CADT
បណ្ឌិត្យសភាបច្ចេកវិទ្យាឌីជីថលកម្ពុជា
Cambodia Academy of Digital Technology

IDT
វិទ្យាស្ថានបច្ចេកវិទ្យាឌីជីថល
Institute of Digital Technology

Week 02

Number Systems: Binary Arithmetic and Conversion

Prepared by: **Leangsiv HAN**

23rd February 2024

Last Week

1. What is a Computer?
2. What is Programming Language?
3. Why do we need to learn Programming Languages?
4. What Programming Language will we learn?
5. What is an IDE?

Learning Objectives

By the end of this lesson, you will be to:

- ❑ explain the number system used in computers.
- ❑ learn about the types of number systems, digit symbols, and bases.
- ❑ calculate binary arithmetic.
- ❑ explain the method of number system conversions.

Introduction

Definition:

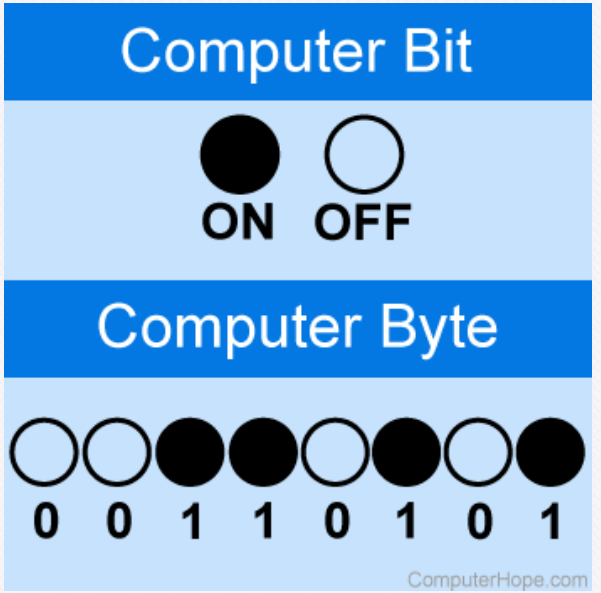
- **Number system** defines a set of values representing quantity.
- **Bit** (binary digit) is the smallest unit of data that a computer can process and store. A bit is always in one of two physical states, similar to an *on/off light switch*. *(Ref. TechTarget)*

Introduction

Definition:

- **Byte** is a unit of memory data equal to either *seven or eight bits*.

(Ref. Computer Hope)



Computer Memory Units

Unit	Equals
1 bit	Binary Digit (0 or 1)
8 bits	1 byte (B)
1 kilobyte (KB)	1024 bytes
1 megabyte (MB)	1, 048, 576 bytes
1 gigabyte (GB)	1, 073, 741, 824 bytes
1 terabyte (TB)	1, 099, 511, 627, 776 bytes
1 petabyte (PB)	1, 125, 899, 906, 842, 624 bytes

Types of Number Systems

Number System	Base	Digital Symbols
Binary	2	0, 1
Ternary	3	0, 1, 2
Quaternary	4	0, 1, 2, 3
Quinary	5	0, 1, 2, 3, 4
Octal	8	0, 1, 2, 3, 4, 5, 6, 7
Decimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Duodecimal	12	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B
Hexadecimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
Vigesimal	20	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, G, H, I, J

A list of number systems with their base and sets of valid digits

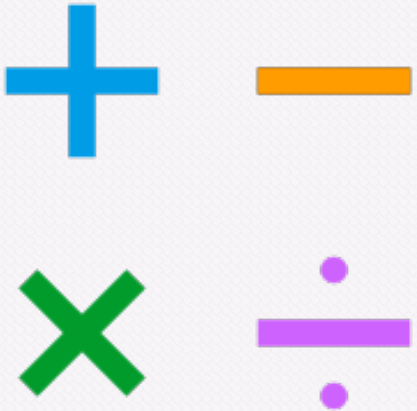
Types of Number Systems

Number System	Base	Digital Symbols
Binary	2	0, 1
Octal	8	0, 1, 2, 3, 4, 5, 6, 7
Decimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Hexadecimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Note: These are **COMMON NUMBER BASES.**

