}$
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0



- Pengurangan lanjut (selain kolom LSB)

$$A_i - B_i - B_{in} = R_i + B_{out} \quad i = 2, 3, 4, \dots$$



Tabel Kebenaran untuk  
Pengurangann 2 bit biner (lanjut)

A <sub>1</sub>	B <sub>1</sub>	B <sub>in</sub>	R <sub>1</sub>	B <sub>out</sub>
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

*Contoh :*

$$\begin{array}{r} 1. \quad 9 \\ - 4 \\ \hline 5 \end{array}$$

$$\begin{array}{r} 1001 \\ + 0100 \\ \hline 0101 = 5_{10} \end{array}$$

$$\begin{array}{r} 2. \quad 18 \\ - 12 \\ \hline 6 \end{array}$$

$$\begin{array}{r} 10010 \\ - 01100 \\ \hline 00110 = 6_{10} \end{array}$$

$$\begin{array}{r} 3. \quad 147 \\ - 75 \\ \hline 72 \end{array}$$

$$\begin{array}{r} 10010011 \\ + 01001011 \\ \hline 10001000 = 72_{10} \end{array}$$

### 3. PERKALIAN

- ✓ Perkalian biner pada dasarnya sama dengan perkalian desimal, nilai yang dihasilkan hanya "0" dan "1"
- ✓ Bergeser satu ke kanan setiap dikalikan 1 bit pengali
- ✓ Setelah proses perkalian masing-masing bit pengali selesai, lakukan penjumlahan masing-masing kolom bit hasil

*Desimal*

$$\begin{array}{r} 13 \\ \times 11 \\ \hline 13 \\ 13 \\ \hline 143 \end{array}$$

*Biner*

$$\begin{array}{r} 1101 \\ \times 1011 \\ \hline 1101 \\ 1101 \\ 0000 \\ 1101 \\ \hline 10001111 = 143_{10} \end{array}$$

→ yang dikalikan  
→ pengali

→ hasil kali

## 4. PEMBAGIAN

- ✓ Pembagian biner pada dasarnya sama dengan pembagian desimal, nilai yang dihasilkan hanya "0" dan "1"
- ✓ 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 di bawah nilai pembagi, maka hasil bagi = 0.

<i>Desimal</i>		<i>Biner</i>	
$\begin{array}{r} 3 \\ 3 \overline{) 9} \\ \underline{- 9} \\ 0 \end{array}$		$\begin{array}{r} 11 \\ \hline 011 \overline{) 1001} \\ \underline{- 011} \\ 0011 \\ \underline{- 011} \\ 0 \end{array}$	$= 3_{10}$ ← <i>hasil bagi</i>
	<i>pembagi</i>		← <i>yang dibagi</i>

# FUNGSI ARITMETIKA untuk sistim bilangan lain

## 1. PENJUMLAHAN

**OCTAL**

Contoh :

$$\begin{array}{r} 73 \\ + 15 \\ \hline 110 \end{array}$$

**HEXADECIMAL**

Contoh :

$$\begin{array}{r} 1D3 \\ + 39 \\ \hline 20C \end{array}$$

**BCD**

Contoh :

$$\begin{array}{r} 47 \quad 0100 \quad 0111 \\ + 15 \quad 0001 \quad 0101 \\ \hline 62 \quad 0101 \quad 1100 \\ \quad \quad \quad 0110 \\ \hline 0110 \quad 0010 \\ \underbrace{\quad\quad} \quad \underbrace{\quad\quad} \\ 6 \quad 2 \end{array}$$

invalid ( $> 9$ ),  
tambahkan 6 (0110)

## 2. PENGURANGAN

**OCTAL**

Contoh :

$$\begin{array}{r} 62 \\ - 34 \\ \hline 26 \end{array}$$

**HEXADECIMAL**

Contoh :

$$\begin{array}{r} 1D3 \\ - 9F \\ \hline 134 \end{array}$$

**BCD**

Contoh :

$$\begin{array}{r} 56 \quad 0101 \quad 0110 \\ - 34 \quad 0011 \quad 0100 \\ \hline 22 \quad 0010 \quad 0010 \\ \quad \underbrace{\hspace{1.5cm}} \quad \underbrace{\hspace{1.5cm}} \\ \quad \quad 2 \quad \quad 2 \end{array}$$

### 3. PERKALIAN

**OCTAL**

Contoh :

$$\begin{array}{r} 14 \\ \times 13 \\ \hline 44 \\ 14 \\ \hline 204 \end{array}$$

**HEXADECIMAL**

Contoh :

$$\begin{array}{r} 1E2 \\ \times 25 \\ \hline 96A \\ 3C4 \\ \hline 45AA \end{array}$$

## 4. PEMBAGIAN

**OCTAL**

Contoh :

$$\begin{array}{r} 62 \\ \hline 5/372 \\ - 36 \\ \hline 12 \\ - 12 \\ \hline 0 \end{array}$$

**HEXADECIMAL**

Contoh :

$$\begin{array}{r} 64 \\ \hline F/5DC \\ - 5A \\ \hline 3C \\ - 3C \\ \hline 0 \end{array}$$



## Soal Latihan

1. Konversikan command berikut ini ke dalam kode ASCII :

```
BEGIN( )  
23:LD A,100h;  
LD B,20h;  
ADD A,B;  
GOTO 23;  
END;
```

2. Sebuah urutan data diterima : 1010101  
dimana :  $e_1 = 1$     $e_2 = 0$     $e_4 = 0$

Dengan kode Hamming, tentukan bit di posisi mana yang salah ? Berapa nilai data asli (sebelum ditambah bit pengecek) ?

