Number Systems

Logic Design of Digital Systems (300-1209) section 1

LECTURE 02

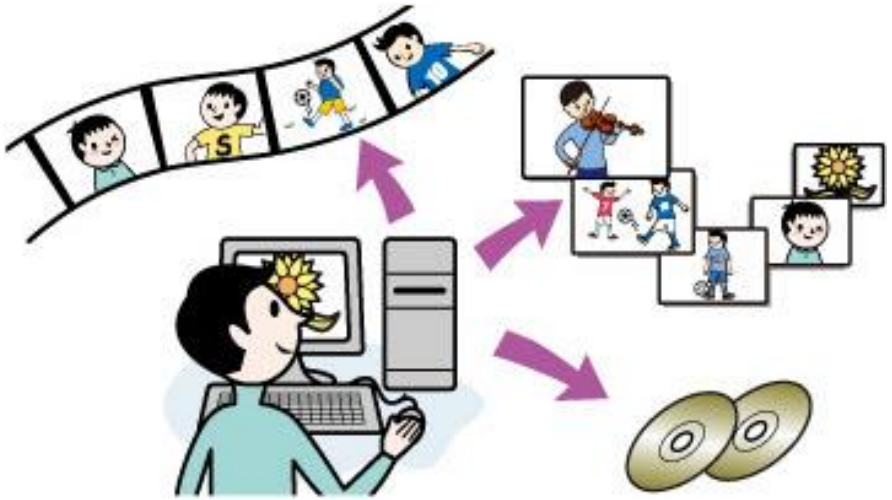
KRISADA PHROMSUTHIRAK

Digital Computers & Digital Systems

Computers are used in scientific calculations, commercial and business data processing, air traffic control, space guidance, the educational field, and many other areas.

The most striking property of a digital computer is its **generality**.

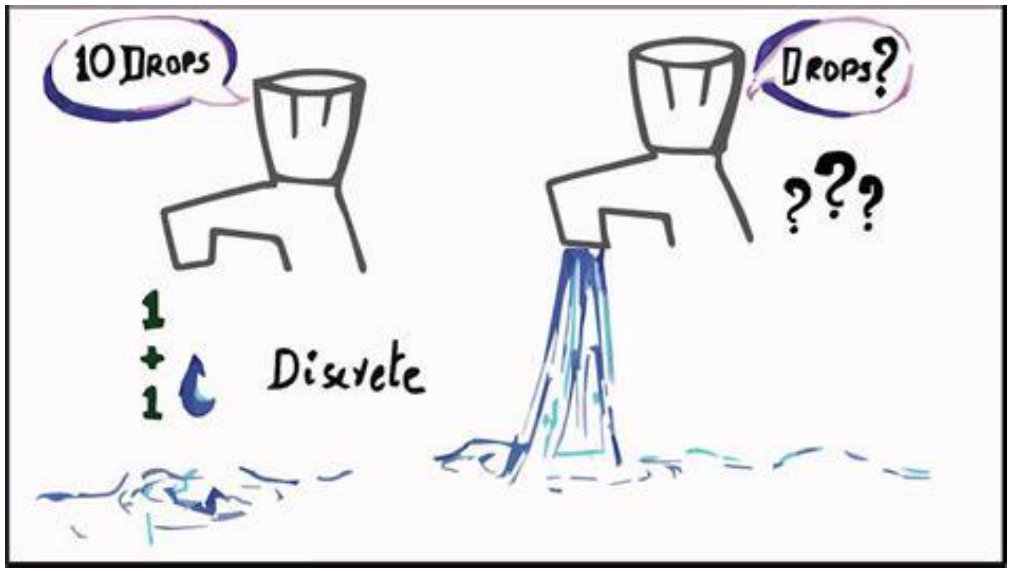
It can follow a sequence of instructions, called a **"program"**, that operates on given data. The user can specify and change programs and/or data according to the specific need.

