

2.1.2. Pengurangan Bilangan Biner

Dasar pengurangan untuk masing - masing digit bilangan binary adalah :

$$\begin{aligned}
 0 - 0 &= 0 \\
 1 - 0 &= 1 & 0 - 1 &= 1 \Rightarrow \text{Dengan borrow of 1 (pinjam)} \\
 1 - 1 &= 0 & & \text{digit 1 dari posisi sebelah kirinya} \\
 0 - 1 &= 1 & &
 \end{aligned}$$

Beberapa contoh pengurangan binary :

- Tanpa terjadinya peminjaman digit

Desimal	Binary
27	1 1 0 1 1
9 _	1 0 0 1 _
18	1 0 0 1 0

- Terjadi peminjaman sebuah bilangan di kolom sebelah kirinya.

Desimal	Binary
29	1 1 1 0 1
11 _	1 0 1 1 _
18	1 0 0 1 0

Caranya :

$$\begin{array}{rcl}
 1 - 1 & = & 0 \\
 0 - 1 & = & 1 \quad , \text{ dengan borrow of 1} \\
 1 - 0 - 1 & = & 0 \\
 1 - 1 & = & 0 \\
 1 - 0 & = & 1
 \end{array}$$

- Tidak terjadi peminjaman sebuah bilangan di kolom sebelahnya karena yang akan dipinjam bernilai 0, sehingga harus meminjam di kolom sebelahnya lagi.

Desimal	Binary
25	1 1 0 0 1
19 _	1 0 0 1 1 _
6	0 0 1 1 0

$$\begin{array}{r}
 \text{Caranya : } 1 - 1 = 0 \\
 0 - 1 = 1 \quad , \text{ dengan borrow of } 1 \\
 0 - 0 - 1 = 1 \quad , \text{ dengan borrow of } 1 \\
 1 - 0 - 1 = 0 \\
 1 - 1 = 0
 \end{array}$$

2.1.3. Bilangan Komplemen

2.1.3.1. Pengurangan Bilangan Dengan Komplemen

Metode pengurangan binary biasa dilakukan oleh manusia, untuk komputer biasanya menggunakan metode komplemen (*complement*) yaitu :

- Komplemen basis - 1 (*Radix minus one complement*)
- Komplemen basis (*Radix*)

Komplemen pada dasarnya merubah bentuk pengurangan menjadi bentuk pertambahan.

Dalam sistem desimal, ada 2 macam komplemen yaitu :

- Komplemen 9 (*9's complement*)
- Komplemen 10 (*10's complement*)

Dalam sistem biner :

- Komplemen 1 (*1's complement*)
- Komplemen 2 (*2's complement*)

- Komplemen 9 (*9's Complement*)

Komplemen 9 dari suatu bilangan desimal dilakukan dengan cara mengurangkan angka 9 untuk masing - masing digit dalam bilangan pengurang.

Contoh :

Komplemen 9 dari 24 adalah 75, yaitu : $99 - 24 = 75$

Komplemen 9 dari 321 adalah 678, yaitu : $999 - 321 = 678$

Pengurangan biasa Komplemen 9 Dalam komplemen 9 digit yang paling

$$\begin{array}{r}
 859 \\
 \underline{523} - \\
 336
 \end{array}
 \Rightarrow
 \begin{array}{r}
 859 \\
 \underline{476} + \\
 1335 \\
 \underline{\quad} 1 \\
 \underline{\quad} + \\
 336
 \end{array}
 \quad \text{kiri ditambahkan pada digit paling kanan.}$$

- **Komplemen 10 (10's Complement)**

Komplemen 10 dari suatu bilangan dilakukan dengan cara, hasil komplemen 9 ditambah 1 (cari komplemen 9 lalu ditambah 1).

