RANGKAIAN ARITMETIKA

Materi :

1. Sistim Bilangan : Desimal, Biner, Oktal, Hexadesimal
2. Konversi Sistim Bilangan
3. Sistim Coding
4. Fungsi-fungsi Aritmetika Biner : penjumlahan, pengurangan, perkalian, pembagian
5. Implementasi fungsi Aritmetika pada sistim Bilangan yang lain
6. Bilangan biner bertanda (positif dan negatif)
7. Sistim 1’st dan 2’s-complement
8. Rangkaian Aritmetika : Adder, Subtractor
9. Arithmetic/Logic Unit

SISTIM BILANGAN

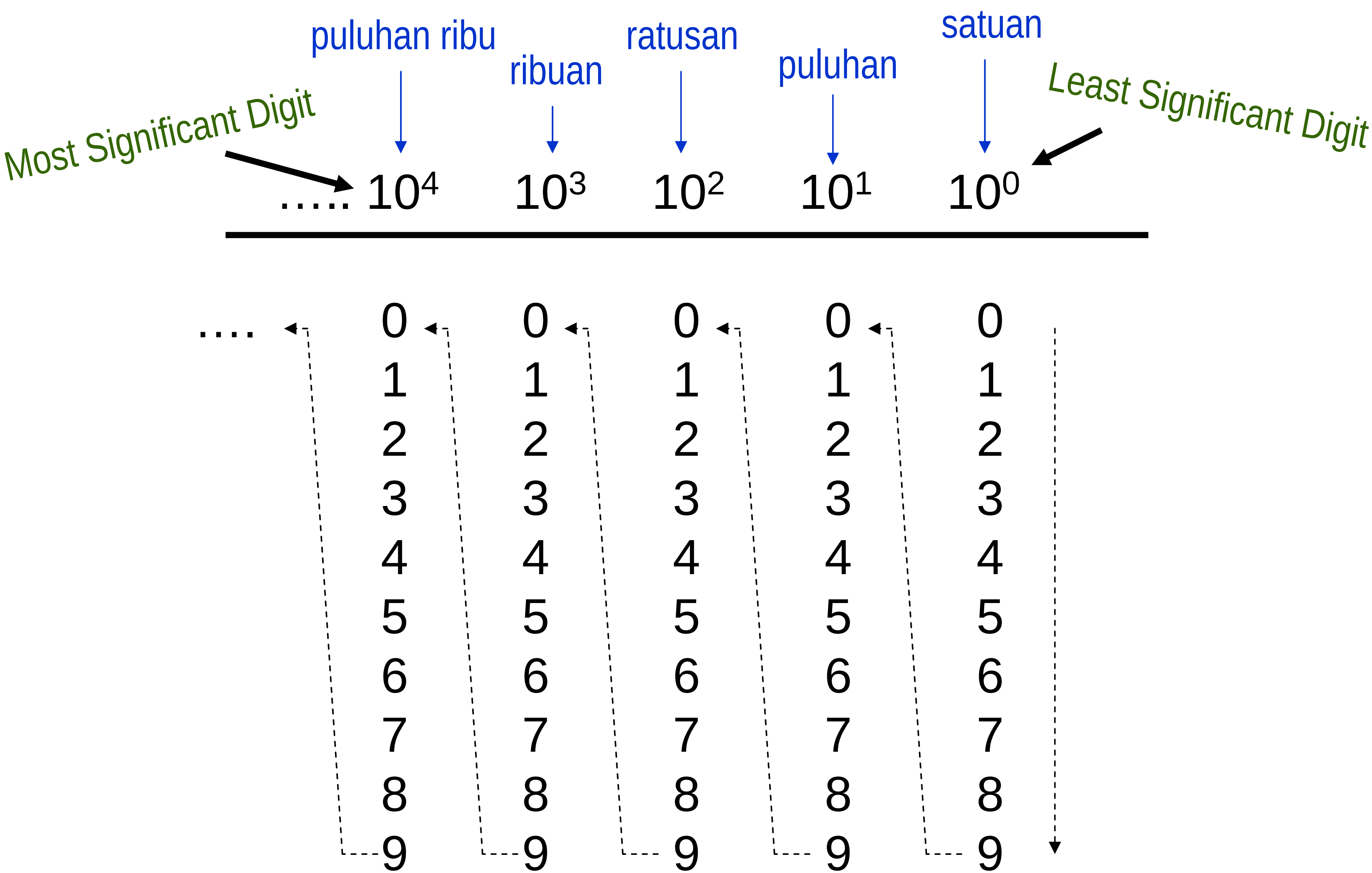
Sistim Bilangan terdiri dari :

1. Sistim Desimal → Dasar 10
2. Sistim Biner → Dasar 2
3. Sistim Oktal → Dasar 8
4. Sistim Hexadesimal → Dasar 16

Aplikasi Sistim Bilangan :

1. Sistim Desimal → nilai mata uang : puluhan, ratusan, ribuan dsb
2. Sistim Biner → rangkaian elektronika digital
3. Sistim Oktal → instruksi komputer dengan kode 3-bit
4. Sistim Hexadesimal → pengalamatan memory pada micro controller

Sistim Desimal



- Cara membilang dengan sistim desimal

0
1
.
.
9
10
11
.
.
99
100
.
.
999
1000
.
.
9999
.
.

- Cara menghitung dengan sistim desimal

Contoh :

4 6 2 3

$3 \times 10^0 = 3$

$2 \times 10^1 = 20$

$6 \times 10^2 = 600$

$4 \times 10^3 = 4000$ +

4623

(empat ribu enam ratus dua puluh tiga)

Sistim Biner

BIT = **BI**nary digi**T**

Most Significant Bit

Least Significant Bit

….. 2^4	2^3	2^2	2^1	2^0
…. 0	0	0	0	0
1	1	1	1	1

- Cara membilang dengan sistim biner

0
1
1 0
1 1
1 0 0
1 0 1
1 1 0
1 1 1
1 0 0 0
1 0 0 1
.
.

- Cara menghitung dengan sistim biner

Contoh :