Binary Arithmetic

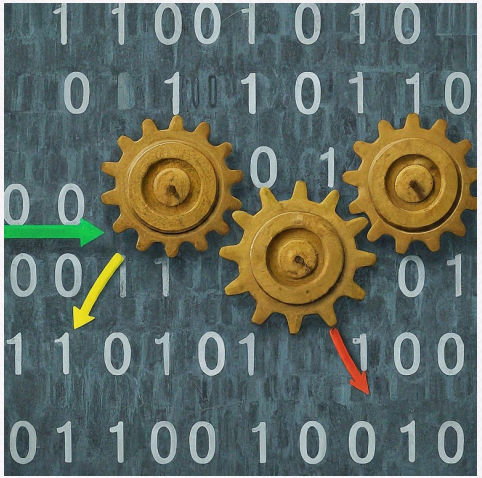
- Arithmetic: the part of mathematics that involves the adding and multiplying, etc. of numbers. *(Cambridge Dictionary)*



Binary Arithmetic

- In computers, numbers are represented in *binary format*, and all arithmetic operations involve binary numbers.

- The following slides detail how **binary arithmetic operations**, including addition (+), subtraction (−), multiplication (×), and division (÷), are executed.



Binary Addition(+)

Rule 1	Rule 2	Rule 3	Rule 4
0	0	1	1
+ 0	+ 1	+ 0	+ 1
0	1	1	10

Rules for binary addition

Example: $(1111)_2 + (110)_2 = (10101)_2$

Binary Subtraction(−)

Rule 1	Rule 2	Rule 3	Rule 4
0	1	1	0
- 0	- 1	- 0	- 1
0	0	1	1

Rules for binary Subtraction

Example: $(1110)_2 - (10101)_2 = (111)_2$

Binary Multiplication (×)

Binary multiplication uses the same techniques as **decimal multiplication**.

When performing binary multiplication, remember the following **rules**.

1. Copy the multiplicand when the multiplier digit is 1. Otherwise, write a row of zeros.
2. Shift the results one column to the left for a new multiplier digit.
3. Add the results using binary addition to find the product.

Example: $(1111)_2 \times (1011)_2 = (10100101)_2$

Binary Division (÷)

Division of binary numbers uses the same technique as **division in the decimal system**.

When doing binary division, some important **rules** need to be remembered.

1. When the remainder is greater than or equal to the divisor, write a 1 in the quotient and subtract.
2. When the remainder is less than the divisor, write a 0 in the quotient and add another digit from the dividend.
3. If all the digits of the dividend have been considered and there is still a remainder, mark a radix point in the dividend and append a zero. Remember that some fractions do not have an exact representation in binary, so not all division problems will terminate.

Example: $(11)_2 \div (1)_2 = (11)_2$; $(11)_2 \div (10)_2 = (1.1)_2$

Number System Conversions

Conversion: the process of converting something from one thing to another. *(Cambridge Dictionary)*

So, we are going to discuss the way **conversions from one number system to other number systems**.

Working with Integer Numbers (1/9)

Conversion of a decimal number to its binary equivalent

(Base-10 to Base-2)

Method: Repeated-division-by-2 method

1. Divide the dividend, that is, the decimal number by two and obtain the quotient and remainder.
2. Divide the quotient by two and obtain the new quotient and remainder.
3. Repeat step 2 until the quotient is equal to zero (0).
4. The first remainder produced is the least significant bit (LSB) in the binary number and the last remainder is the most significant bit (MSB). Accordingly, the binary number is then written (from left to right) with the MSB occurring first (list the remainder values in reverse order). This is the binary equivalent.

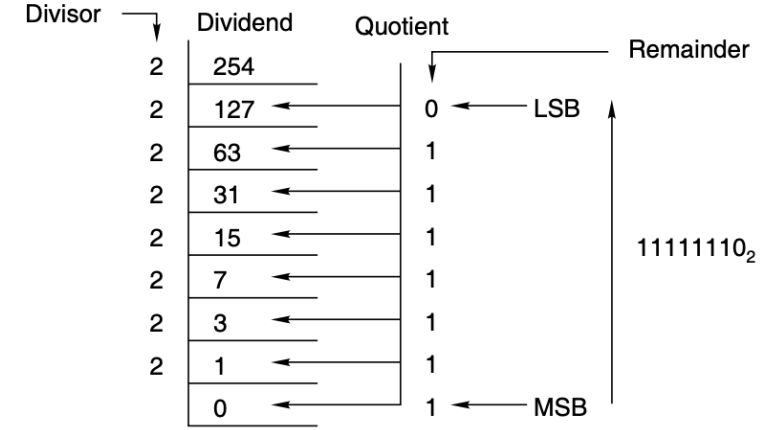
Working with Integer Numbers (2/9)

Conversion of a decimal number to its binary equivalent

(Base-10 to Base-2)

EXAMPLE

Converting the decimal number 254 into its binary equivalent.