Contoh : Komplemen 10 dari 24 \Rightarrow 9's = 75
 $10's = 75 + 1$
 $= 76$

Contoh pengurangan :

$\begin{array}{r} 9's : 859 \\ \underline{523} - \\ 336 \end{array}$	$10's : 859 \Rightarrow 9's : 476 + 1$ $\begin{array}{r} \underline{477} + \\ 1336 \\ \underline{\quad} \text{ diabaikan / dibuang} \end{array}$
--	--

- **Komplemen 1 (1's Complement)**

Komplemen 1 dari suatu bilangan biner dilakukan dengan cara mengurangkan semua digit dengan nilai bit 1/ merubah bit '0' menjadi '1' atau bit '1' menjadi '0'.

Contoh :

1's dari 10110 = 01001

$25 \Rightarrow$	$11001 \Rightarrow$	$1's : 11001$
$\underline{22} -$	$\underline{10110} -$	$\underline{01001} +$
3	00011	$\underline{100010}$
		$\underline{\quad} 1 +$
		00011

Dalam komplemen 1, digit 1 paling ujung kiri ditambahkan pada digit paling kanan.

- **Komplemen 2 (2's Complement)**

Komplemen 2 dari suatu bilangan biner dilakukan dengan cara, hasil komplemen 1 ditambah 1.

Komplemen 2 dari bilangan biner 10110 \Rightarrow

10110	\Rightarrow	10110
		$1's : \underline{01001} + 1$
		$2's : 01010$

$$\begin{array}{r}
 25 \Rightarrow 11001 \Rightarrow 2's : 11001 \\
 \underline{22} \quad \underline{10110} - \quad \underline{01010} + \\
 3 \quad \quad 00011 \quad \quad \quad 100011 \\
 \quad \quad \quad \quad \quad \quad \quad 00011 \\
 \quad \quad \quad \quad \quad \quad \quad \text{dibuang / diabaikan}
 \end{array}$$

- **Komplemen 7 (7's Complement)**

Komplemen 7 dari suatu bilangan oktal dilakukan dengan cara, mengurangi angka 7 untuk masing - masing digit dalam bilangan pengurangan.

- **Komplemen 8 (8's Complement)**

Komplemen 8 dari suatu bilangan dilakukan dengan cara, hasil komplemen 7 ditambah 1 (cari komplemen 7 dulu lalu ditambah 1).

- **Komplemen 15 (15's Complement)**

Komplemen 15 dari suatu bilangan hexadesimal dilakukan dengan cara, mengurangi angka 15 untuk masing - masing digit dalam bilangan pengurangan.

- **Komplemen 16 (16's Complement)**

Komplemen 16 dari suatu bilangan dilakukan dengan cara, hasil komplemen 15 ditambah 1 (cari komplemen 15 dulu lalu ditambah 1).

2.1.4. Perkalian Bilangan Biner

- Dilakukan seperti perkalian pada bilangan desimal
- Perkalian binari dilakukan dengan cara operasi pertambahan yang dilakukan secara berulang - ulang.
- Dasar perkalian untuk masing - masing digit bilangan biner

Bilangan biner :

$$\begin{array}{l}
 0 \times 0 = 0 \\
 0 \times 1 = 0 \\
 1 \times 0 = 0 \\
 1 \times 1 = 1
 \end{array}$$

Contoh :

$$\begin{array}{r}
 14 \\
 \underline{12 \times} \\
 28 \\
 14 + \\
 \hline
 168
 \end{array}
 \qquad
 \begin{array}{r}
 1110 \\
 \underline{1100 \times} \\
 0000 \\
 1110 \\
 1110 + \\
 \hline
 10101000
 \end{array}$$

Jika pengali yang berupa digit biner bernilai 1, hasilnya berupa bilangan biner yang dikali (disalin saja), jika pengali = 0, maka hasilnya = 0

2.1.5. Pembagian Bilangan Biner

- Dilakukan seperti pembagian pada bilangan desimal
- Pembagian dengan digit biner 0 tidak mempunyai arti
- Pembagian Binary yang dilakukan dengan cara operasi pengurangan yang dilakukan secara berulang - ulang.
- Dasar pembagian untuk masing - masing digit bilangan biner