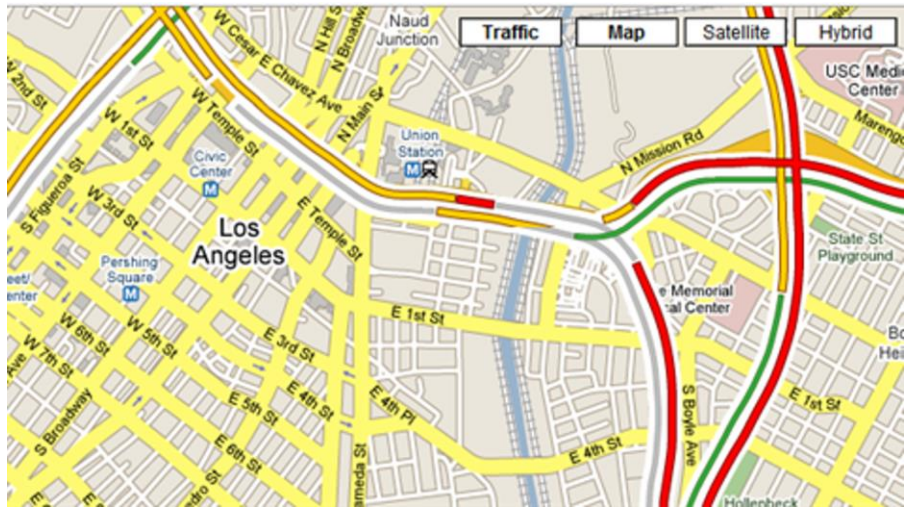


Digital Computers & Digital Systems

Characteristic of a digital system is its manipulation of *discrete elements* of information.

Such discrete elements may be *electric impulses*, the *decimal digits*, the *letters of an alphabet*, *arithmetic operations*, punctuation marks, or any other set of meaningful symbols.



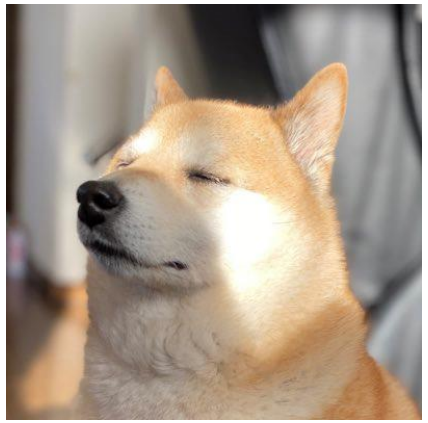


Traffic data in color label

Digital Computers & Digital Systems

For example, the letters d, o, and g form the word dog. We will be able to visualize what it means. Thus, a sequence of discrete elements forms a language, that is, a discipline that conveys information.

D O G



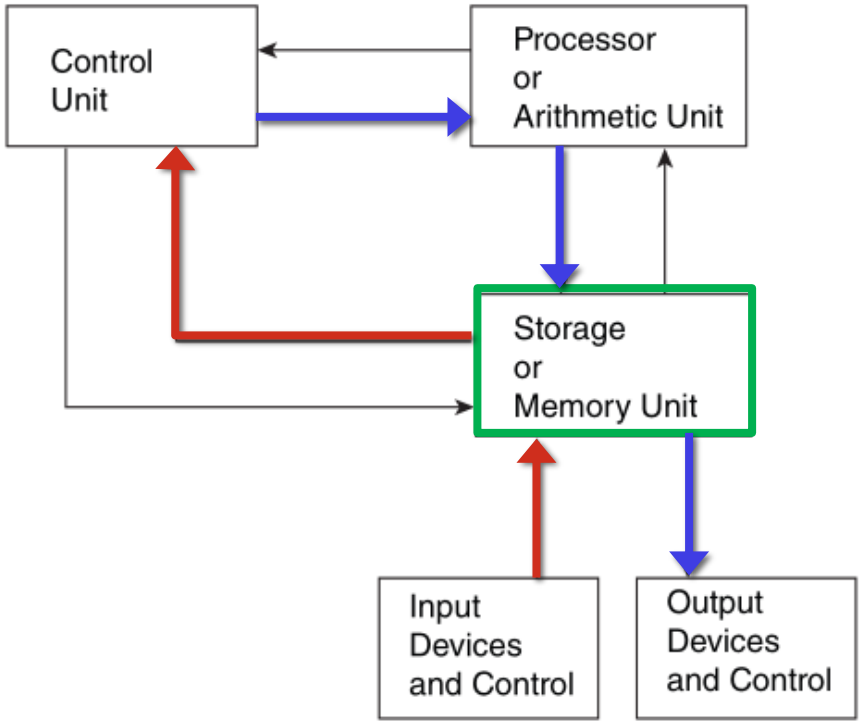
Early digital computers were used mostly for numerical computations. In this case the discrete elements used are the digits. From this application, the term digital computer has emerged. A more appropriate name for a digital computer would be a

"Discrete information processing system"

Digital Computers & Digital Systems

A block diagram of the digital computer

- ❑ The memory unit stores programs as well as input, output, and intermediate data.
- ❑ The processor unit performs arithmetic and other data-processing tasks as specified by a program.
- ❑ The control unit supervises the flow of information between the various units.



