COMPUTER ARCHITECTURE & NETWORKS

CHAPTER 2:

NUMBER SYSTEMS

LEARNING OUTCOMES

This chapter discusses about number systems.

To understand the conversion of number systems in computational methods.

CONTENT

- Introduction to number systems:
  - Binary > 0 - 1
  - Octal > 0 - 7
  - Hexadecimal > 0 - F
  - Decimal > 0 - 9

System	Base	Symbols
Decimal	10	0, 1, … 9
Binary	2	0, 1
Octal	8	0, 1, 2, …7
Hexa-decimal	16	0, 1, … 9, A, B, … F

NUMBER SYSTEMS

- A set of values used to represent different quantities is known as Number System.
- A number system can be used to represent the number of students in a class or number of presenters in a session and etc.
- The digital computer represents all kind of data and information in binary numbers.

NUMBER SYSTEMS

- It includes audio, graphics, video, text and numbers.
- The total number of digits used in a number system is called its base or radix.

TYPE OF NUMBER SYSTEM

- Decimal Number System
- Binary Number System
- Octal Number System
- Hexadecimal Number System

DECIMAL NUMBER SYSTEM

- The number system that we use in our day-to-day life is the decimal number system.
- Decimal number system has Base 10 as it uses 10 digits.
- 10 symbols : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Weightage of each symbol depends on the symbol position
- Example :

  $853_{10} = (8\times10^2) + (5\times10^1) + (3\times10^0)$

  The successive positions to the left of decimal point represent units, tens, hundreds, thousands, and so on.

DECIMAL NUMBER SYSTEM

The position of each digit in a weighted number system is assigned a weight based on the **base** or **radix** of the system. The radix of decimal numbers is ten, because only ten symbols (0 through 9) are used to represent any number.

The column weights of decimal numbers are powers of ten that increase from right to left beginning with $10^0 = 1$:

$$\ldots 10^5\ 10^4\ 10^3\ 10^2\ 10^1\ 10^0.$$

For fractional decimal numbers, the column weights are negative powers of ten that decrease from left to right:

$$10^2\ 10^1\ 10^0.\ 10^{-1}\ 10^{-2}\ 10^{-3}\ 10^{-4}\ \ldots$$

DECIMAL NUMBERS

Consider the number 1984

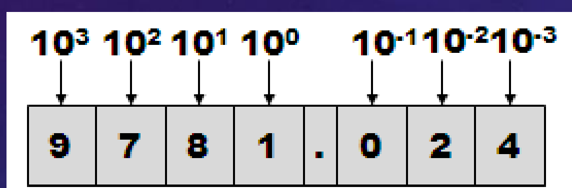
Position	4	3	2	1
Weight	**1000**	**100**	**10**	**1**
Decimal number	1	9	8	4

1984 = 1 x 1000 + 9 x 100 + 8 x 10 + 4 x 1

DECIMAL NUMBER SYSTEM

- It is the most widely used numeral system.

- Position of each digit in a decimal number system carries a weight.
- For example: 9781.024 is a decimal number
- Each digit 9,7,8,1,0,2,4 carries different weight.



- $9781.024_{10} = (9 \times 10^3) + (7 \times 10^2) + (8 \times 10^1) + (1 \times 10^0) + (0 \times 10^{-1}) + (2 \times 10^{-2}) + (4 \times 10^{-3})$

Test Yourself:

Express the decimal number 2745.214 as a sum of the values of each digit.

DECIMAL SYSTEM

weights	10^3	10^2	10^1	10^0		10^{-1}	10^{-2}	10^{-3}
	↓	↓	↓	↓		↓	↓	↓
	2	**7**	**4**	**5**	**.**	**2**	**1**	**4**

2745.214_{10}

$= (2x10^3)+(7x10^2)+(4x10^1)+(5x10^0)+(2x10^{-1})+(1x10^{-2})+(4x10^{-3})$

$= (2x1000)+(7x100)+(4x10)+(5x1)+(2x0.1)+(1x0.01)+(4x0.001)$

$= (2000)+(700)+(40)+(5)+(0.2)+(0.01)+(0.004)$

BINARY NUMBER SYSTEM

- Base 2 number system
- Uses two digits: 0 and 1
- Each position in a binary number represents a 0 power of the base (2). Example 2^0
- Weightage of each bit depends on the bit position
- Example :

  $1011_2 = (1\times2^3) + (0\times2^2) + (1\times2^1) + (1\times2^0)$

POSITIVE POWERS OF TWO (WHOLE NUMBERS)									NEGATIVE POWERS OF TWO (FRACTIONAL NUMBER)					
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
256	128	64	32	16	8	4	2	1	1/2	1/4	1/8	1/16	1/32	1/64
									0.5	0.25	0.125	0.0625	0.03125	0.015625

BINARY NUMBER SYSTEM

For digital systems, the binary number system is used. Binary has a radix of two and uses the digits 0 and 1 to represent quantities.