1 0 1 1

$1x\ 2^0 = 1$

$1x\ 2^1 = 2$

$0x\ 2^2 = 0$

$1x\ 2^3 = 8+$

11_{10}

1 0 1 0 0 1

$1x2^0 = 1$

$0x2^1 = 0$

$0x2^2 = 0$

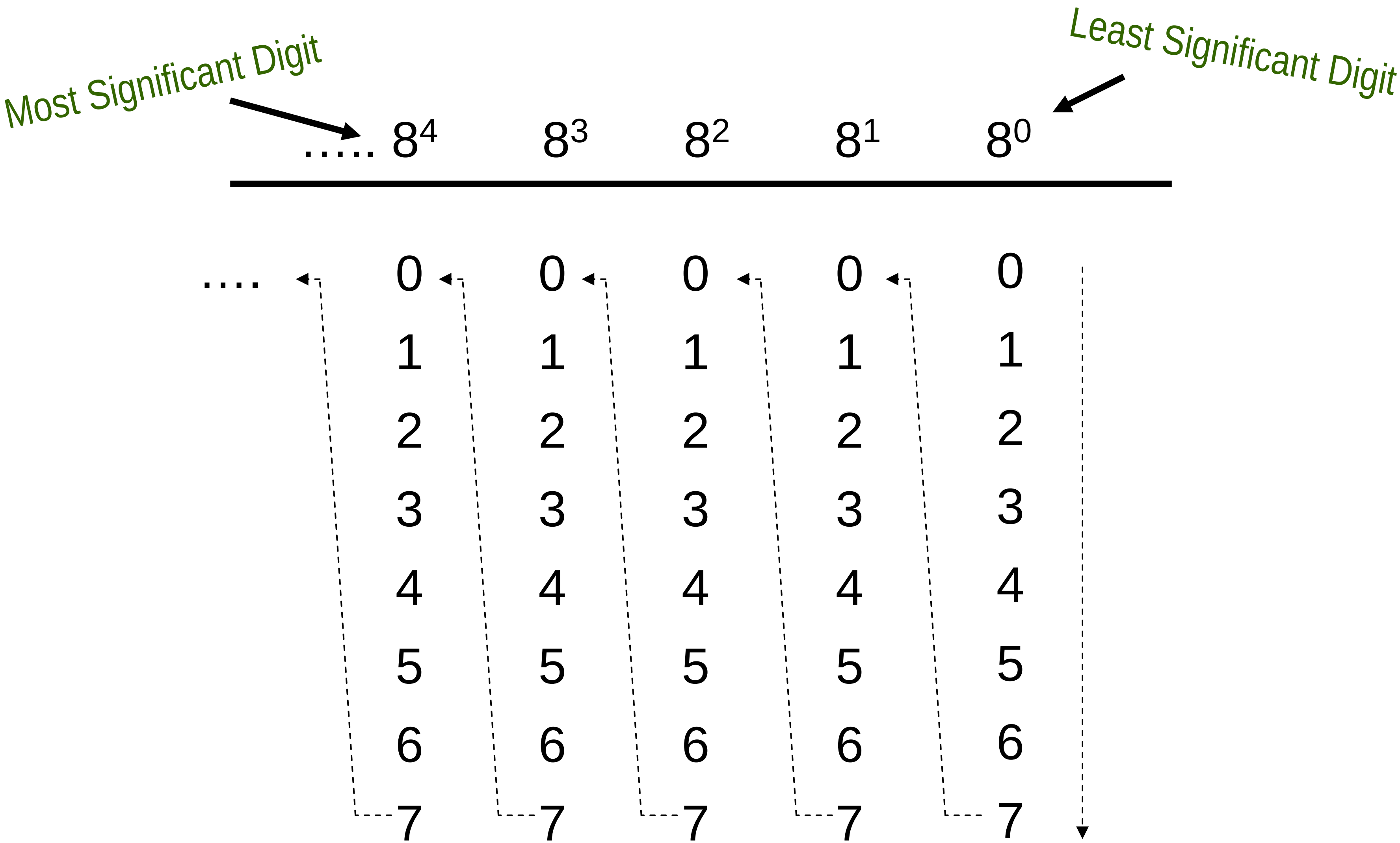
$1x2^3 = 8$

$0x2^4 = 0$

$1x2^5 = 32+$

41_{10}

Sistim Oktal



- Cara membilang dengan sistim Oktal

0
1
.
.
7
1 0
1 1
.
.
7 7
1 0 0
1 0 1
1 0 2
.
.
7 7 7
1 0 0 0
1 0 0 1
.
.

- Cara menghitung dengan sistim Oktal

Contoh :

5 6 7 4

$4 \times 8^0 = 4$

$7 \times 8^1 = 56$

$6 \times 8^2 = 384$

$5 \times 8^3 = \underline{2560} +$

3004_{10}

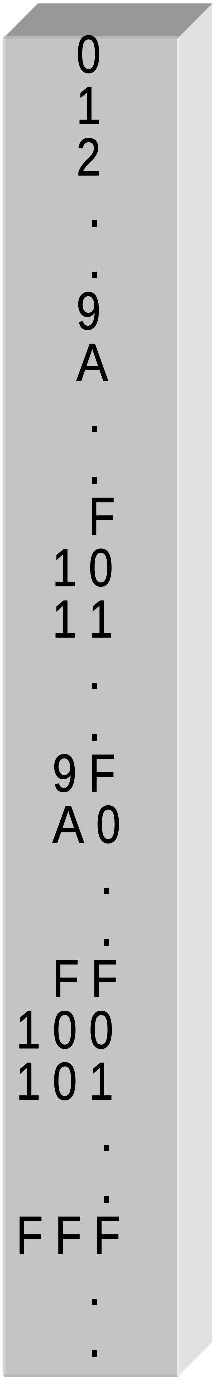
Sistim Hexadesimal

Most Significant Digit

Least Significant Digit

.....	16^4	16^3	16^2	16^1	16^0
....	0	0	0	0	0
	1	1	1	1	1
	2	2	2	2	2
	3	3	3	3	3
	4	4	4	4	4
	5	5	5	5	5
	6	6	6	6	6
	7	7	7	7	7
	8	8	8	8	8
	9	9	9	9	9
	A	A	A	A	A
	B	B	B	B	B
	C	C	C	C	C
	D	D	D	D	D
	E	E	E	E	E
	F	F	F	F	F

- Cara membilang dengan sistim Hexadesimal



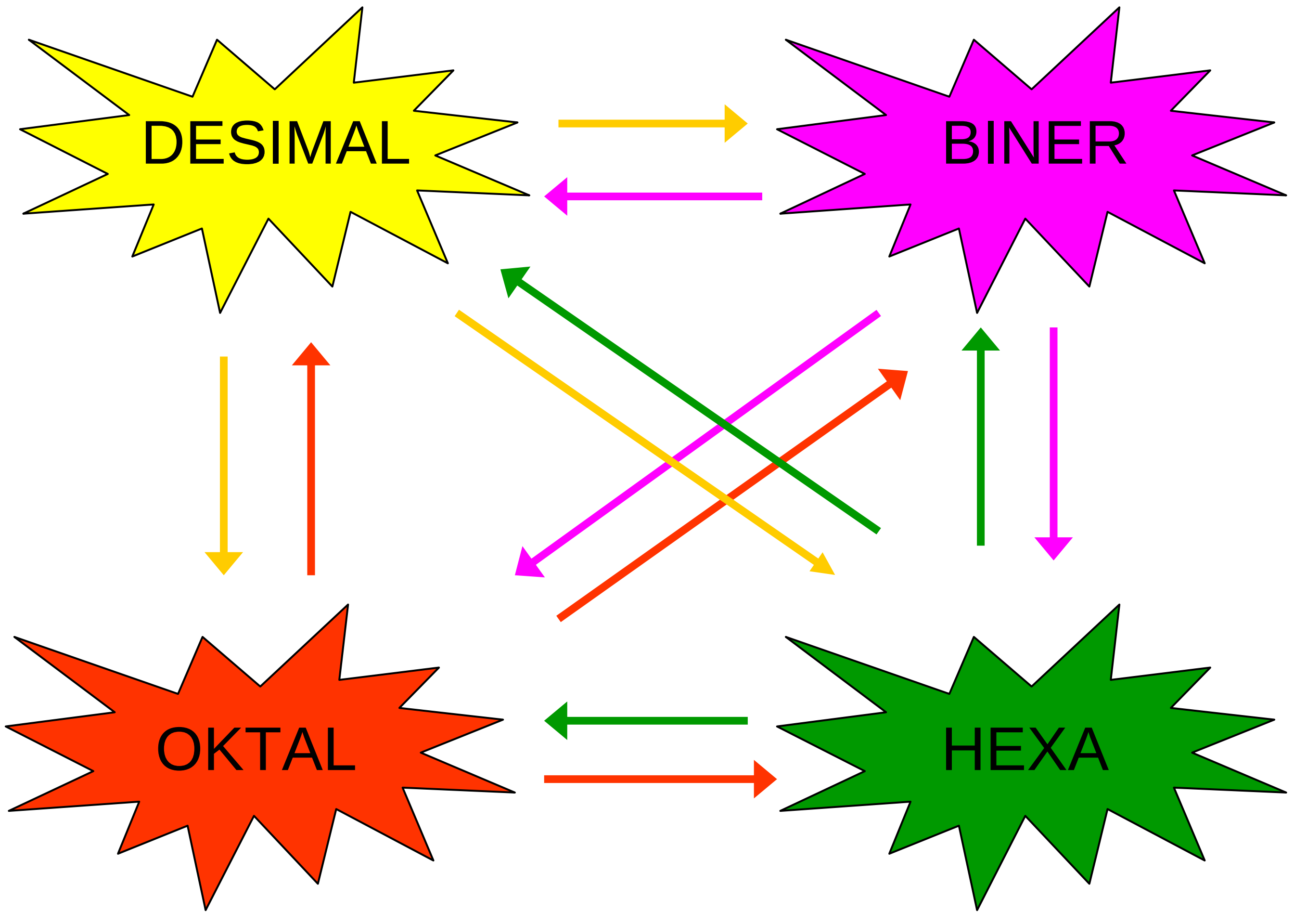
- Cara menghitung dengan sistim Hexadesimal

Contoh :

2 E 5 C

$$12 \times 16^0 = 12$$
$$5 \times 16^1 = 80$$
$$14 \times 16^2 = 3584$$
$$2 \times 16^3 = 8192 \; +$$
$$11868_{10}$$

KONVERSI SISTIM BILANGAN



1. DESIMAL → BINER

Contoh :

1) $28_{10} = \ldots\ldots\ldots_{2}$?

2	28	0	LSB
2	14	0	
2	7	1	
2	3	1	
	1		MSB

$28_{10} = 11100_{2}$

2) $345_{10} = \ldots\ldots\ldots_{2}$?