3. Selesaikan seluruh operasi aritmetika berikut menggunakan sistim bilangan :

1) *biner*

2) *oktal*

3) *hexadecimal*

a.  $19 + 3 = \dots$

c.  $22 - 8 = \dots$

b.  $12 \times 5 = \dots$

d.  $48 : 12 = \dots$

# RANGKAIAN ARITMETIKA 3

## *Pokok Bahasan :*

1. Bilangan biner bertanda (positif dan negatif)
2. Sistem 1'st dan 2's-complement
3. Rangkaian Aritmetika : Adder, Subtractor
4. Arithmetic/Logic Unit

## *Tujuan Instruksional Khusus :*

1. Mahasiswa dapat membentuk bilangan biner bertanda dari bilangan desimal positif dan negatif
2. Mahasiswa dapat melakukan operasi penjumlahan bilangan-bilangan biner bertanda dengan bentuk 2's complement
3. Mahasiswa dapat membuat rangkaian Adder dan Subtractor
4. Mahasiswa dapat menggunakan IC Arithmetic/Logic Unit

# BILANGAN BINER BERTANDA

+5 → 0 0101

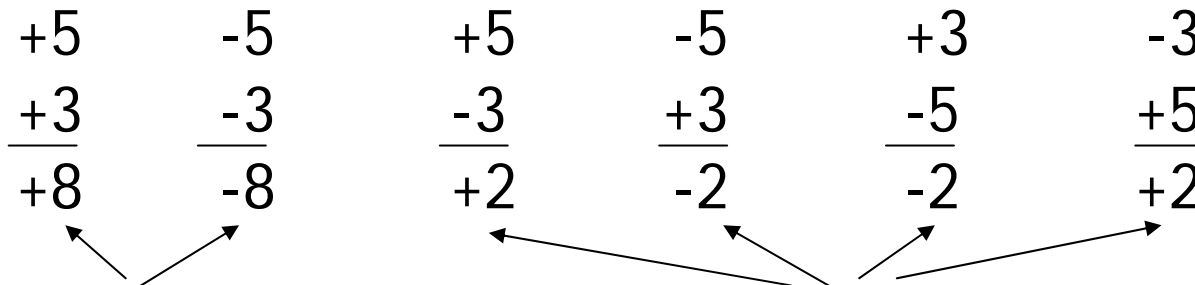
Tanda + dinyatakan sebagai biner "0"

-5 → 1 0101

Tanda - dinyatakan sebagai biner "1"



Tanda di depan bilangan membingungkan dalam menyatakan besaran dari bilangan itu sendiri



Hanya menjumlahkan besaran dari 2 bilangan, tanda sesuai dengan tanda kedua bilangan

Merupakan pengurangan dari bilangan besar dengan bilangan kecil, tanda mengikuti bilangan yang besar

# SISTIM 1'S dan 2'S COMPLEMENT

## *1'S COMPLEMENT*

Bilangan Komplemen :  $\left\{ \begin{array}{l} \text{Biner "0" menjadi "1"} \\ \text{Biner "1" menjadi "0"} \end{array} \right.$

*Contoh :*

Carilah komplemen dari 10110

10110  $\rightarrow$  komplemen-nya : 01001

Carilah komplemen dari 110

110  $\rightarrow$  komplemen-nya : 001

## *2'S COMPLEMENT*

- Bentuk ini banyak digunakan dalam sistim komputer untuk memproses persamaan aritmetika dan bilangan biner.
- Dengan bentuk ini mudah membedakan bilangan biner positif dan negatif

### *Cara membuat 2's Complement :*

1. Jika yang diketahui adalah bilangan desimal, jadikan ke bentuk biner.
2. Apabila bilangan tersebut bertanda +, biarkan ke bentuk biner yang sudah ada
3. Apabila bilangan tersebut bertanda -, lakukan cara sbb :
  - a. Carilah komplemen dari bilangan biner-nya.
  - b. Tambahkan 1.
  - c. Untuk kembali ke bentuk desimal, lakukan konversi biner ke desimal

Contoh :

1. Konversikan  $+35_{10}$  ke bentuk 2's complement-nya

Jawab :

$$35 = 010011$$

$$2's\ compl : \mathbf{010011}$$

2. Konversikan  $-35_{10}$  ke bentuk 2's complement-nya

Jawab :

$$35 = 010011$$

$$1's\ compl : 101100$$

$$+ 1 : 1$$

$$2's\ compl : \mathbf{101101}$$

3. Konversikan bentuk 2's complement 1101 1101 kembali ke bentuk desimal-nya

Jawab :

2's compl : 1101 1101

1's compl : 0010 0010

+ 1 : 1

biner : 0010 0011

desimal : -35

4. Konversikan  $-98_{10}$  ke bentuk 2's complement-nya

Jawab :

biner : 0110 0010

1's compl : 1001 1101

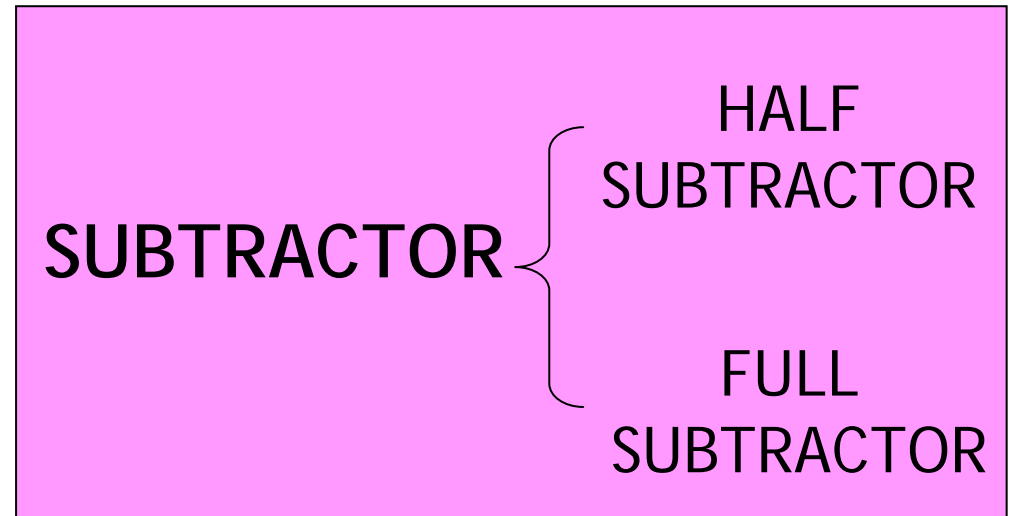
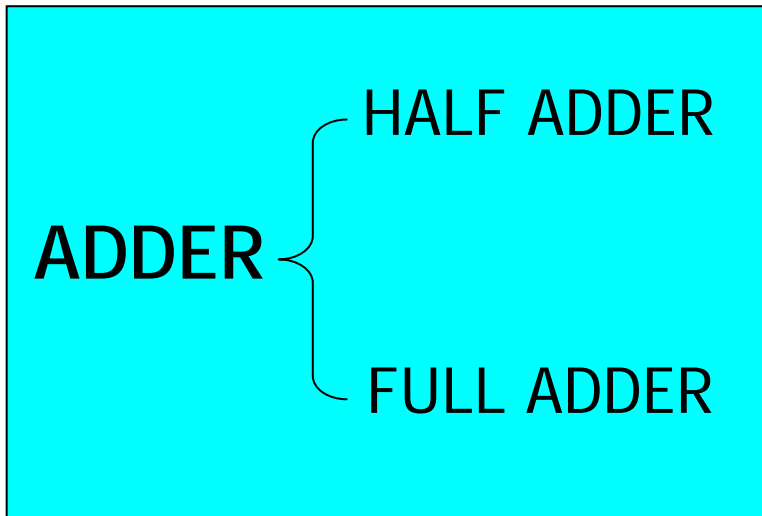
+ 1 : 1