The column weights of binary numbers are powers of two that increase from right to left beginning with $2^0 = 1$:

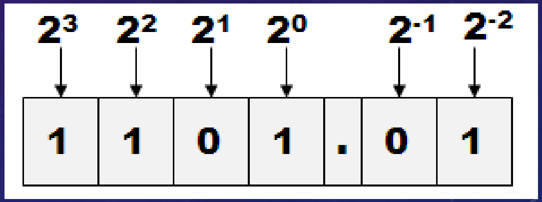
$\ldots 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0.$

For fractional binary numbers, the column weights are negative powers of two that decrease from left to right:

$2^2\ 2^1\ 2^0.\ 2^{-1}\ 2^{-2}\ 2^{-3}\ 2^{-4}\ \ldots$

BINARY NUMBER SYSTEM

- Binary system is used internally by all modern computers.
- Similarly to the decimal number system, the position of each digit also carries a weight.
- For example, 1101.01



- 1101.01_2
- $=((1x2^3) + (1x2^2) + (0x2^1) + (1x2^0) + (0x2^{-1}) + (1x2^{-2}))_{10}$
- $=((1x8) + (1x4) + (0x2) + (1x1) + (0x0.5) + (1x0.25))_{10}$
- $= (8 + 4 + 0 + 1 + 0 + 0.25)_{10}$
- $= 13.25_{10}$

TEST YOURSELF

Convert the following binary numbers to decimal

01101001

TEST YOURSELF

Convert the following binary numbers to decimal

01101001

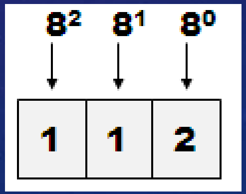
$=0 \times 128 + 1 \times 64 + 1 \times 32 + 0 \times 16 + 1 \times 8 + 0 \times 4 + 0 \times 2 + 1 \times 1$

$= 64 + 32 + 8 + 1$

$= 105_{10}$

OCTAL NUMBER SYSTEM

- The octal numeral system uses the digits 0 to 7.
- Base 8 number system.
- The position of each digit in an octal number can be assigned a weight. Example 8^0
- 112_8
- = (1 x $8^{2)}$ + (1 x $8^{1)}$ + (2 x 8^0)
- = $(64 + 8 + 2)_{10}$
- = 74_{10}



Octal Numbers

Octal uses eight characters the numbers 0 through 7 to represent numbers. There is no 8 or 9 character in octal.

Binary number can easily be converted to octal by grouping bits 3 at a time and writing the equivalent octal character for each group.

Decimal	Octal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	10	1000
9	11	1001
10	12	1010
11	13	1011
12	14	1100
13	15	1101
14	16	1110
15	17	1111

Example

Express $1\ 001\ 011\ 000\ 001\ 110_2$ in octal:

Solution

Group the binary number by 3-bits starting from the right. Thus, 113016_8

Octal Numbers

Octal is also a weighted number system. The column weights are powers of 8, which increase from right to left.

Column weights	8^3	8^2	8^1	8^0
	512	64	8	1

Example Express 3702_8 in decimal.

Solution Start by writing the column weights:

512 64 8 1

3 7 0 2_8

$3(512) + 7(64) + 0(8) + 2(1) = 1986_{10}$

Decimal	Octal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	10	1000
9	11	1001
10	12	1010
11	13	1011
12	14	1100
13	15	1101
14	16	1110
15	17	1111

HEXADECIMAL NUMBER SYSTEM

- Base 16. Example 16^0
- 16 symbols : Uses 10 digits and 6 letters,

  0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
- Letters represents numbers starting from 10.

  A = 10, B = 11, C = 12, D = 13, E = 14, F = 15
- Last position in a hexadecimal number represents a x power of the base (16). Example 16^x where x represents the last position -1.

HEXADECIMAL NUMBER SYSTEM

- Often used in digital electronics and computer engineering.
- The position of each digit in a hexadecimal number can be assigned a weight.