2	345	1	LSB
2	172	0	
2	86	0	
2	43	1	
2	21	1	
2	10	0	
2	5	1	
2	2	0	
	1		MSB

$345_{10} = 101011001_{2}$

2. DESIMAL → OKTAL

Contoh :

1) $28_{10} = \ldots\ldots\ {}_{8}$?

8 | 28 | 4 LSD

3 MSD

$28_{10} = 34_{8}$

2) $345_{10} = \ldots\ldots\ {}_{8}$?

8 | 345 | 1 LSD

8 | 43 | 3

5 MSD

$345_{10} = 531_{8}$

3. DESIMAL → HEXADESIMAL

Contoh :

1) $28_{10} = \ldots\ldots\ldots_{16}$?

$$\begin{array}{r|l} 16 \; \underline{28} & 12 = C \quad \text{LSD} \\ 1 & \text{MSD} \end{array}$$

$28_{10} = 1C_{16}$

2) $345_{10} = \ldots\ldots\ldots_{16}$?

$$\begin{array}{r|l} 16 \; \underline{345} & 9 \quad \text{LSD} \\ 16 \; \underline{21} & 5 \\ 1 & \text{MSD} \end{array}$$

$345_{10} = 159_{16}$

4. BINER $\rightarrow$ DESIMAL

Contoh :

1) $1101_2 = \ldots\ldots\ {}_{10}$?

$$1101_2 = 1x2^3+1x2^2+0x2^1+1x2^0$$
$$= 8 + 4 + 0 + 1$$
$$= 13_{10}$$

$\boldsymbol{1101_2 = 13_{10}}$

2) $10110111_2 = \ldots\ldots\ {}_{10}$?

$\boldsymbol{10110111_2 = 183_{10}}$

$$10110111_2 = 1x2^7+0x2^6+1x2^5+1x2^4$$
$$+ 0x2^3+1x2^2+1x2^1+1x2^0$$
$$= 128+0+32+16+0+4+2+1$$
$$= 183_{10}$$

5. OKTAL → DESIMAL

Contoh :

1) $75_8 = \ldots\ldots\ _{10}$?

$$
\begin{aligned}
75_8 &= 7x8^1 + 5x8^0 \\
&= 56 + 5 \\
&= 61_{10}
\end{aligned}
$$

$\mathbf{75_8 = 61_{10}}$

2) $6341_8 = \ldots\ldots\ _{10}$?

$$
\begin{aligned}
6341_8 &= 6x8^3 + 3x8^2 + 4x8^1 + 1x8^0 \\
&= 3072 + 192 + 32 + 1 \\
&= 3297_{10}
\end{aligned}
$$

$\mathbf{6341_8 = 3297_{10}}$

6. HEXADESIMAL → DESIMAL

Contoh :

1) $9F_{16} = \ldots\ldots\ _{10}$?

$$9F_{16} = 9 \times 16^1 + 15 \times 16^0$$
$$= 144 + 15$$
$$= 159_{10}$$

$\boldsymbol{9F_{16} = 159_{10}}$

2) $3FE8_{16} = \ldots\ldots\ _{10}$?

$$3FE8_{16} = 3 \times 16^3 + 15 \times 16^2 + 14 \times 16^1 + 8 \times 16^0$$
$$= 12288 + 3840 + 224 + 8$$
$$= 16360_{10}$$

$\boldsymbol{3FE8_{16} = 16360_{10}}$

7. BINER → OKTAL

Contoh :

$1101011_2 = \ldots\ldots\ {}_8$?

Cara 1 :

Konversikan Biner → Desimal ⟶ Desimal → Oktal

$1101011_2 = 1x2^6+1x2^5+1x2^3+1x2^1+1x2^0$

$= 64+32+8+2+1$

$= 107_{10}$

$8 \mid 107$ sisa 3

$8 \mid 13$ sisa 5

1

$1101011_2 = 153_8$

Cara 2 :

Ambil per – 3bit menjadi 1 kelompok, mulai dari LSB.

Bit MSB ditambahkan “0”

$1101011 \rightarrow \underbrace{001}_{1}\ \underbrace{101}_{5}\ \underbrace{011}_{3_8}$

8. BINER → HEXADESIMAL

Contoh :

$1101011_2 = \ldots\ldots\ldots_{16}$?

Cara 1 :

Konversikan Biner → Desimal ⟶ Desimal → Hexadesimal

$1101011_2 = 1x2^6+1x2^5+1x2^3+1x2^1+1x2^0$

$= 64+32+8+2+1$

$= 107_{10}$

$$16 \, \underline{107} \; 11=C$$
$$6$$

$1101011_2 = 6C_{16}$

Cara 2 :

Ambil per – 4bit menjadi 1 kelompok, mulai dari LSB.

Bit MSB ditambahkan “0”

$1101011 \rightarrow \underbrace{0110}_{6} \; \underbrace{1011}_{C} {}_{16}$

9. OKTAL → BINER

Contoh :

$64_8 = \ldots\ldots\ldots_2$?

Cara 1 :

Konversikan Oktal → Desimal ⟶ Desimal → Biner

$64_8 = 6x8^1+4x8^0$

$= 48 + 4$

$= 52_{10}$

2	52	0
2	26	0
2	13	1
2	6	0
2	3	1
	1	

$64_8 = 110100_2$

Cara 2 :

Masing-masing digit dikonversikan menjadi 3 bit biner.

64 → 6 4

110 100_2

10. HEXADESIMAL → BINER

Contoh :

$7D_{16}$ = ……. $_2$?

Cara 1 :

Konversikan Hexa → Desimal ⟶ Desimal → Biner

$7D_{16} = 7x16^1+13x16^0$

$= 112 + 14$

$= 125_{10}$

2	125	1
2	62	0
2	31	1
2	15	1
2	7	1
2	3	1
	1	

$7D_{16} = 1111101_2$

Cara 2 :

Masing-masing digit dikonversikan menjadi 4 bit biner.

7D → 7 D

0111 1101_2

11. OKTAL → HEXADESIMAL

Contoh :

$57_8 = \ldots\ldots\ _{16}$?

Cara 1 :

Konversikan Oktal → Desimal ⟶ Desimal → Hexa

$57_8 = 5x8^1+7x8^0$

$= 40 + 7$

$= 47_{10}$

$16 \,\underline{\;47\;}\, 15=F$

2

$57_8 = 2F_{16}$

Cara 2 :

Konversikan Oktal → Biner ⟶ Biner → Hexa

57 → 5 7

101 111_2

$\underbrace{0010}_{2}\ \underbrace{1111}_{F}\ _{16}$

12. HEXADESIMAL → OKTAL

Contoh :

$6A_{16} = \ldots\ldots\ _8$?

Cara 1 :

Konversikan Hexa → Desimal ⟶ Desimal → Oktal

$6A_{16} = 6x16^1+10x16^0$

$= 96 + 10$

$= 106_{10}$

$$\begin{array}{r|l} 8 & 106 \quad 2 \\ 8 & 13 \quad 5 \\ & 1 \end{array}$$

$6A_{16} = 152_8$

Cara 2 :

Konversikan Hexa → Biner ⟶ Biner → Oktal

6A → 6 A

0110 1010_2

$\underbrace{001}_{1}\ \underbrace{101}_{5}\ \underbrace{010}_{2}\ _8$

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 :

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

Nilai desimal

Nilai bobot (tergantung jenis kode pembobot)

Contoh :

1) $7_{10} = \ldots\ldots_{BCD\ (8421)}$?

$7_{10} = 8x0 + 4x1 + 2x1 + 1x1 \rightarrow 7_{10} = 0111_{BCD(8421)}$

2) $18_{10} = \ldots\ldots_{BCD\ (5421)}$?

$18_{10} = 5x0 + 4x0 + 2x0 + 1x1 \quad 5x1 + 4x0 + 2x1 + 1x1$
$= 0001\ \ 1011_{BCD(5421)}$

3) $48_{10} = \ldots\ldots_{BCD\ (2421)}$?