2's compl : 1001 1110



# RANGKAIAN ARITMETIKA

- Rangkaian Aritmetika yang dipelajari di sini adalah rangkaian Adder (penjumlah) dan Subtractor (pengurang)
- Bentuk data yang dijumlah / dikurangkan adalah BINER
- Adder merupakan dasar dari Multiplier (Perkalian)
- Subtractor merupakan dasar dari Divider (Pembagian)



## *HALF ADDER*

Merupakan implementasi operasi penjumlahan dasar dua bilangan

$$A_0 + B_0 = \Sigma_0 + C_{out}$$

$A_0$	←	<i>Augend / yang dijumlahkan</i>
+ $B_0$	←	<i>Addend / penjumlah</i>
<hr/>		
$\Sigma_0$	←	<i>Sum / hasil</i>
+ $C_{out}$	←	<i>Carry</i>

## Tabel Kebenaran untuk Penjumlahan 2 bit biner (LSB)

$A_0$	$B_0$	$\Sigma_0$	$C_{out}$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

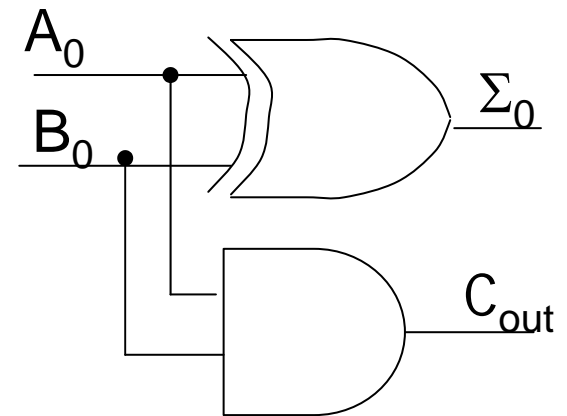
Dari Tabel Kebenaran, dapatkan persamaan untuk  $\Sigma_0$  dan  $C_{out}$  (menggunakan K-Map)

$A_0 \backslash B_0$	0	1
0	0	1
1	1	0

$$\begin{aligned}\Sigma_0 &= \overline{A_0} \cdot B_0 + A_0 \cdot \overline{B_0} \\ &= A_0 \oplus B_0\end{aligned}$$

$A_0 \backslash B_0$	0	1
0	0	0
1	0	1

$$C_{out} = A_0 \cdot B_0$$

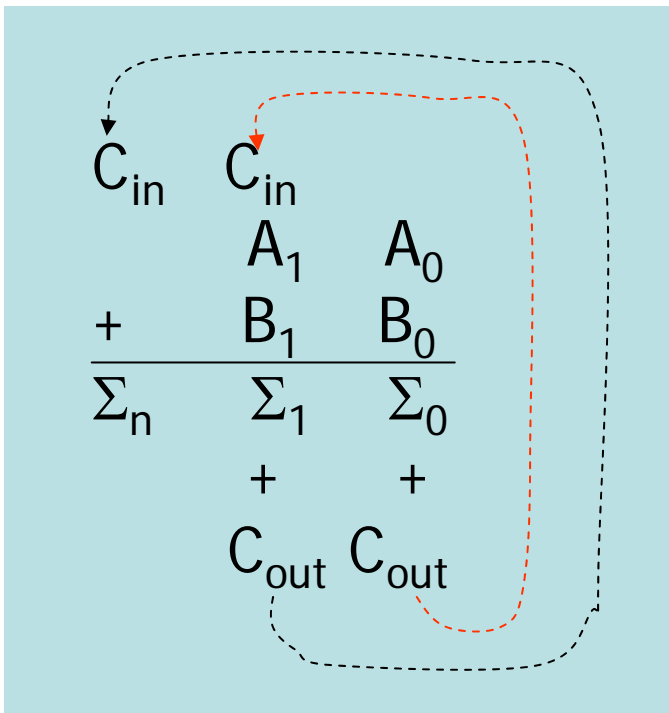


Rangkaian Half Adder

# *FULL ADDER*

Merupakan implementasi operasi penjumlahan dasar dua bilangan

$$A_i + B_i + C_{in} = \Sigma_i + C_{out} \quad i = 2,3,4,\dots$$



## Tabel Kebenaran untuk Penjumlahan 2 bit biner (lanjut)

$A_1$	$B_1$	$C_{in}$	$\Sigma_1$	$C_{out}$
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 Kebenaran, dapatkan persamaan untuk  $\Sigma_0$  dan  $C_{out}$  (menggunakan K-Map)