Bilangan biner : $0 : 1 = 0$
 $1 : 1 = 1$

Contoh :

$$\begin{array}{r}
 \underline{25} \\
 5 \overline{)125} \\
 \underline{10} - \\
 25 \\
 \underline{25} - \\
 0
 \end{array}
 \qquad
 \begin{array}{r}
 \underline{11001} \\
 101 \overline{)1111101} \\
 \underline{101} \\
 101 \\
 \underline{101} \\
 101 \\
 \underline{101} \\
 0
 \end{array}$$

LATIHAN-1

1. Tuliskan jawaban dari hasil operasi aritmatika biner berikut ini :

1. $1010 + 101 = \dots$
2. $1111 + 1 = \dots$
3. $1011 - 1010 = \dots$
4. $10000 - 1 = \dots$
5. $1010 \times 1010 = \dots$

2. Carilah komplementnya : $(250)_{10}$

1's = ...

2's = ...

8's = ...

9's = ...

16's = ...

2. Gunakan 1's dan 2's untuk menghitung operasi pengurangan di bawah ini:

a) $10011101 - 11100110$

b) $01101110 - 11001111$

3. Hitung :

a) $1011 + 1110 + 1101$

b) $11011 + 10110$

4. Carilah 7's dan 8's dari :

a) 176

b) 325

c) 6072

5. Hitunglah pengurangan berikut dengan 7's dan 8's :

a) $435 - 176$

b) $513 - 325$

c) $7151 - 6072$

6. Carilah 15's dan 16's dari :

a) 1A6

b) AB3

c) 402D

2.2. Bilangan Biner Bertanda

- Bilangan biner positif mempunyai nilai antara $0000\ 0000_2 = 00_{10}$ dan $1111\ 1111_2 = 255_{10}$.
- Untuk membedakan bilangan positif dengan negatif sebuah bilangan desimal diberi tanda '-' disebelah kiri bilangan. Misal : -25_{10}
- Dalam bilangan biner tanda bilangan yaitu '-' disandikan dengan cara tertentu yang mudah dikenal dalam sistem digital. Untuk menyatakan bilangan negatif pada bilangan biner, bilangan yang dikenal dengan bit tanda bilangan (sign bit) ditambahkan di sebelah kiri MSB.
- Bilangan biner yang ditulis dengan cara di atas, menunjukkan tanda dan besarnya bilangan.
- Jika bit tanda (sign bit) = 0 \Rightarrow menunjukkan bilangan positif.
- Jika bit tanda (sign bit) = 1 \Rightarrow menunjukkan bilangan negatif.
- Pada bilangan biner bertanda yang terdiri dari 8 bit, bit yang paling kiri menunjukkan tanda, dan 7 bit berikutnya menunjukkan besarnya bilangan.

Contoh :

No. Bit	7	6	5	4	3	2	1	0
Bit	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	(tanda)	64	32	16	8	4	2	1

Maka,

$$[0]1100111 = + (64+32+4+2+1) \quad [0]1111111 = + (64+32+16+8+4+2+1)$$

$$= + 103 \quad = + 127_{10}$$

$$[1]1010101 = - (64+16+4+1) \quad [1]1111111 = - (64+32+16+8+4+2+1)$$

$$= - 85_{10} \quad = - 127_{10}$$

Karena hanya 7 bit yang menunjukkan besarnya bilangan, maka bilangan terkecil dan terbesar yang ditunjukkan bilangan biner bertanda terdiri dari 8 bit adalah :

$$[0] 1111111 = + 127_{10}$$

$$[1] 1111111 = - 127_{10}$$

Bilangan biner tak bertanda yang terdiri dari n bit mempunyai nilai maksimum

$$M = 2^n - 1$$

Bilangan biner bertanda yang terdiri dari n bit mempunyai nilai maksimum

$$M = 2^{(n-1)} - 1$$

Jadi untuk register 8 bit di dalam mikroprocessor yang menggunakan sistem bilangan bertanda, nilai terbesar yang bisa disimpan dalam register tersebut adalah :