$48_{10} = 2x0 + 4x1 + 2x0 + 1x0 \quad 2x1 + 4x1 + 2x1 + 1x0$
$= 0100\ \ 1110_{BCD(2421)}$

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 $10010_{\text{2 of 5}}$

Ringkasan Kode BCD

Digit desimal	Kode 8421	Kode 5421	Kode 2421	Kode Excess-3	Kode 2 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	1010	0101	0101	0000	sembarang
digunakan	1011	0110	0110	0001	pola
	1100	0111	0111	0010	yg lain
	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 → 2^7 = 128 kombinasi kode

7 bit → 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 f 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

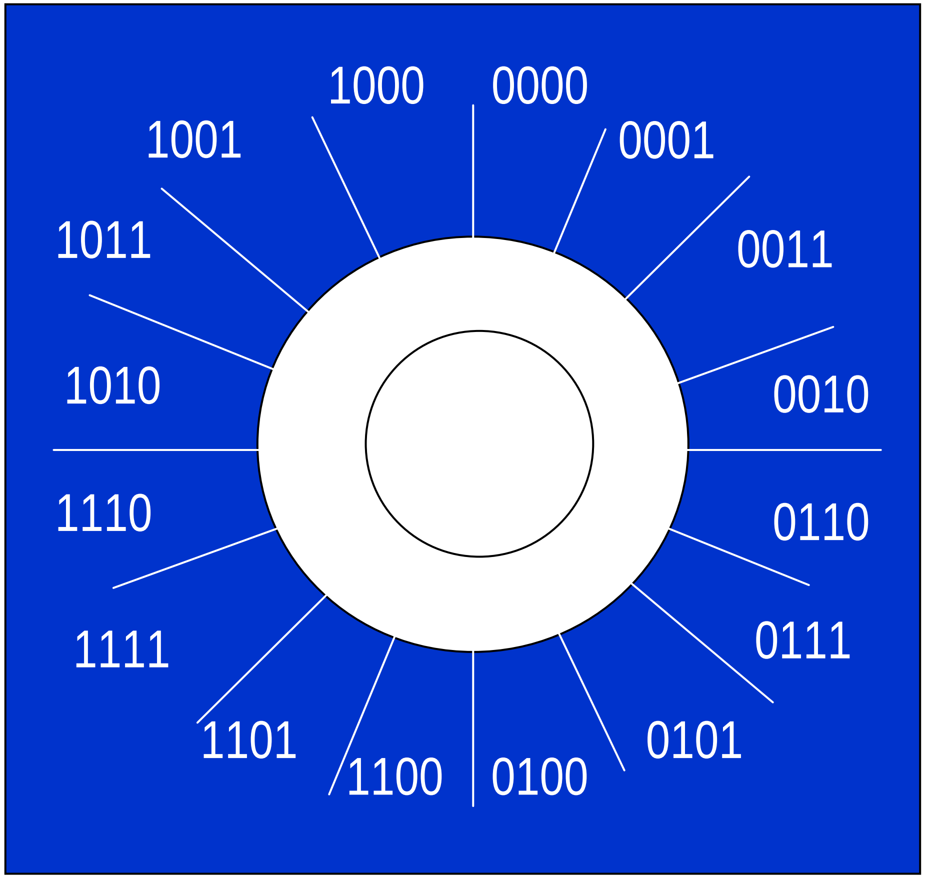
M = 100 1101

3. Gray Code

Digunakan dalam peng-kode an 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^o.

Masing-masing kode digunakan untuk perbedaan sudut $22{,}5^o$ ($= 360^o/16$)



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

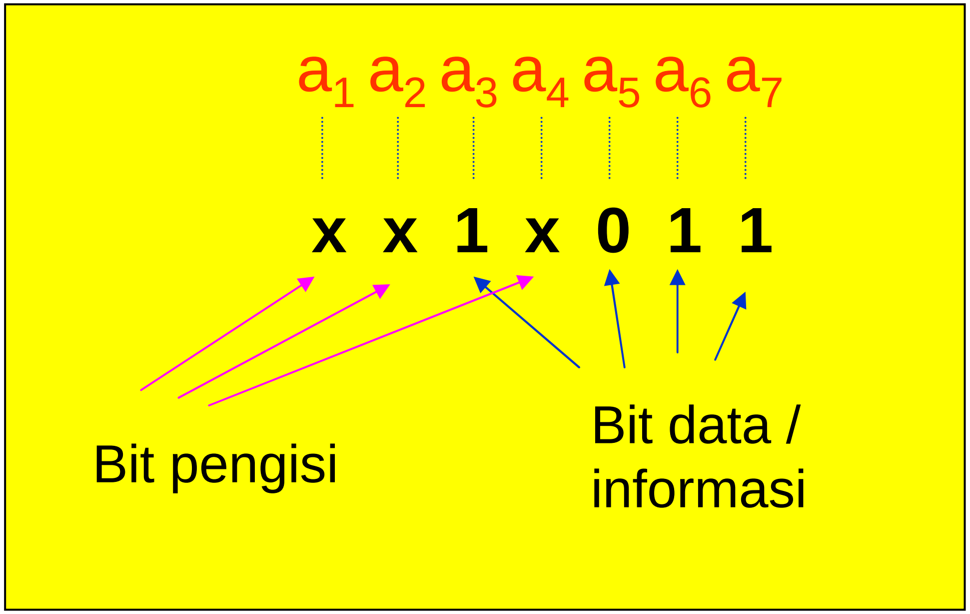
Kode ini dikenalkan oleh Richard Hamming (1950) sebagai kode tunggal pengoreksi kesalahan (*single error-correcting code*).

Bit penge-cek 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 penge-cek tambahan diberikan pada bit-bit informasi sebelum ditransmsisikan, sedangkan pada sisi penerima dilakukan penge-cek an dengan algoritma yang sama dengan pembangkitan bit penge-cek 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) → K = jumlah bit data - 1

Jumlah bit informasi =
2^n – **n** – 1
(**n** → jumlah bit cek)

Σ **Bit pengisi/cek**	Σ **bit informasi**
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,...
a_2= 3,6,7,10,11,14,15,...
a_4= 5,6,7,12,13,14,15,20,21,22,23,...
a_8= 9-15,24-31,40-47,...
a_{16}= 17-31,48-63,80-95,...
a_{32}= 33-63,96-127,160-191,...
dst.

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

Tabel Hamming untuk informasi 4-bit

Data/bit	a_1	a_2	a_3	a_4	a_5	a_6	a_7
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 penge-cek an 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 **e = 0**, maka seluruh data yang diterima adalah benar

Untuk informasi n-bit, cara penge-cek an adalah :

1. Tanda semua posisi bit yang merupakan pangkat dua sebagai bit penge-cek (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 menge-cek 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 : e1 = 0   e2 = 1   e4 = 0

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

Jawab :

e1 = a1 + a3 + a5 + a7 = 0 + 1 + 0 + 1 = 0 → benar

e2 = a2 + a3 + a6 + a7 = 0 + 1 + 1 + 1 = 1 → salah

e3 = a4 + a5 + a6 + a7 = 0 + 0 + 1 + 1 = 0 → benar

a1 = a3 + a5 + a7 = 1 + 0 + 1 = 0 → sama dengan yang dikirim

a2 = a3 + a6 + a7 = 1 + 1 + 1 = 1 → tidak sama dengan yang dikirim

a3 = a5 + a6 + a7 = 0 + 1 + 1 = 0 → sama dengan yang dikirim

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

Nilai data asli = a3a5a6a7 = 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	Σ_0	C_{out}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- Penjumlahan lanjut (selain kolom LSB)

$$\mathbf{A_i + B_i + C_{in} = \Sigma_i + C_{out}}$$

i = 2,3,4,..

$$\begin{array}{lll}
C_{in} & C_{in} & \\
 & A_1 & A_0 \\
+ & B_1 & B_0 \\
\hline
\Sigma_n & \Sigma_1 & \Sigma_0 \\
 & + & + \\
 & C_{out} & C_{out}
\end{array}$$

Tabel Kebenaran untuk
Penjumlahan 2 bit biner (lanjut)

A_1	B_1	C_{in}	Σ_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 :

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

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

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

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)

$$\mathbf{A_i - B_i - B_{in} = R_i + B_{out}}$$

i = 2,3,4,..

B_{in}	B_{in}	
	A_1	A_0
-	B_1	B_0
R_n	R_1	R_0
	+	+
	B_{out}	B_{out}

Tabel Kebenaran untuk
Pengurangann 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

Contoh :

1.

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

2.

$$\begin{array}{r} 18 \\ -12 \\ \hline 6 \end{array} \qquad\qquad \begin{array}{r} 10010 \\ -\ 01100 \\ \hline 00110 \end{array} = 6_{10}$$

3.

$$\begin{array}{r} 147 \\ -\ \ 75 \\ \hline 72 \end{array} \qquad\qquad \begin{array}{r} 10010011 \\ +\ 01001011 \\ \hline 10001000 \end{array} = 72_{10}$$

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	*Biner*	
13	1101	→ yang dikalikan
x 11	x 1011	→ pengali
13	1101	
13	1101	
143	0000	
	1101	
	$1000111 = 143_{10}$	→ 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.