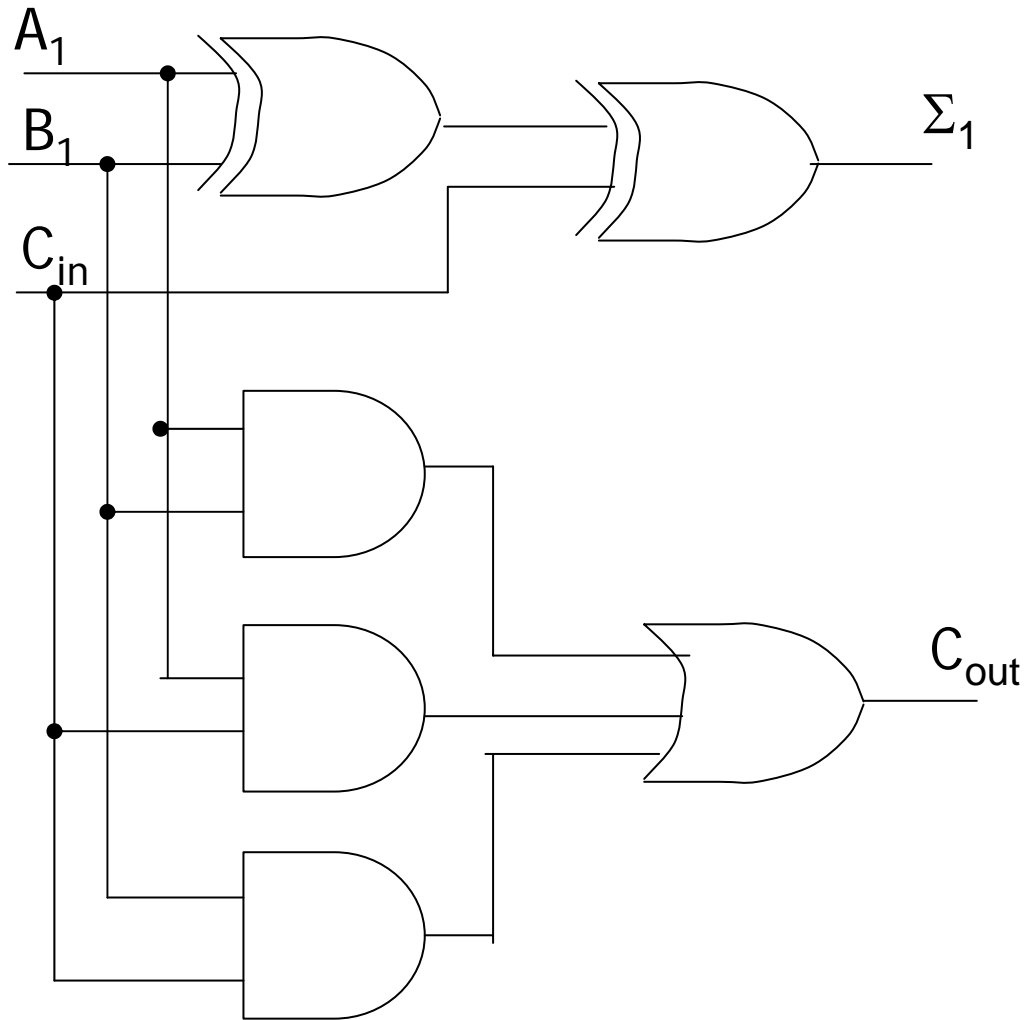
$B_1 C_{in}$ $A_1$	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$$\begin{aligned}\Sigma_1 &= \bar{A}_1 \bar{B}_1 C_{in} + A_1 \bar{B}_1 \bar{C}_{in} \\ &\quad + \bar{A}_1 B_1 \bar{C}_{in} + A_1 B_1 C_{in} \\ &= A_1 \oplus B_1 \oplus C_{in}\end{aligned}$$

$B_1 C_{in}$ $A_1$	00	01	11	10
0	0	0	1	0
1	0	1	1	1

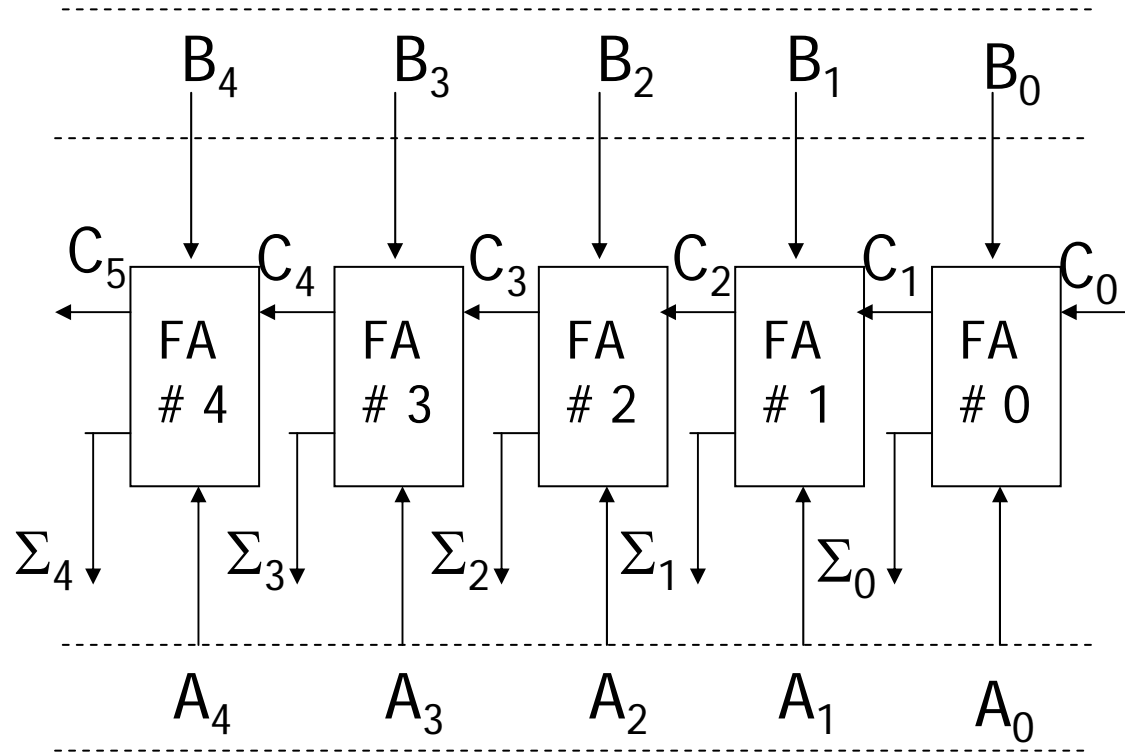
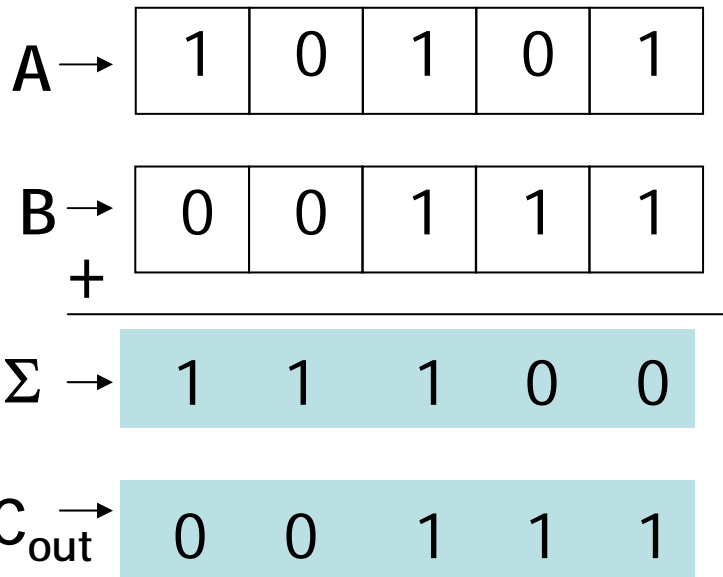
$$C_{out} = A_1 C_{in} + A_1 B_1 + B_1 C_{in}$$