Thus, the binary equivalent is **11111110.**

Working with Integer Numbers (3/9)

Conversion from binary to decimal

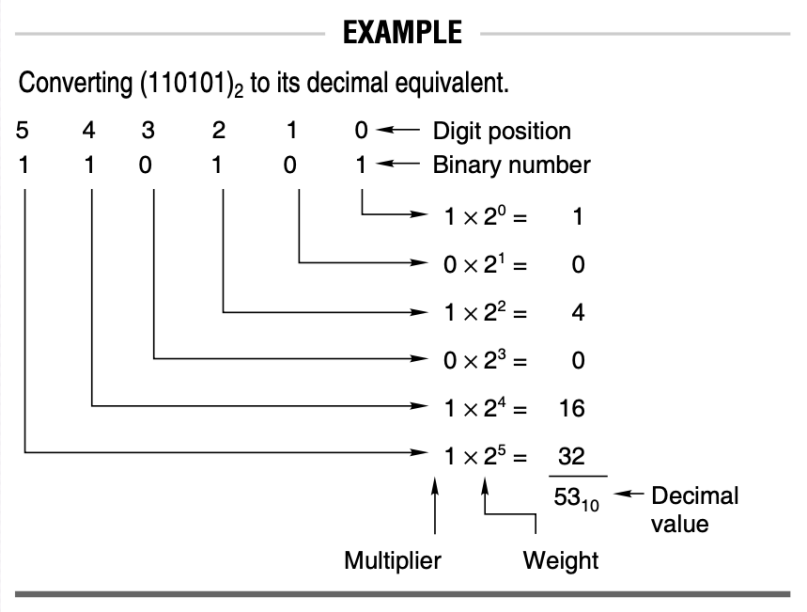
(Base-2 to Base-10)

To express the value of a given binary number as its decimal equivalent, sum the binary digits after each digit has been multiplied by its associated weight.

Working with Integer Numbers (4/9)

Conversion from binary to decimal

(Base-2 to Base-10)



Working with Integer Numbers (5/9)

Conversion of an octal number to its decimal equivalent

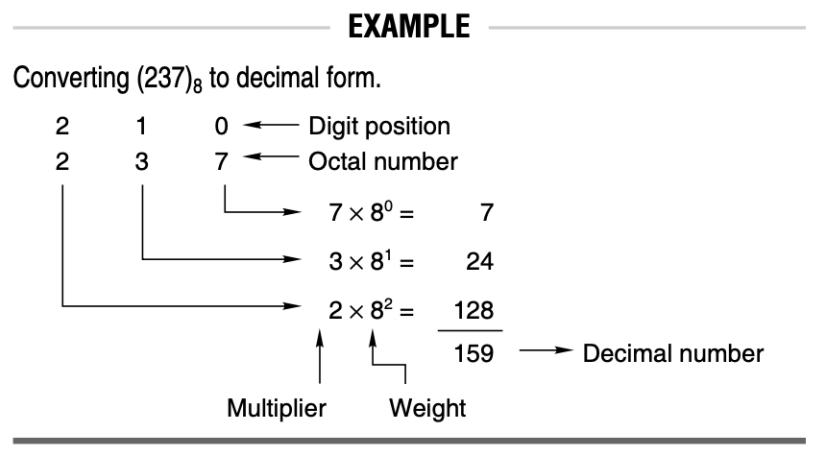
(Base-8 to Base-10)

To express the value of a given octal number as its decimal equivalent, add the octal digits after each digit has been multiplied by its associated weight.

Working with Integer Numbers (6/9)

Conversion of an octal number to its decimal equivalent

(Base-8 to Base-10)



Working with Integer Numbers (7/9)

Conversion of an octal number to its binary equivalent

(Base-8 to Base-2)

Octal digit	Equivalent binary number
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Binary equivalents for octal digits

Working with Integer Numbers (8/9)

Conversion of an octal number to its binary equivalent

(Base-8 to Base-2)

EXAMPLE

Converting the octal number 236700321 to its equivalent binary number.