Desimal	*Biner*
3	11 $= 3_{10}$ ← *hasil bagi*
3 / 9	011 / 1001 ← *yang dibagi*
- 9	- 011
0	0011
	- 011
	0

pembagi (011)

FUNGSI ARITMETIKA untuk sistim bilangan lain

1. PENJUMLAHAN

OCTAL

Contoh :

```
  73
+ 15
----
 121
```

HEXADECIMAL

Contoh :

```
  1D3
+  39
-----
  21C
```

BCD

Contoh :

```
  47      0100  0111
+ 15      0001  0101
----     -----------
  62      0101  1100
                0110
         -----------
          0110  0010
```

$\underbrace{0110}_{6}\quad\underbrace{0010}_{2}$

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 :

$$\begin{array}{rcc} 56 & 0101 & 0110 \\ -\ 34 & 0011 & 0100 \\ \hline 22 & \underbrace{0010}_{2} & \underbrace{0010}_{2} \end{array}$$

3. PERKALIAN

OCTAL

Contoh :

```
   14
x 13
-----
   44
 14
-----
 204
```

HEXADECIMAL

Contoh :

```
   1E2
x   25
------
   96A
 3C4
------
 45AA
```

4. PEMBAGIAN

OCTAL

Contoh :

```
     62
   ______
5 / 372
 - 36
   ____
    12
 -  12
   ____
     0
```

HEXADECIMAL

Contoh :

```
     64
   ______
F /  5DC
  - 5A
    ____
     3C
  -  3C
    ____
      0
```

BILANGAN BINER BERTANDA

+5 → 0 0101

-5 → 1 0101

Tanda + dinyatakan sebagai biner "0"

Tanda - dinyatakan sebagai biner "1"

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

+5	-5	+5	-5	+3	-3
+3	-3	-3	+3	-5	+5
+8	-8	+2	-2	-2	+2

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 : Biner "0" menjadi "1"
Biner "1" menjadi "0"

Contoh :

Carilah komplemen dari 10110

10110 → komplemen-nya : 01001

Carilah komplemen dari 110

110 → 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 : **010011**

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

   Jawab :

   35 = 010011

   1's compl : 101100

   + 1 : 1

   2's compl : **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)

ADDER
- HALF ADDER
- FULL ADDER

SUBTRACTOR
- HALF SUBTRACTOR
- FULL SUBTRACTOR

HALF ADDER

Merupakan implementasi operasi penjumlahan dasar dua bilangan

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

$$\begin{array}{rl} & \mathbf{A_0} \\ + & \mathbf{B_0} \\ \hline & \mathbf{\Sigma_o} \\ & \mathbf{+} \\ & \mathbf{C_{out}} \end{array}$$

Augend / yang dijumlahkan

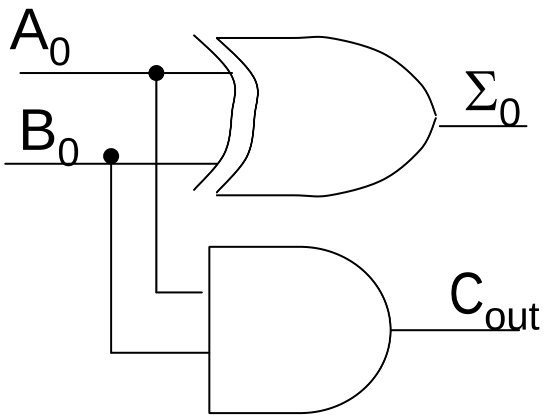
Addend / penjumlah

Sum / hasil

Carry

Tabel Kebenaran untuk Penjumlahan 2 bit biner (LSB)

A_0	B_0	Σ_0	C_{out}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Rangkaian Half Adder

Dari Tabel Kebenaran, dapatkan persamaan untuk Σ_0 dan C_{out} (menggunakan K-Map)

A_0 \ B_0	0	1
0	0	1
1	1	0

$$\Sigma_0 = \overline{A_0}.B_0 + A_0.\overline{B_0}$$
$$= A_0 \oplus B_0$$

A_0 \ B_0	0	1
0	0	0
1	0	1

$$C_{out} = A_0.B_0$$

FULL ADDER

Merupakan implementasi operasi penjumlahan dasar dua bilangan

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

$$\begin{array}{llll}
C_{in} & C_{in} & & \\
 & A_1 & A_0 & \\
+ & B_1 & B_0 & \\
\hline
\Sigma_n & \Sigma_1 & \Sigma_0 & \\
 & + & + & \\
 & C_{out} & C_{out} &
\end{array}$$

Tabel Kebenaran untuk
Penjumlahan 2 bit biner (lanjut)

A_1	B_1	C_{in}	Σ_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

$A_1 \backslash B_1C_{in}$	00	01	11	10
0	0	1	0	1
1	1	0	1	0

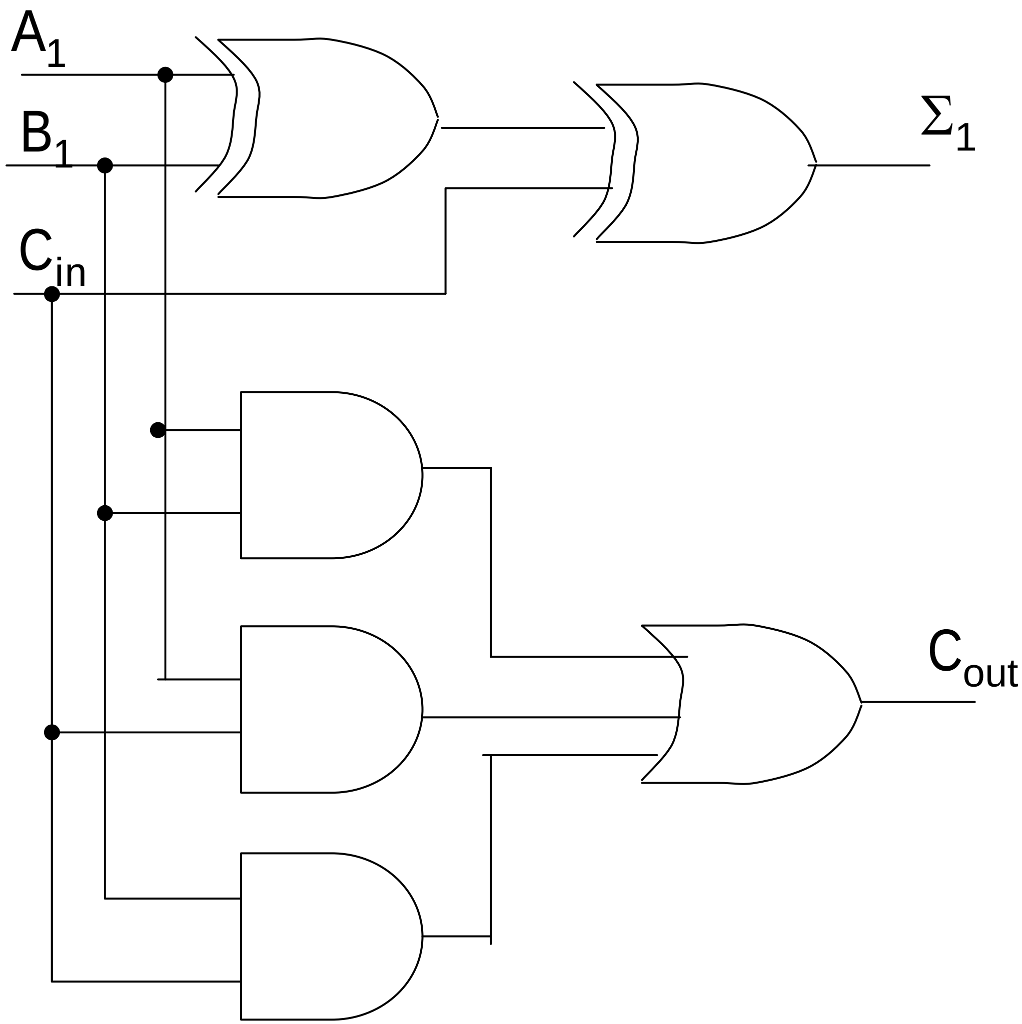
$$\Sigma_1 = \overline{A}_1\overline{B}_1C_{in} + A_1\overline{B}_1\overline{C}_{in} + \overline{A}_1B_1\overline{C}_{in} + A_1B_1C_{in}$$
$$= A_1 \oplus B_1 \oplus C_{in}$$