HEXADECIMAL NUMBER SYSTEM

- $=19FDE_{16}$
- $= ((1x16^4)+(9x16^3)+(Fx16^2)+(Dx16^1)+(Ex16^0))_{10}$
- $=((1x16^4)+(9x16^3)+(15x16^2)+(13x16^1)+(14x16^0))_{10}$
- $=(65536 + 36864 + 3840 + 208 + 14)_{10}$
- $=106462_{10}$

HEXADECIMAL NUMBER SYSTEM

Large binary number can easily be converted to hexadecimal by grouping bits 4 at a time and writing the equivalent hexadecimal character.

Example Express 1001 0110 0000 1110_2 in hexadecimal:

Solution Group the binary number by 4-bits starting from the right. Thus, 960E

Decimal	Hexadecimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Hexadecimal Numbers

Hexadecimal is a weighted number system. The column weights are powers of 16, which increase from right to left.

Column weights $\begin{cases} 16^3 \quad 16^2 \quad 16^1 \quad 16^0 \\ 4096 \quad 256 \quad 16 \quad 1 \end{cases}$

Example Express $1A2F_{16}$ in decimal.

Solution Start by writing the column weights:

4096 256 16 1

1 A 2 F_{16}

$1(4096) + 10(256) + 2(16) + 15(1) = 6703_{10}$

Decimal	Hexadecimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Hexadecimal

Base 16:
Digits 0 to 9 and letters A,B,C,D,E,F to represent the decimal numbers 10,11,12,13,14,15

Position	**4**	**3**	**2**	**1**
Weight	**4096**	**256**	**16**	**1**
Hex number	**1**	**C**	**A**	**D**

1CAD16 = 1 x 4096 + 12 x 256 + 10 x 16 + 13 x 1 = 7341_{10}

Decimal	Binary	Octal	Hexadecimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

WHY HEXADECIMAL?

- Let's say you are writing a program, and a very important data is 1111101011011110_2.
- Can anyone tell me what that number was again?
- Now, try this: $FADE_{16}$.
- What was that number again?
- $FADE_{16}$. See?

MEMORISATION

- Memorise this much, at least

Decimal	Binary	Hexadecimal
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

THE POWER OF BASES

Base 10 (decimal), Base 2 (binary) & Base 16 (hexadecimal)

$$7 \cdot \boxed{10}^{2} + 5 \cdot 10^{1} + 4 \cdot 10^{\boxed{0}} = 700 + 50 + 4 = 754$$

base

digit position

10100101b =

$$= 1 \cdot \boxed{2}^{7} + 0 \cdot 2^{6} + 1 \cdot 2^{5} + 0 \cdot 2^{4} + 0 \cdot 2^{\boxed{3}} + 1 \cdot 2^{2} + 0 \cdot 2^{1} + 1 \cdot 2^{0}$$

$$= 128 + 0 + 32 + 0 + 0 + 4 + 0 + 1 = 165$$

(decimal value)

base

digit position

$$1 \cdot 16^{3} + 2 \cdot \boxed{16}^{2} + 3 \cdot 16^{1} + 4 \cdot 16^{\boxed{0}} = 4096 + 512 + 48 + 4 = 4660$$

(decimal value)

base

digit position

NUMBERING SYSTEMS CONVERSION

- Binary ⇨ Decimal (sum of all base 2 weighting)
- Binary ⇨ Hexadecimal (4 bits grouping)
- Binary ⇨ Octal (3 bits grouping)
- Decimal ⇨ Binary (divide by 2)
- Decimal ⇨ Hexadecimal (divide by 16)
- Decimal ⇨ Octal (divide by 8)
- Hexadecimal ⇨ Binary (equivalent 4 bits of each hexadecimal digit)
- Hexadecimal ⇨ Decimal (sum of all base 16 weighting)
- Octal ⇨ Binary (equivalent 3 bits of each digit)
- Octal ⇨ Decimal (sum of all base 8 weighting)

CONVERSION OF NUMBER SYSTEM

DECIMAL TO OTHER BASE SYSTEM

- Step 1 : Divide the decimal number to be converted by the value of the new base.
- Step 2: Get the remainder from Step 1 as the rightmost digit (least significant digit) of new base number.
- Step 3: Divide the quotient of the previous divide by the new base.
- Step 4: Record the remainder from Step 3 as the next digit (to the left) of the new base number.

CONVERSION OF NUMBER SYSTEM

DECIMAL TO OTHER BASE SYSTEM

- Repeat the Steps 3 and 4, getting remainders from right to left, until the quotient becomes zero in Step 3.
- The last remainder thus obtained will be the most significant digit (MSD) of the new base number.