# Rangkaian Full Adder

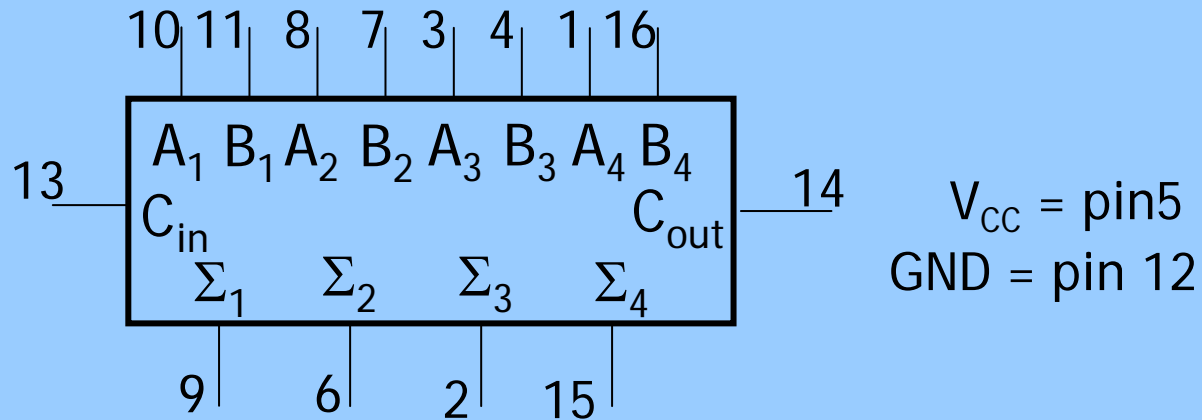


# PARALLEL ADDER

Terdiri dari beberapa Full adder yang dirangkai seri, sehingga dapat melakukan operasi penjumlahan dua bilangan dengan lebih dari 1 bit biner



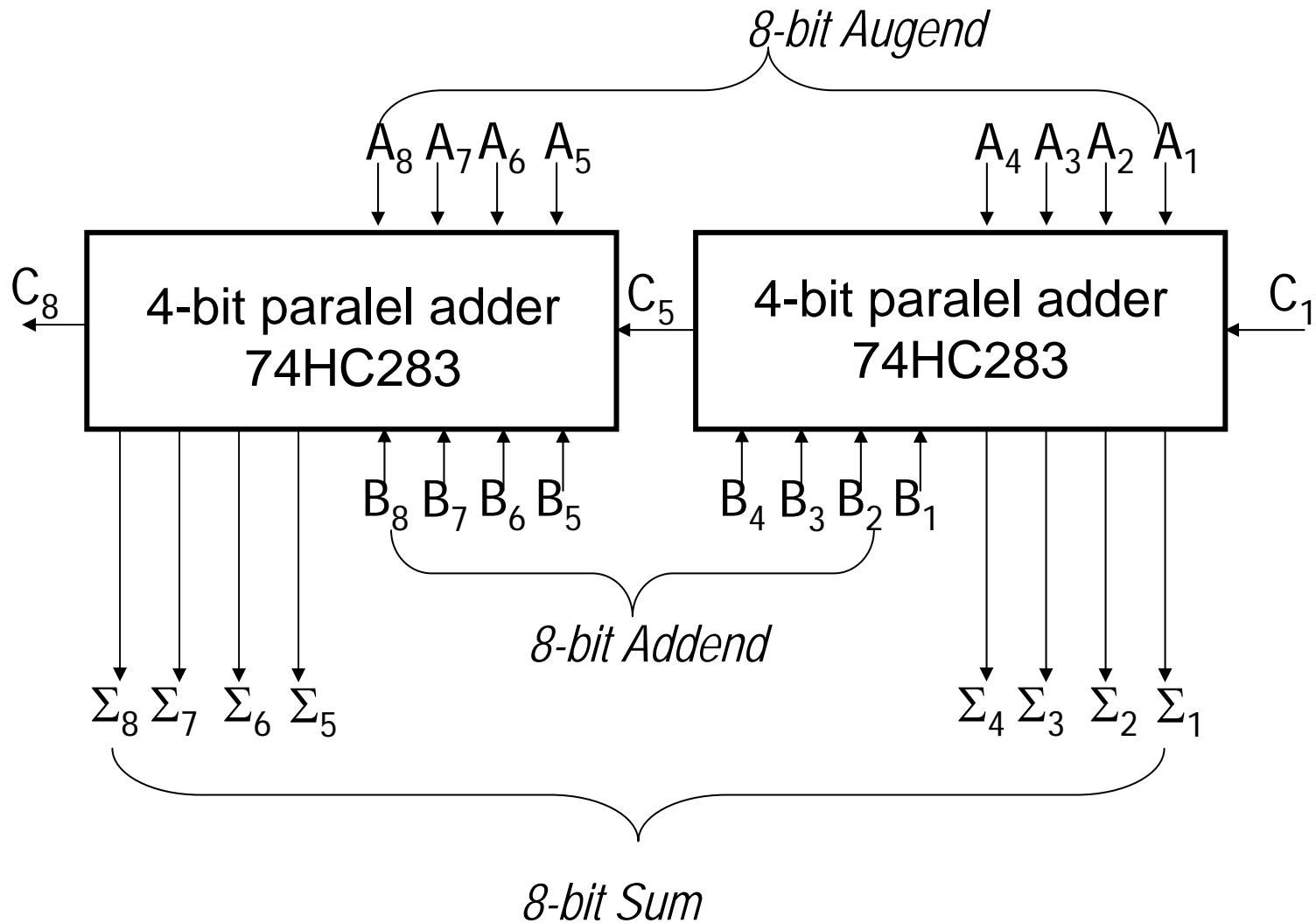
# IC PARALLEL ADDER (74HC283)