$A_1 \backslash B_1C_{in}$	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$C_{out} = A_1C_{in} + A_1B_1 + B_1C_{in}$$

Dari Tabel Kebenaran,
dapatkan persamaan untuk Σ_0
dan C_{out} (menggunakan K-Map)

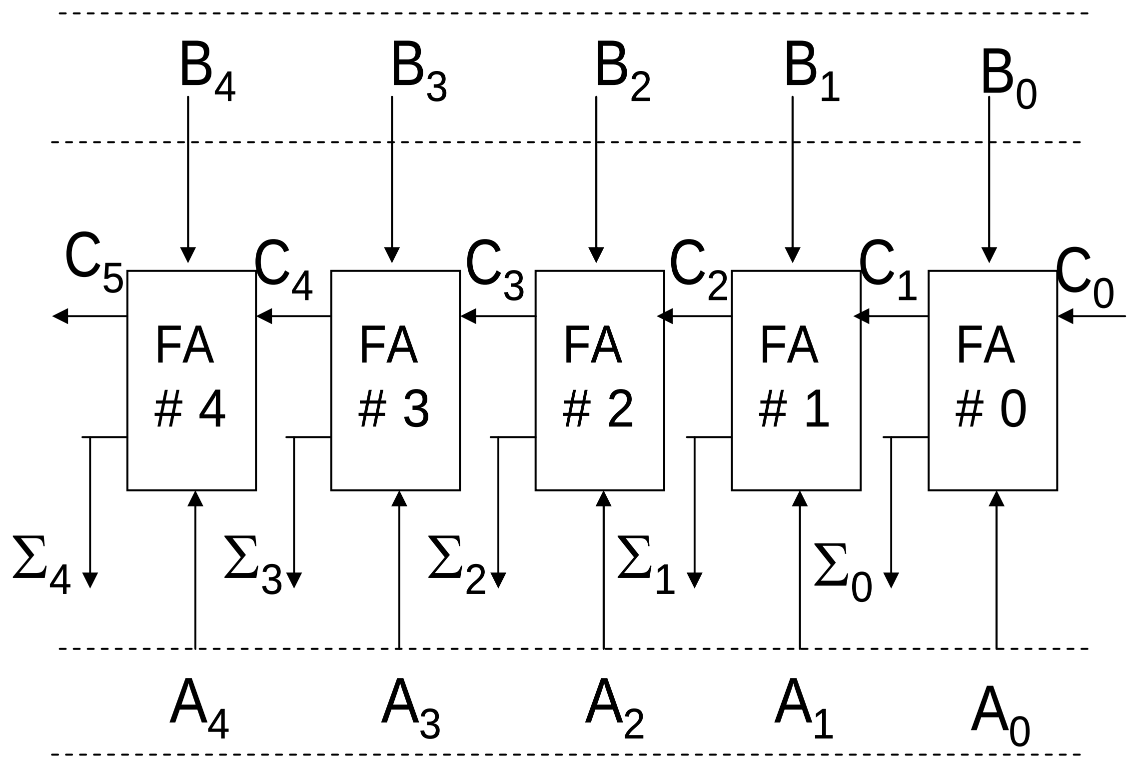
Rangkaian Full Adder



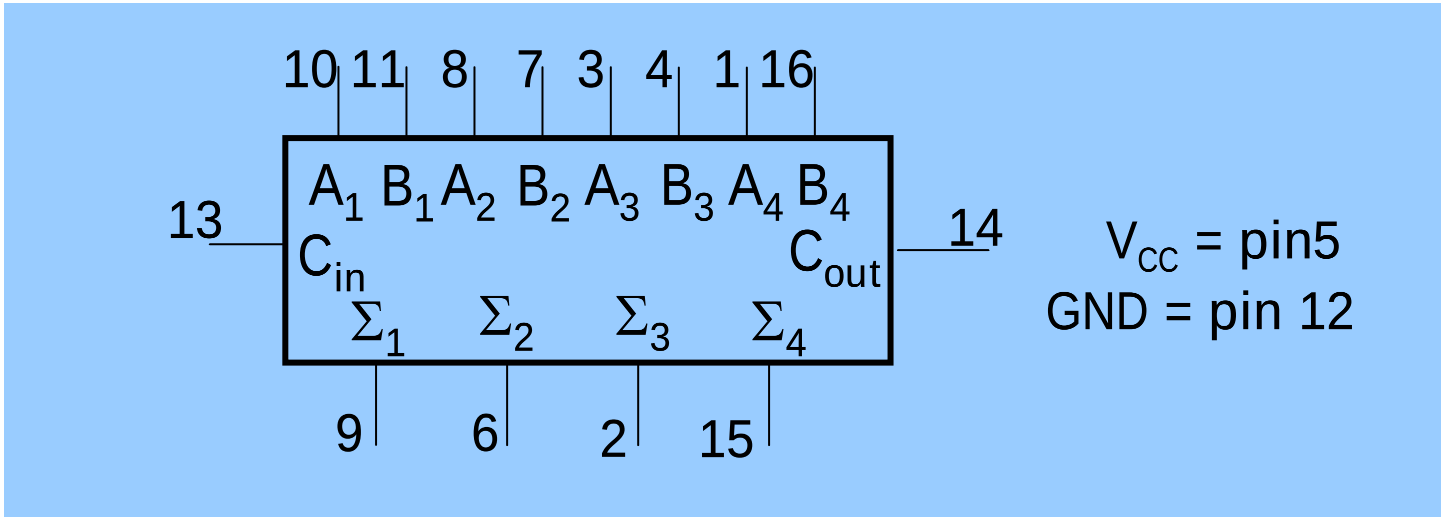
PARALLEL ADDER

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

A →	1	0	1	0	1
B →	0	0	1	1	1
+					
Σ →	1	1	1	0	0
$\mathbf{C_{out}}$ →	0	0	1	1	1



IC PARALLEL ADDER (74HC283)