CONVERSION OF NUMBER SYSTEM

DECIMAL TO OTHER BASE SYSTEM

Decimal number : 29

Step	Operation	Result	Remainder
Step 1	29/2	14	1
Step 2	14/2	7	0
Step 3	7/2	3	1
Step 4	3/2	1	1
Step 5	1/2	0	1

Decimal number : 29_{10} = Binary Number : 11101_2

CONVERSION OF NUMBER SYSTEM

Converting a Number of Another Base to a Decimal Number

Method

Step 1 : Determine the column (positional) value of each digit

Step 2 : Multiply the obtained column values by the digits in the corresponding columns

Step 3 : Calculate the sum of these products.

Decimal	Binary	Octal	Hexidecimal
0	00000000	0	0
1	00000001	1	1
2	00000010	2	2
3	00000011	3	3
4	00000100	4	4
5	00000101	5	5
6	00000110	6	6
7	00000111	7	7
8	00001000	10	8
9	00001001	11	9
10	00001010	12	A
11	00001011	13	B
12	00001100	14	C
13	00001101	15	D
14	00001110	16	E
15	00001111	17	F
16	00010000	20	10
17	00010001	21	11
18	00010010	22	12
19	00010011	23	13
20	00010100	24	14
21	00010101	25	15
22	00010110	26	16
23	00010111	27	17
24	00011000	30	18
25	00011001	31	19
26	00011010	32	1A
27	00011011	33	1B
28	00011100	34	1C
29	00011101	35	1D
30	00011110	36	1E
31	00011111	37	1F
32	00100000	40	20

Decimal to Binary Conversion

2 |	46	(decimal)	
2 |	23	remainder	0
2 |	11	remainder	1
2 |	5	remainder	1
2 |	2	remainder	1
2 |	1	remainder	0
	0	remainder	1

i.e. $\mathbf{46_{10} = 101110_2}$

Decimal to Hex Conversion

16 |	1982_{10}		
16 |	123	remainder	14 (E)
16|	7	remainder	11 (B)
	0	remainder	7

i.e. $\mathbf{1982_{10} = 7BE_{16}}$

BCD

Binary coded decimal (BCD) is a weighted code that is commonly used in digital systems when it is necessary to show decimal numbers such as in clock displays.

The table illustrates the difference between straight binary and BCD. BCD represents each decimal digit with a 4-bit code. Notice that the codes 1010 through 1111 are not used in BCD.

Decimal	Binary	BCD
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	1010	0001 0000
11	1011	0001 0001
12	1100	0001 0010
13	1101	0001 0011
14	1110	0001 0100
15	1111	0001 0101

<u>Table ASCII -I</u>

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

TABLE ASCII -II

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ç	160	A0	á	192	C0	└	224	E0	α
129	81	ü	161	A1	í	193	C1	┴	225	E1	ß
130	82	é	162	A2	ó	194	C2	┬	226	E2	Γ
131	83	â	163	A3	ú	195	C3	├	227	E3	π
132	84	ä	164	A4	ñ	196	C4	─	228	E4	Σ
133	85	à	165	A5	Ñ	197	C5	┼	229	E5	σ
134	86	å	166	A6	ª	198	C6	╞	230	E6	µ
135	87	ç	167	A7	º	199	C7	╟	231	E7	τ
136	88	ê	168	A8	¿	200	C8	╚	232	E8	Φ
137	89	ë	169	A9	⌐	201	C9	╔	233	E9	Θ
138	8A	è	170	AA	¬	202	CA	╩	234	EA	Ω
139	8B	ï	171	AB	½	203	CB	╦	235	EB	δ
140	8C	î	172	AC	¼	204	CC	╠	236	EC	∞
141	8D	ì	173	AD	¡	205	CD	═	237	ED	φ
142	8E	Ä	174	AE	«	206	CE	╬	238	EE	ε
143	8F	Å	175	AF	»	207	CF	╧	239	EF	∩
144	90	É	176	B0	░	208	D0	╨	240	F0	≡
145	91	æ	177	B1	▒	209	D1	╤	241	F1	±
146	92	Æ	178	B2	▓	210	D2	╥	242	F2	≥
147	93	ô	179	B3	│	211	D3	╙	243	F3	≤
148	94	ö	180	B4	┤	212	D4	╘	244	F4	⌠
149	95	ò	181	B5	╡	213	D5	╒	245	F5	⌡
150	96	û	182	B6	╢	214	D6	╓	246	F6	÷
151	97	ù	183	B7	╖	215	D7	╫	247	F7	≈
152	98	ÿ	184	B8	╕	216	D8	╪	248	F8	°
153	99	Ö	185	B9	╣	217	D9	┘	249	F9	∙
154	9A	Ü	186	BA	║	218	DA	┌	250	FA	·
155	9B	¢	187	BB	╗	219	DB	█	251	FB	√
156	9C	£	188	BC	╝	220	DC	▄	252	FC	ⁿ
157	9D	¥	189	BD	╜	221	DD	▌	253	FD	²
158	9E	₧	190	BE	╛	222	DE	▐	254	FE	■
159	9F	ƒ	191	BF	┐	223	DF	▀	255	FF	□