$A_1 - A_4 = \text{Augend}$   
 $B_1 - B_4 = \text{Addend}$   
 $\Sigma_1 - \Sigma_4 = \text{Sum}$   
 $C_{in} = \text{Carry In}$   
 $C_{out} = \text{Carry out}$



## 74HC283 sebagai Adder 8-bit



# HALF SUBTRACTOR

Merupakan implementasi operasi pengurangan dasar dua bilangan

$$A_0 - B_0 = R_0 + B_{out}$$

$$\begin{array}{r} A_0 \\ - B_0 \\ \hline R_0 \\ + \\ B_{out} \end{array}$$

Tabel Kebenaran untuk Pengurangan 2 bit biner (LSB)

$A_0$	$B_0$	$R_0$	$B_{out}$
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Dari Tabel Kebenaran, dapatkan persamaan untuk  $R_0$  dan  $B_{out}$  (menggunakan K-Map)

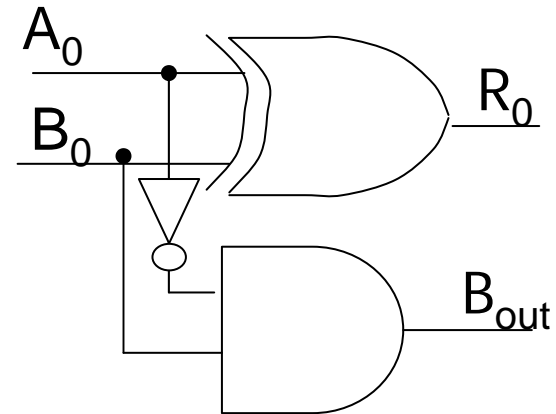
$A_0 \backslash B_0$	0	1
0	0	1
1	1	0

$$R_0 = \overline{A_0} \cdot B_0 + A_0 \cdot \overline{B_0}$$

$$= A_0 \oplus B_0$$

$A_0 \backslash B_0$	0	1
0	0	1
1	0	0

$$B_{out} = \overline{A_0} \cdot B_0$$

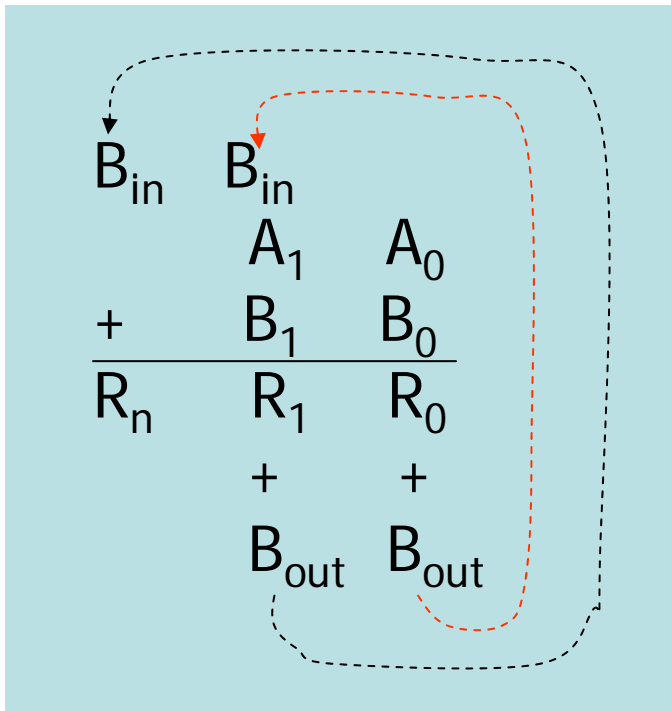


Rangkaian Half Subtractor

# FULL SUBTRACTOR

Merupakan implementasi operasi pengurangan dasar dua bilangan

$$A_i - B_i - B_{in} = R_i + B_{out} \quad i = 2, 3, 4, \dots$$



## Tabel Kebenaran untuk Pengurangan 2 bit biner (lanjut)

$A_1$	$B_1$	$B_{in}$	$R_1$	$B_{out}$
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

Dari Tabel Kebenaran, dapatkan persamaan untuk  $\Sigma_0$  dan  $C_{out}$  (menggunakan K-Map)