$$M = 2^{(n-1)} - 1$$

$$= 2^{(8-1)} - 1$$

$$= 2^7 - 1$$

$$= 128 - 1$$

$$= 127_{10}$$

Jadi jangkauannya = - 127₁₀ sampai + 127₁₀

2.3. Pengkodean

- Kode adalah karakter - karakter khusus (numerik/alphabet) yang dipakai sebagai simbol untuk karakter lain.
- Dalam komponen digital karakter khusus tersebut adalah 1 dan 0.
- Manfaat kode di dalam komputer adalah :
 - Untuk mempermudah operasi aritmatika.
 - Untuk alasan - alasan efisiensi.
 - Dipakai untuk mendeteksi bahkan mengoreksi data - data yang ditransmisikan.
 - Untuk alasan keamanan.
- Kode terbagi dalam 3 kategory besar :
 - Weighted Code (Kode dengan pembobot), tiap posisi memiliki kode pembobot. Contoh : BCD dan Excess-3
 - Un-weighted Code, tiap posisi tidak ada pembobot, tergantung bit berdekatan, Contoh : Grey Code
 - Sequential Code, kode selanjutnya = lebih besar 1 binary dari sebelumnya. Contoh : BCD dan Excess-3
- Binary Coded Decimal (BCD). Bilangan desimal yang dikodekan dalam bentuk bilangan biner.
 - Keluarga 421.

Desimal	8421	5421	7421	$\overline{7421}$	2^*421
0	0000	0000	0000	0000	0000
1	0001	0001	0001	0111	0001
2	0010	0010	0010	0110	0010
3	0011	0011	0011	0101	0011
4	0100	0100	0100	0100	0100
5	0101	1000	0101	1010	1011
6	0110	1001	0110	1001	1100
7	0111	1010	1000	1000	1101
8	1000	1011	1001	1111	1110
9	1001	1100	1010	1110	1111
10	0001 0000	0001 0000	0001 0000	0111 0000	0001 0000
31	0011 0001	0011 0001	0011 0001	0101 0111	0011 0001

– Selain keluarga 421. 74210-1 (???)

Desimal	6311	Exess-3	2 out of 5
0	0000	0011	00011
1	0001	0100	00101
2	0011	0101	00110
3	0100	0110	01001
4	0101	0111	01010
5	0111	1000	01100
6	1000	1001	10001
7	1001	1010	10010
8	1011	1011	10100
9	1100	1100	11000
10	0001 0000	0100 0011	00101 00011

Contoh :

$$\begin{aligned}
 (153)_{10} &= (0001\ 1000\ 0011)_{5421} \\
 (8391)_{10} &= (1110\ 0011\ 1111\ 0001)_{2^*421} \\
 (1011\ 0100\ 1111)_{2^*421} &= (5\ 4\ 9)_{10} \\
 (4213)_{10} &= (0111\ 0101\ 0100\ 0110)_{\text{Excess-3}}
 \end{aligned}$$

Penjumlahan dalam excess-3 akan menjadi excess-6 untuk menjadikannya excess-3, harus dikurangi 3.

$$\begin{array}{r}
 0101\ (2) \\
 + 1000\ (5) \\
 \hline
 1101\ (10) \quad \rightarrow \text{Excess-6} \\
 - 0011\ (3) \\
 \hline
 1010\ (7) \quad \rightarrow \text{Excess-3}
 \end{array}$$

$$\begin{array}{r}
 1111\ 1 \\
 0101\ 1100\ (29) \\
 + 0110\ 1100\ (39) \\
 \hline
 1100\ 1000\ (95) \\
 - 0011\ +0011 \\
 \hline
 1001\ 1011\ (68)
 \end{array}$$

2.3.1. Kode Gray

Konversi biner ke kode gray :

- MSB biner = MSB Gray
- Selanjutnya bilangan biner menentukan nilai dari gray.
 - Jika bit biner sebelum = bit biner sesudah, maka gray bernilai “0”
 - Jika bit biner sebelum <> bit biner sesudah, maka gray bernilai “1”

Konversi dari kode gray ke biner

- MSB gray = MSB biner
- Selanjutnya bit gray, menjadi bit kontrol.
 - Jika bit gray 1 maka akan mengubah digit biner sebelumnya.
 - Jika bit gray 0 maka akan mengulang digit biner sebelumnya.