Binary Addition

The rules for binary addition are

0 + 0 = 0	Sum = 0, carry = 0
0 + 1 = 0	Sum = 1, carry = 0
1 + 0 = 0	Sum = 1, carry = 0
1 + 1 = 10	Sum = 0, carry = 1

When an input carry = 1 due to a previous result, the rules are

1 + 0 + 0 = 01	Sum = 1, carry = 0
1 + 0 + 1 = 10	Sum = 0, carry = 1
1 + 1 + 0 = 10	Sum = 0, carry = 1
1 + 1 + 1 = 10	Sum = 1, carry = 1

Binary Subtraction

The rules for binary subtraction are

$$0 - 0 = 0$$

$$1 - 1 = 0$$

$$1 - 0 = 1$$

$$10 - 1 = 1 \text{ with a borrow of } 1$$

Binary Addition

$$\begin{array}{rcl} 101101 & & (45_{10}) \\ \underline{111110} & + & (62_{10}) \\ 1101011 & & (107_{10}) \end{array}$$

Binary Subtraction

$$\begin{array}{rcl} 110011 & & (51) \\ \underline{011101} & - & (29) \\ 010110 & & (22) \end{array}$$

BINARY MULTIPLICATION

$$
\begin{array}{rll}
1101 & & (13_{10}) \\
\underline{1011} & \times & (11_{10}) \\
1101 & & \\
11010 & & \\
000000 & & \\
\underline{1101000} & + & \\
10001111 & & (143_{10})
\end{array}
$$

BINARY ADDITION – SELF TEST 1

1. 11011 + 1001010 =
2. 1011001 + 111010 =

- 11011 + 1001010 = 1100101:

$$
\begin{array}{cccccccc}
 & & 1 & 1 & & 1 & & \\
 & & & 1 & 1 & 0 & 1 & 1 \\
+ & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
\hline
 & 1 & 1 & 0 & 0 & 1 & 0 & 1
\end{array}
$$

- 1011001 + 111010 = 10010011:

$$
\begin{array}{ccccccccc}
 & 1 & 1 & 1 & 1 & & & & \\
 & & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\
+ & & & 1 & 1 & 1 & 0 & 1 & 0 \\
\hline
 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1
\end{array}
$$

BINARY SUBTRACTION – SELF TEST 2

1. 1000101 + 101100 =
2. 1110110 + 1010111 =

- 1000101 − 101100 = 11001:

$$
\begin{array}{cccccccc}
 & 0 & 1 & 1 & & & & \\
 & \cancel{1} & \cancel{{}^{1}0} & \cancel{{}^{1}0} & {}^{1}0 & 1 & 0 & 1 \\
- & & 1 & 0 & 1 & 1 & 0 & 0 \\
\hline
 & & & 1 & 1 & 0 & 0 & 1
\end{array}
$$

- 1110110 − 1010111 = 11111:

$$
\begin{array}{cccccccc}
 & & 0 & {}^{1}0 & 1 & {}^{1}0 & {}^{1}0 & \\
 & 1 & \cancel{1} & \cancel{1} & \cancel{{}^{1}0} & \cancel{1} & \cancel{1} & {}^{1}0 \\
- & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\
\hline
 & & & 1 & 1 & 1 & 1 & 1
\end{array}
$$

BINARY ADDITION (8-BIT FIXED WIDTH ADDITION) – SELF TEST 3

1. 1010110 + 110100 =
2. 110001+ 11100100 =

- 1010110 + 110100 = 10001010:

$$
\begin{array}{ccccccccc}
 & 1 & 1 & 1 & & 1 & & & \\
 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\
+ & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
\hline
 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0
\end{array}
$$

No overflow. Sum is correct.

- 110001 + 11100100 = 100010101:

$$
\begin{array}{cccccccccc}
\boxed{1} & 1 & 1 & & & & & & \\
 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
+ & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
\hline
 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1
\end{array}
$$

Overflow. The carry-out is discarded, and the sum is not correct