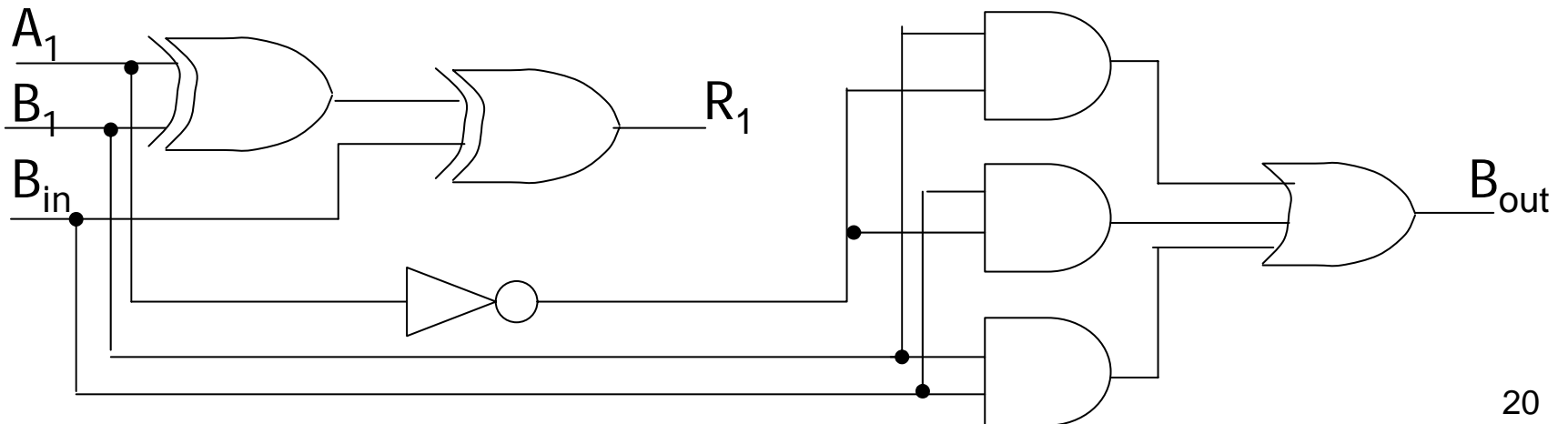
$B_1 B_{in}$		00	01	11	10
$A_1$	0	0	1	0	1
	1	1	0	1	0

$B_1 B_{in}$		00	01	11	10
$A_1$	0	0	1	1	1
	1	0	0	1	0

$$\begin{aligned}
 R_1 &= \bar{A}_1 \bar{B}_1 B_{in} + A_1 \bar{B}_1 \bar{B}_{in} \\
 &\quad + \bar{A}_1 B_1 \bar{B}_{in} + A_1 B_1 B_{in} \\
 &= A_1 \oplus B_1 \oplus B_{in}
 \end{aligned}$$

$$B_{out} = \bar{A}_1 B_{in} + \bar{A}_1 B_1 + B_1 B_{in}$$

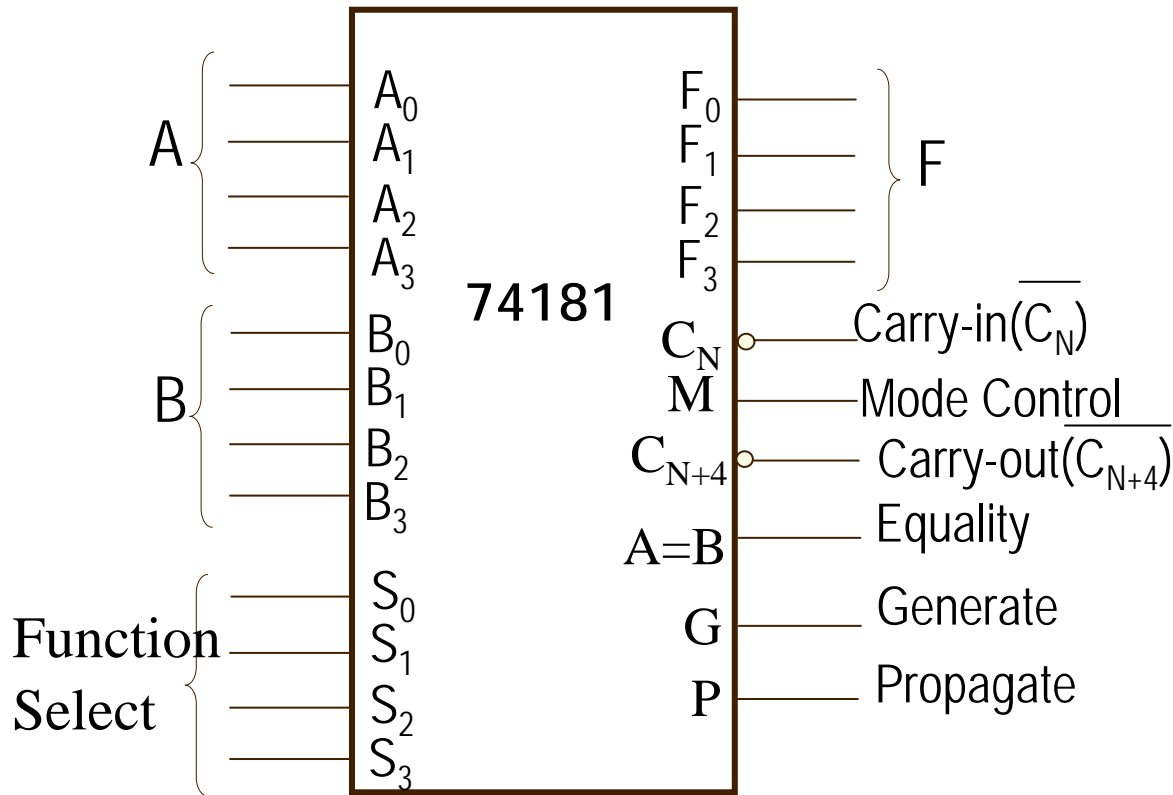
### Rangkaian Full Subtractor



# ARITHMETIC/LOGIC UNIT (ALU)

Merupakan paket Large Scale Integrated-Circuit (LSI).