A_1 – A_4 = Augend

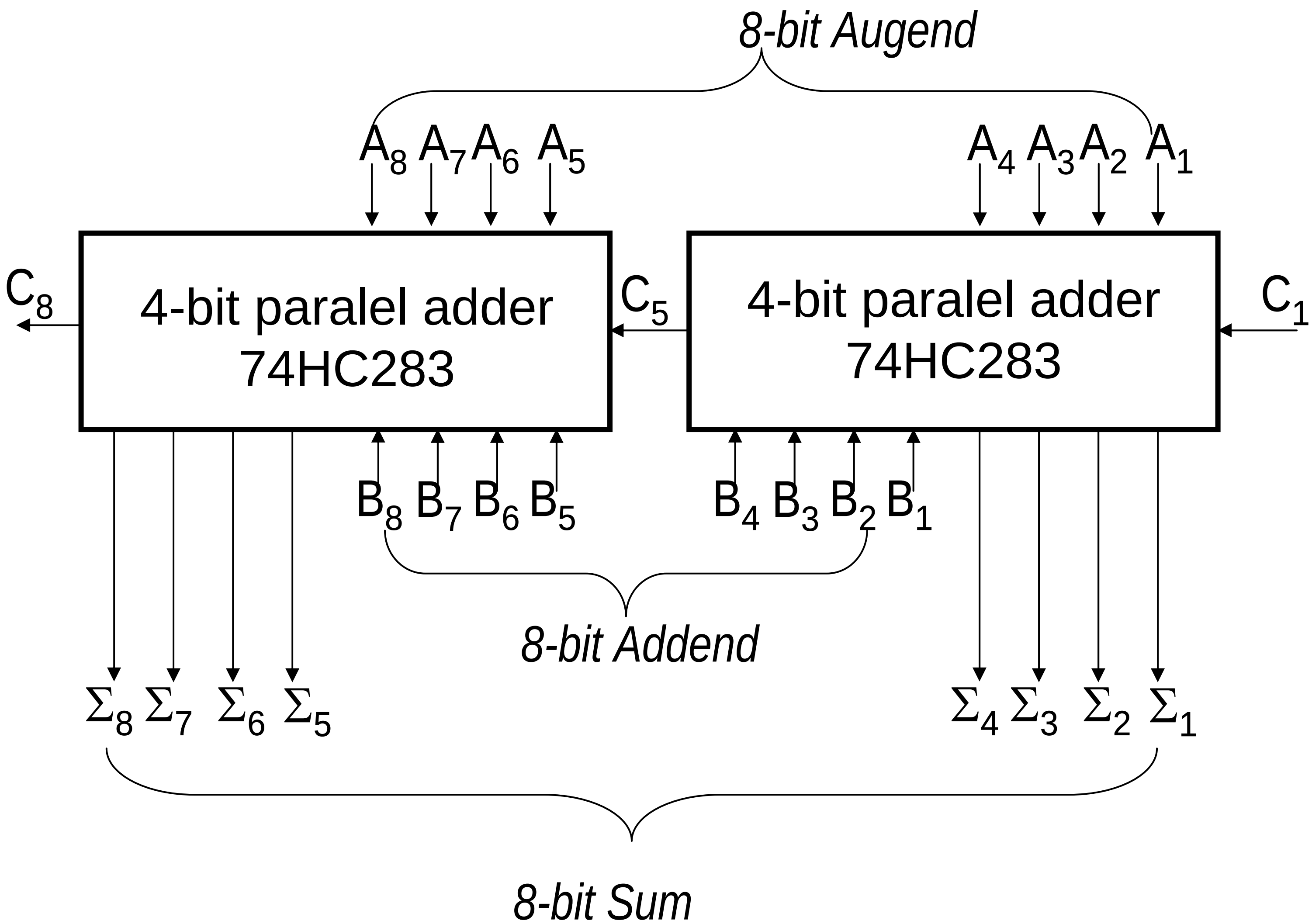
B_1 – B_4 = Addend

Σ_1 – Σ_4 = Sum

C_{in} = Carry In

C_{out} = Carry out

74HC283 sebagai Adder 8-bit



HALF SUBTRACTOR

Merupakan implementasi operasi pengurangan dasar dua bilangan

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

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

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

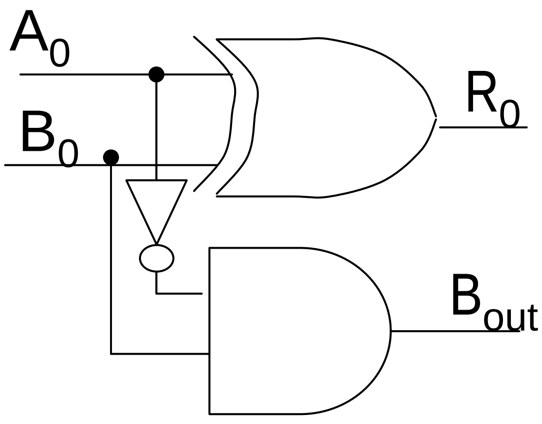
Dari Tabel Kebenaran,
dapatkan persamaan untuk R_0 dan B_{out} (menggunakan K-Map)

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

$$R_0 = \overline{A_0}.B_0 + A_0.\overline{B_0}$$
$$= A_0 \oplus B_0$$

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

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

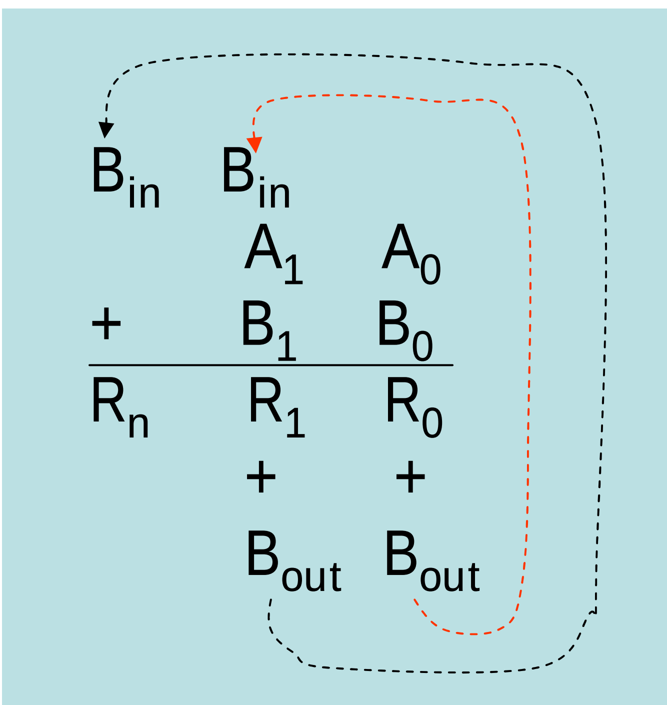


Rangkaian Half Subtractor

FULL SUBTRACTOR

Merupakan implementasi operasi pengurangan dasar dua bilangan

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



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 Σ_0 dan C_{out} (menggunakan K-Map)

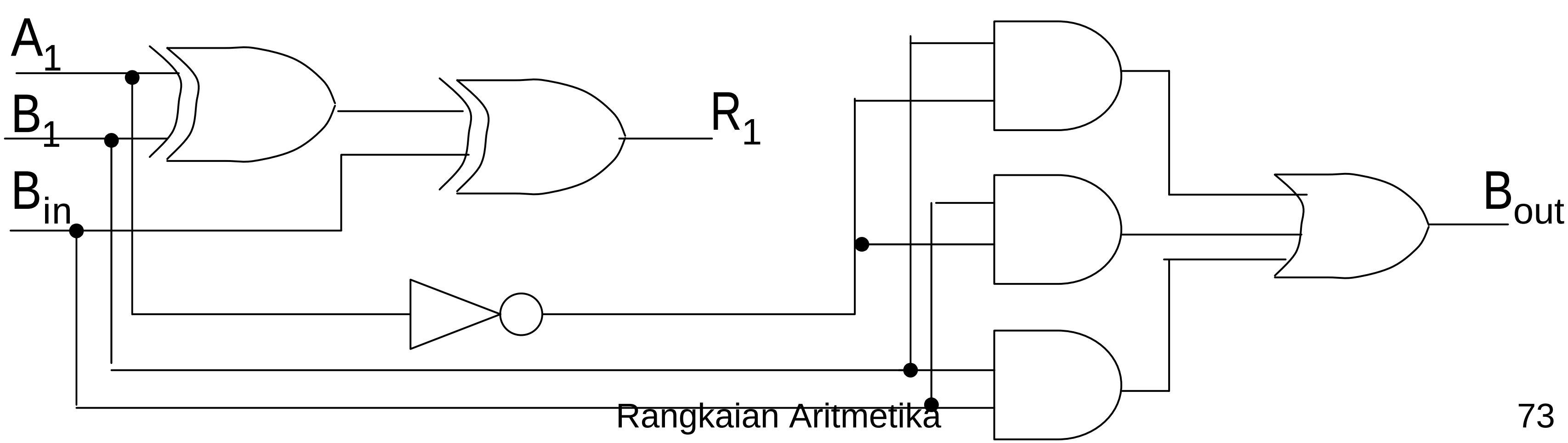
A_1 \ B_1B_{in}	00	01	11	10
0	0	1	0	1
1	1	0	1	0

A_1 \ B_1B_{in}	00	01	11	10
0	0	1	1	1
1	0	0	1	0

$$R_1 = \bar{A}_1\bar{B}_1B_{in} + A_1\bar{B}_1\bar{B}_{in} + \bar{A}_1B_1\bar{B}_{in} + A_1B_1B_{in}$$
$$= A_1 \oplus B_1 \oplus B_{in}$$