Contoh :

(11001010010) biner = (10101111011) gray

(10101111011) gray = (11001010010) biner

2.3.2. Kode ASCII

Pemrosesan data oleh komputer terdiri dari angka, huruf, dan simbol. Agar data tersebut dapat digunakan oleh semua aplikasi komputer, perlu adanya standarisasi bentuk yang harus dapat dikonversikan ke dalam bilangan biner. ASCII code (American Standard Code for Information Interchange) diciptakan untuk menyelesaikan permasalahan tersebut. ASCII terdiri dari 7 bit kode sehingga akan menghasilkan 2^7 (128) kombinasi yang akan mewakili angka, huruf, dan simbol.

ASCII Code								ASCII Code								ASCII Code							
6	5	4	3	2	1	0		6	5	4	3	2	1	0		6	5	4	3	2	1	0	
s	0	1	0	0	0	0	0	@	1	0	0	0	0	0	0	`	1	1	0	0	0	0	0
!	0	1	0	0	0	0	1	A	1	0	0	0	0	0	1	a	1	1	0	0	0	0	1
“	0	1	0	0	0	1	0	B	1	0	0	0	0	1	0	b	1	1	0	0	0	1	0
#	0	1	0	0	0	1	1	C	1	0	0	0	0	1	1	c	1	1	0	0	0	1	1
\$	0	1	0	0	1	0	0	D	1	0	0	0	1	0	0	d	1	1	0	0	1	0	0
%	0	1	0	0	1	0	1	E	1	0	0	0	1	0	1	e	1	1	0	0	1	0	1
&	0	1	0	0	1	1	0	F	1	0	0	0	1	1	0	f	1	1	0	0	1	1	0
'	0	1	0	0	1	1	1	G	1	0	0	0	1	1	1	g	1	1	0	0	1	1	1
(0	1	0	1	0	0	0	H	1	0	0	1	0	0	0	h	1	1	0	1	0	0	0
)	0	1	0	1	0	0	1	I	1	0	0	1	0	0	1	i	1	1	0	1	0	0	1
*	0	1	0	1	0	1	0	J	1	0	0	1	0	1	0	j	1	1	0	1	0	1	0
+	0	1	0	1	0	1	1	K	1	0	0	1	0	1	1	k	1	1	0	1	0	1	1
,	0	1	0	1	1	0	0	L	1	0	0	1	1	0	0	l	1	1	0	1	1	0	0
-	0	1	0	1	1	0	1	M	1	0	0	1	1	0	1	m	1	1	0	1	1	0	1
.	0	1	0	1	1	1	0	N	1	0	0	1	1	1	0	n	1	1	0	1	1	1	0
/	0	1	0	1	1	1	1	O	1	0	0	1	1	1	1	o	1	1	0	1	1	1	1
0	0	1	1	0	0	0	0	P	1	0	1	0	0	0	0	p	1	1	1	0	0	0	0
1	0	1	1	0	0	0	1	Q	1	0	1	0	0	0	1	q	1	1	1	0	0	0	1
2	0	1	1	0	0	1	0	R	1	0	1	0	0	1	0	r	1	1	1	0	0	1	0
3	0	1	1	0	0	1	1	S	1	0	1	0	0	1	1	s	1	1	1	0	0	1	1
4	0	1	1	0	1	0	0	T	1	0	1	0	1	0	0	t	1	1	1	0	1	0	0
5	0	1	1	0	1	0	1	U	1	0	1	0	1	0	1	u	1	1	1	0	1	0	1
6	0	1	1	0	1	1	0	V	1	0	1	0	1	1	0	v	1	1	1	0	1	1	0
7	0	1	1	0	1	1	1	W	1	0	1	0	1	1	1	w	1	1	1	0	1	1	1
8	0	1	1	1	0	0	0	X	1	0	1	1	0	0	0	x	1	1	1	1	0	0	0
9	0	1	1	1	0	0	1	Y	1	0	1	1	0	0	1	y	1	1	1	1	0	0	1
:	0	1	1	1	0	1	0	Z	1	0	1	1	0	1	0	z	1	1	1	1	0	1	0
;	0	1	1	1	0	1	1	[1	0	1	1	0	1	1	{	1	1	1	1	0	1	1
<	0	1	1	1	1	0	0	\	1	0	1	1	1	0	0		1	1	1	1	1	0	0
=	0	1	1	1	1	0	1]	1	0	1	1	1	0	1	}	1	1	1	1	1	0	1
>	0	1	1	1	1	1	0	^	1	0	1	1	1	1	0	~	1	1	1	1	1	1	0
?	0	1	1	1	1	1	1	_	1	0	1	1	1	1	1	d	1	1	1	1	1	1	1