Mempunyai dua jenis operasi, yaitu : Aritmetika dan Logika



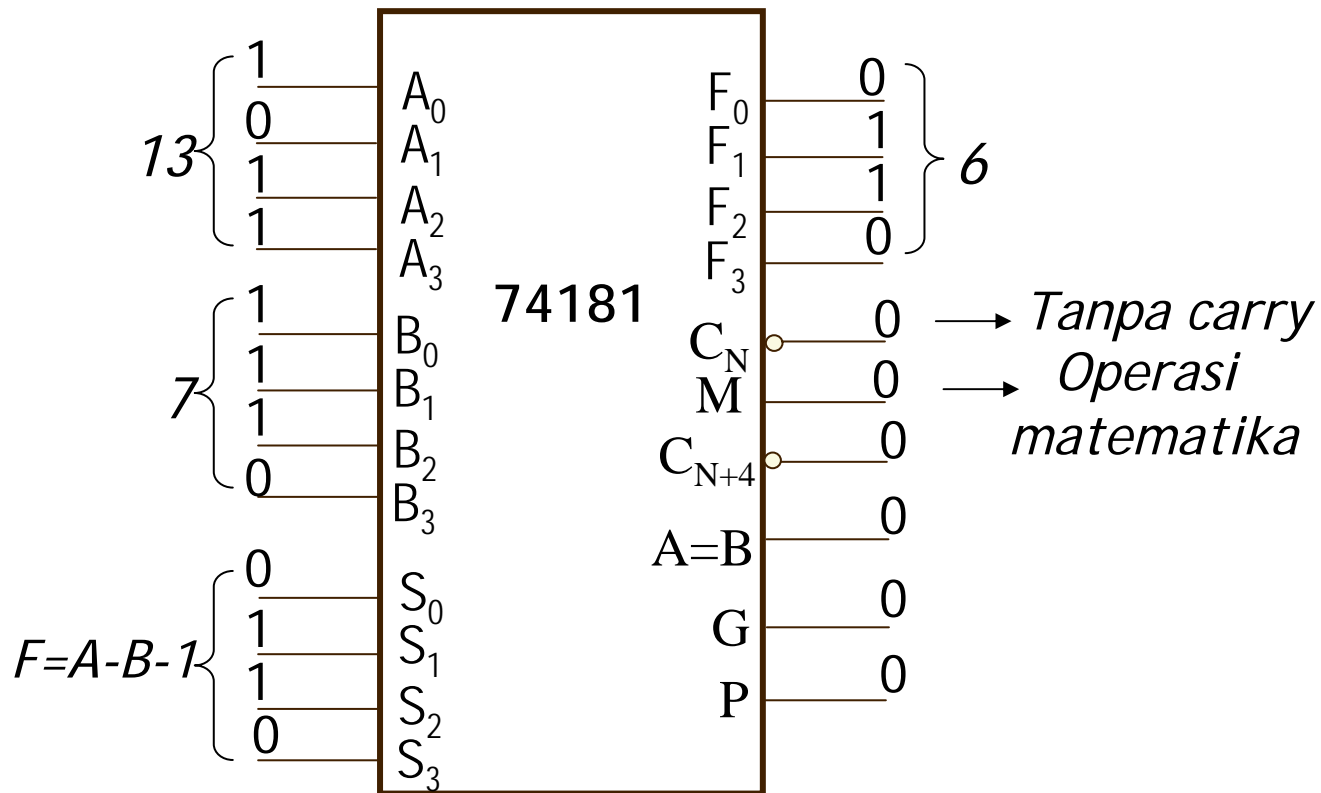
# Tabel Fungsi ALU 74181

SELECTION				M=H Logic Function	M= L Aritmetic Operation
S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>		C <sub>n</sub> =H (no carry)
L	L	L	L	$F = A'$	$F=A$
L	L	L	H	$F = (A+B)'$	$F=A+B$
L	L	H	L	$F=A'B$	$F=A+B'$
L	L	H	H	$F = 0$	$F=\text{minus } 1 \text{ (2's comp)}$
L	H	L	L	$F=(AB)'$	$F=A \text{ plus } AB'$
L	H	L	H	$F=B'$	$F=(A+B) \text{ plus } AB'$
L	H	H	L	$F=A \oplus B$	$F=A \text{ minus } B \text{ minus } 1$
L	H	H	H	$F=AB'$	$F=AB' \text{ minus } 1$
H	L	L	L	$F=A'+B$	$F=A \text{ plus } AB$
H	L	L	H	$F=(A \oplus B)'$	$F=A \text{ plus } B$
H	L	H	L	$F=B$	$F=(A+B') \text{ plus } AB$
H	L	H	H	$F=AB$	$F=AB \text{ minus } 1$
H	H	L	L	$F=1$	$F=A \text{ plus } A^*$
H	H	L	H	$F=A+B'$	$F=(A+B) \text{ plus } A$
H	H	H	L	$F=A+B$	$F=(A+B') \text{ plus } A$
H	H	H	H	$F=A$	$F=A \text{ minus } 1$



## Contoh :

Tunjukkan bagaimana meng-implementasi kan pengurangan 13 - 7 menggunakan 74181



## Soal Latihan

1. Konversikan :

*Desimal*  $\rightarrow$  *8-bit 2's complement*

a) 12      b) -15      c) -112      d) 125

*2's complement*  $\rightarrow$  *desimal*

a) 0101 1100      b) 1110 1111      c) 1000 0011

2. Selesaikan operasi aritmetika berikut menggunakan bentuk 2's complement

a) 5	b) 32	c) -28	d) -38
<u>+7</u>	<u>-18</u>	<u>35</u>	<u>-46</u>

3. Selesaikan operasi penjumlahan berikut menggunakan bentuk BCD

$$\begin{array}{r} \text{a) } 8 \\ +3 \\ \hline \end{array}$$

$$\begin{array}{r} \text{b) } 43 \\ +72 \\ \hline \end{array}$$

$$\begin{array}{r} \text{c) } 7 \\ +38 \\ \hline \end{array}$$

$$\begin{array}{r} \text{d) } 80 \\ +23 \\ \hline \end{array}$$

4. Ubahlah rangkaian Half Adder hanya menggunakan gerbang NOR saja

5. Buat rangkaian 4-bit Parallel Adder menggunakan 3 buah rangkaian Full Adder dan 1 buah Half Adder