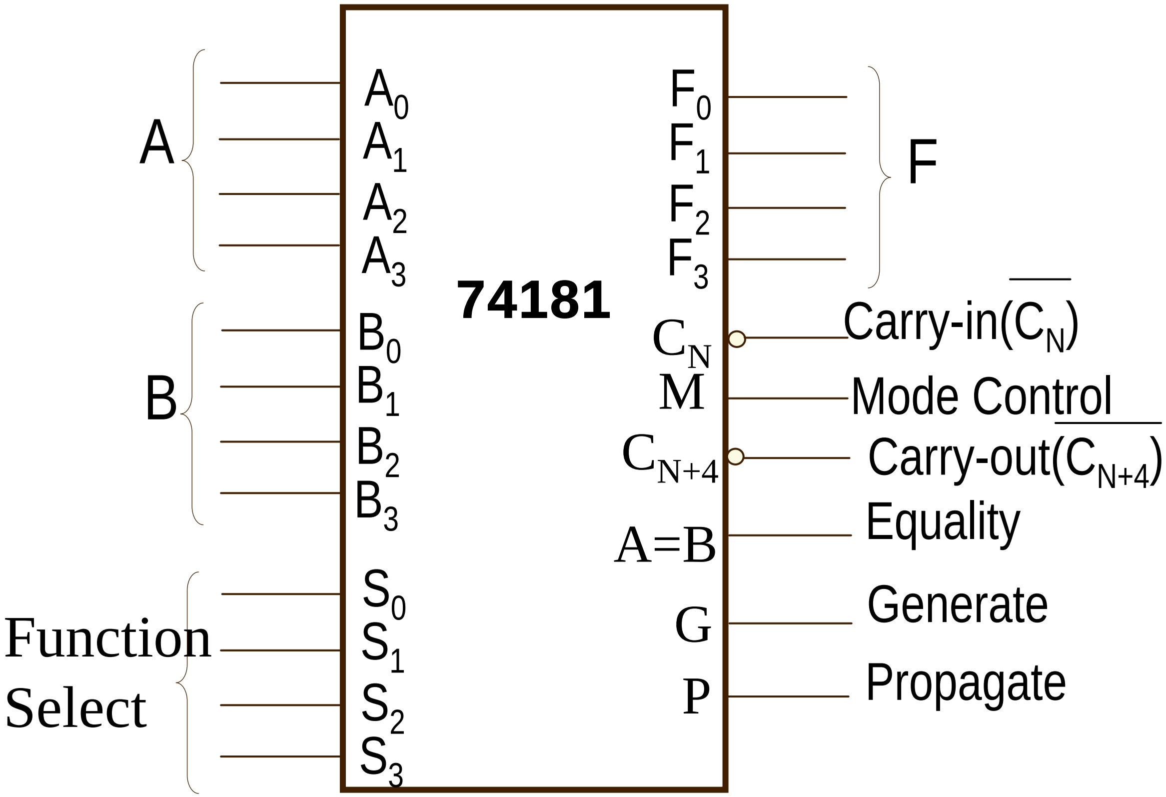
$$B_{out} = \bar{A}_1B_{in} + \bar{A}_1B_1 + B_1B_{in}$$

Rangkaian Full Subtractor



ARITHMETC/LOGIC UNIT (ALU)

Merupakan paket Large Scale Integrated-Circuit (LSI).
Mempunyai dua jenis operasi, yaitu : Aritmetika dan Logika

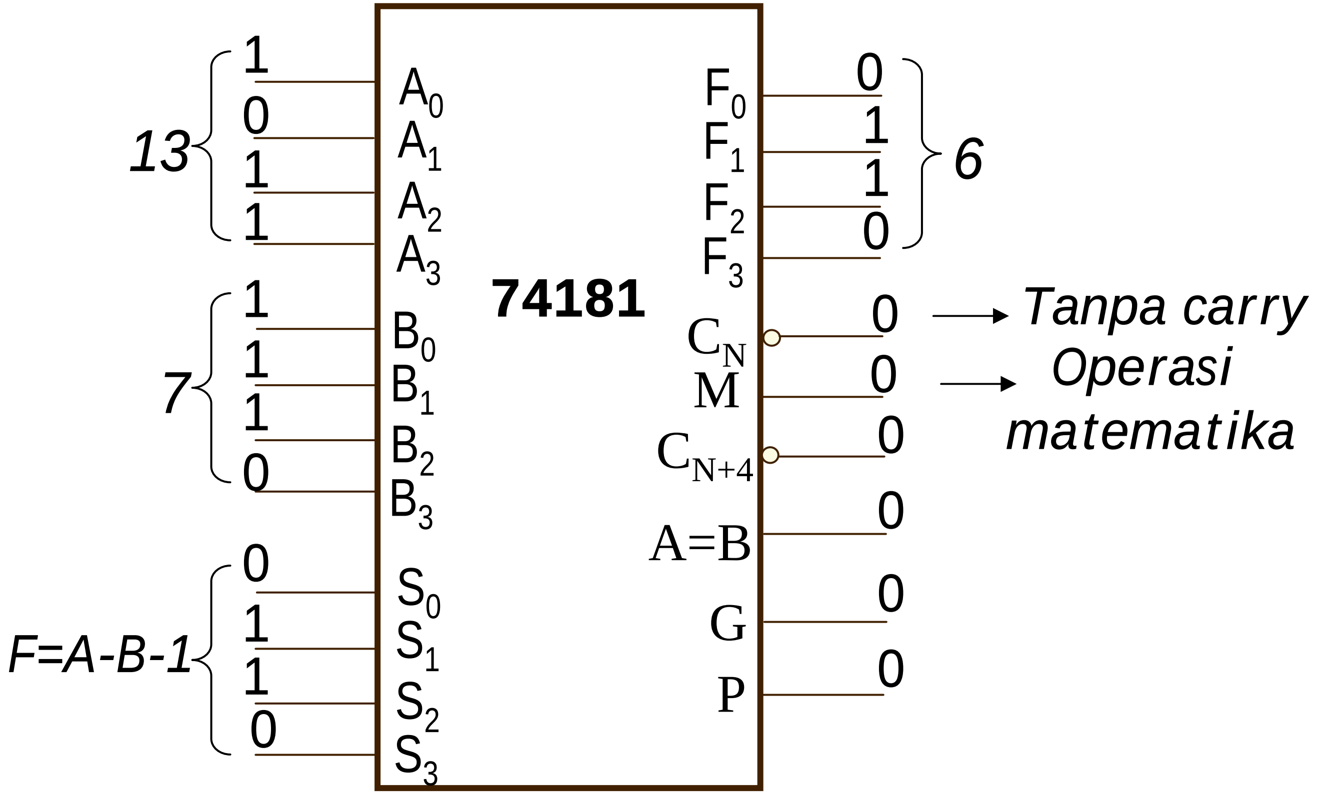


Tabel Fungsi ALU 74181

SELECTION				M=H Logic Function	M= L Aritmetic Operation
S_3	S_2	S_1	S_0		C_n=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=minus 1 (2's comp)
L	H	L	L	F=(AB)'	F=A plus AB'
L	H	L	H	F=B'	F=(A+B) plus AB'
L	H	H	L	$F=A \oplus B$	F=A minus B minus 1
L	H	H	H	F=AB'	F=AB' minus 1
H	L	L	L	F=A'+B	F=A plus AB
H	L	L	H	$F=(A \oplus B)'$	F=A plus B
H	L	H	L	F=B	F=(A+B') plus AB
H	L	H	H	F=AB	F=AB minus 1
H	H	L	L	F=1	F=A plus A*
H	H	L	H	F=A+B'	F=(A+B) plus A
H	H	H	L	F=A+B	F=(A+B') plus A
H	H	H	H	F=A	F=A minus 1

Contoh :

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



Soal Latihan

1. Konversikan sistim bilangan berikut :
   a. 27_{10} = …….$_{2}$
   b. 11010_{12} = ……$_{8}$
   c. 63_{8} = ……….$_{10}$
   d. $6FE_{16}$ = …….$_{2}$
   e. $1000\ 0101_{BCD}$ = …..$_{16}$
   f. 517_{8} = ………….$_{10}$
   g. $D3A_{16}$ = ……….$_{8}$
   h. 47_{8} = ………$_{BCD}$
   i. 756_{8} = …………$_{16}$
   j. $4C_{16}$ = ……….$_{2}$

2. Konversikan command berikut ini ke dalam kode ASCII :

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

3. Sebuah urutan data diterima : 1010101
Dengan : e1 = 1   e2 = 0   e4 = 0
Dengan kode Hamming, tentukan bit di posisi mana yang salah ? Berapa nilai data asli (sebelum ditambah bit penge-cek) ?

4. Selesaikan seluruh operasi aritmetika berikut menggunakan sistim bilangan :
1) biner   *2) oktal*   *3) hexadecimal*
a. 19 + 3 = ….   c. 22 – 8 = …..
b. 12 x 5 = ….   d. 48 : 12 = ….

5. Konversikan :
Desimal → 8-bit 2's complement
a) 12   b) -15   c) -112   d) 125
2's complement → desimal
a) 0101 1100   b) 1110 1111   c) 1000 0011

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

a) 5	b) 32	c) -28	d) -38
+7	-18	35	-46

7. Selesaikan operasi penjumlahan berikut menggunakan bentuk BCD

a) 8	b) 43	c) 7	d) 80
+3	+72	+38	+23

8. Ubahlah rangkaian Half Adder hanya menggunakan gerbang NOR saja

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