Berdasarkan tabel ASCII tersebut, suatu karakter akan dapat langsung diubah ke dalam bentuk binernya.

Contoh :

Karakter “ S t a r t “ dapat dikodekan dalam ASCII menjadi :

1010011 1110100 1100001 1110010 1110100
 S t a r t

2.3.3. Paritas Dalam Kode

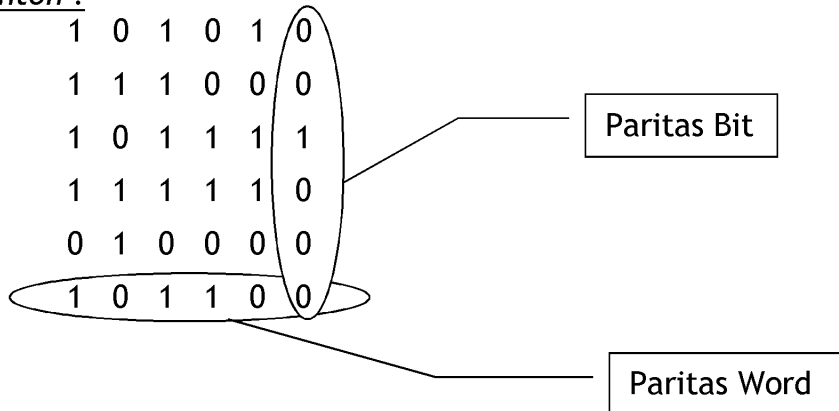
Sifat Paritas :

1. Paritas Genap
Jumlah angka 1 dalam word + paritas = genap
2. Paritas Ganjil
Jumlah angka 1 dalam word + paritas = ganjil

Jenis Paritas :

1. Paritas Bit
Menambahkan 1 bit
2. Paritas Word
Menambahkan 1 word / kumpulan bit

Contoh :



LATIHAN - 3

☞ Kodekan bilangan berikut ini :

1. $(193)_{10} = (\dots)_{7421}$
2. $(456)_{10} = (\dots)_{2^*421}$

3. (100101001110) biner = (\dots) gray
 (110111101001) gray = (\dots) biner

4. STMIK - TI - PelNus = (\dots) ascii

☞ Data-data di samping merupakan pesan yang akan dikirim. Pada rangkaian data tersebut sudah termasuk paritas bit dan paritas word. Dengan menggunakan paritas ganjil, periksalah apakah ada kesalahan atau tidak pada data tersebut. Jika ada tunjukkan data mana yang salah dan bagaimana data yang seharusnya dikirim.

1001001
0100011
1110110
1001111
0101010
1110011
0100110
0001000
1011010