Octal number	Equivalent binary number
2	010
3	011
6	110
7	111
0	000
0	000
3	011
2	010
1	001

Working with Integer Numbers (9/9)

- Self-study: try to learn more conversions, including:
  - Base-2 to Base-8
  - Base-10 to Base-16
  - Base-16 to Base-10
  - Base-16 to Base-2
  - Base-2 to Base-16
  - Base-8 to Base-16
  - Base-16 to Base-8

Working with fractional Numbers (1/5)

Conversion from decimal fractions to binary

When converting a fractional decimal value to binary, a slightly different approach is needed. Instead of dividing by 2, the decimal fraction is multiplied by 2. If the result is greater than or equal to 1, then 1 is to be put as the quotient. If the result is less than 1, then 0 is put as the quotient.

Working with fractional Numbers (2/5)

Conversion from decimal fractions to binary

EXAMPLE

Converting $(0.375)_{10}$ to binary.

$0.375 \times 2 = 0.750$	0
$0.750 \times 2 = 1.500$	1
$0.500 \times 2 = 1.000$	1

done.

Note that the last operation is complete when the fraction part equals zero. It is rarely possible to accurately represent a fractional value in binary. The answer to this problem is: **.011**

It is important to note that many decimal fractions do not have an exact representation in binary. This is illustrated in the following example.

Working with fractional Numbers (3/5)

Conversion from binary fraction to decimal

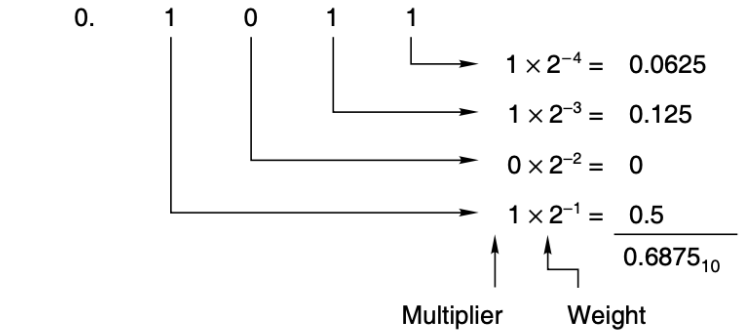
To express the value of a given fractional binary number in equivalent decimal value, each bit is multiplied by its associated weight and the summation of these gives the desired decimal number.

Working with fractional Numbers (4/5)

Conversion from binary fraction to decimal

EXAMPLE

Converting $(0.1011)_2$ to a decimal number.



Working with fractional Numbers (5/5)

- Self-study: try to learn more conversions of fractional number, including:
  - Conversion from octal fraction to decimal.
  - Conversion from decimal fractions to octal.
  - Conversion from octal fraction to binary.
  - Conversion from binary fraction to octal.
  - Conversion from hexadecimal fraction to binary.
  - Conversion from binary fraction to hexadecimal.

Table of ASCII values

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

ASCII stands for American Standard Code for Information Interchange.

Table of ASCII values

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[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	[NEGATIVE 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 OF 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	[SUBSTITUTE]	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	[DEL]

Online Base Calculator and Converter

- https://www.rapidtables.com/calc/math/base-calculator.html
- https://www.calculator.net/binary-calculator.html
- etc.

Key Takeaways

You are now able to:

- ✓ explain the number system used in computers.
- ✓ learn about the types of number systems, digit symbols, and bases.
- ✓ calculate binary arithmetic.
- ✓ explain the method of number system conversions.

Reference

- Dey, P., & Ghosh, M. (2013). *Computer fundamentals and programming in C*.

Week 02 – Homework

📣**Instruction**:

📝 Please put **No, Gen, Group, Name** [Follow the format put in the Homework].

📄 Submit your answer sheet as a Physical Paper.
🙌🏻 I require a **Mazer or Representative** to gather the homework from all your classmates and submit it to me during our next lab class.

‼️*Note:*
- ***No**: you can find your number in the list of students in "Student and Score" Topic.*
- *Late submissions will not be accepted.*
- *Didn't follow the instruction, 50% deductions.*
- *Cheating equals Zero* 🥚.

Thank you !

Questions or Feedbacks?

Contact Me via:

leangsiv.han@cadt.edu.kh

@leangsiv

leangsivhan