1. The control unit retrieves the instructions, one by one, from the program which is stored in memory.
2. For each instruction, the control unit informs the processor to execute the operation specified by the instruction.
3. Both **program** and **data** are stored in **memory**.
4. The control unit supervises the program instructions, and the processor manipulates the data as specified by the program.

Digital Computers & Digital Systems

It has already been mentioned that a digital computer manipulates discrete elements of information and that these elements are represented in the binary form. Operands used for calculations may be expressed in the binary number system.

Other discrete elements, including the decimal digits, are represented in binary codes. Data processing is carried out by means of binary logic elements using binary signals. Quantities are stored in binary storage elements.

That's why we have to focus on the binary system

and we will start gradually from the number system that we are familiar with it.

Number Systems

Decimal numbers, Binary numbers, and Base-r systems

Number Systems: Decimal Numbers

A decimal number such as 7392 represents a quantity equal to 7 thousands plus 3 hundreds, plus 9 tens, plus 2 units. The thousands, hundreds, etc., are powers of 10 implied by the position of the coefficients. To be more exact, 7392 should be written as:

$$(7 \times 10^3) + (3 \times 10^2) + (9 \times 10^1) + (2 \times 10^0)$$

However, the convention is to write only the coefficients and from their position deduce the necessary powers of 10. In general, a number with a decimal point is represented by a series of coefficients as follows:

The a_j coefficients are one of the ten digits (0, 1, 2,…, 9), and the subscript value j gives the place value and, hence, the power of 10 by which the coefficient must be multiplied/

$$(a_3 \times 10^3) + (a_2 \times 10^2) + (a_1 \times 10^1) + (a_0 \times 10^0)$$

Number Systems: Decimal Numbers

10^{+n}	10^2	10^1	10^0	decimal point	10^{-1}	10^{-2}	10^{-3}	10^{-n}
……	100	10	1		0.1	0.01	0.001	……

$$(3 \times 10^3) + (2 \times 10^2) + (5 \times 10^1) + (6 \times 10^0) + (2 \times 10^{-1}) + (5 \times 10^{-2}) + (7 \times 10^{-3})$$

$$(3{,}000) + (200) + (50) + (6) + (0.2) + (0.05) + (0.007) = 3{,}256.257$$

The decimal number system is said to be of base, or radix, 10 because it uses ten digits and the coefficients are multiplied by powers of 10.

Number Systems: Binary Numbers

The binary system is a different number system. The coefficients of the binary numbers system have two possible values: 0 and 1. Each coefficient a_j is multiplied by 2^j. For example, the decimal equivalent of the binary number 11010.11 is 26.75, as shown from the multiplication of the coefficients by powers of 2:

$$\ldots + (a_3 \times 2^3) + (a_2 \times 2^2) + (a_1 \times 2^1) + (a_0 \times 2^0) + (a_0 \times 2^{-1}) + (a_0 \times 2^{-2}) + \cdots$$

Example

$$(11010.11)_2$$

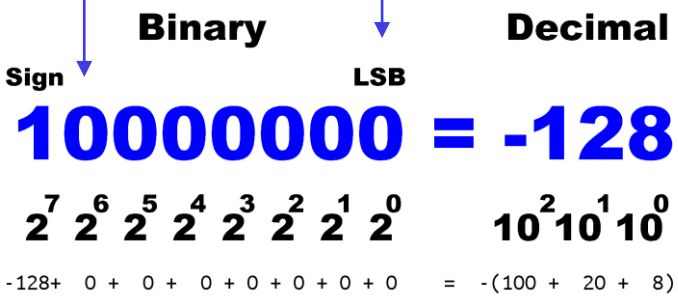
$$(1 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (0 \times 2^0) + (1 \times 2^{-1}) + (1 \times 2^{-2}) = 26.75$$

Number Systems: Binary Numbers

2^{+n}	2^2	2^1	2^0	point	2^{-1}	2^{-2}	2^{-3}	2^{-n}
……	4	2	1		0.5	0.25	0.125	……

Most Significant Bit (MSB)

Least Significant Bit (LSB)



ระบบเลขฐาน 2

(1) ไล่เลขฐาน 2 ตามลำดับจาก 0,1,2,3,4,5,...

ระบบเลขฐาน 2

[2] การแปลงเลขฐาน 2 ให้กลายเป็นเลขฐาน 10

ระบบเลขฐาน 2

[3] การแปลงเลขฐาน 10 ให้กลายเป็นเลขฐาน 2

Number Systems: base-r system (r<10 | r>10)

It is customary to borrow the needed r digits for the coefficients from the decimal system when the base of the number is less than 10.

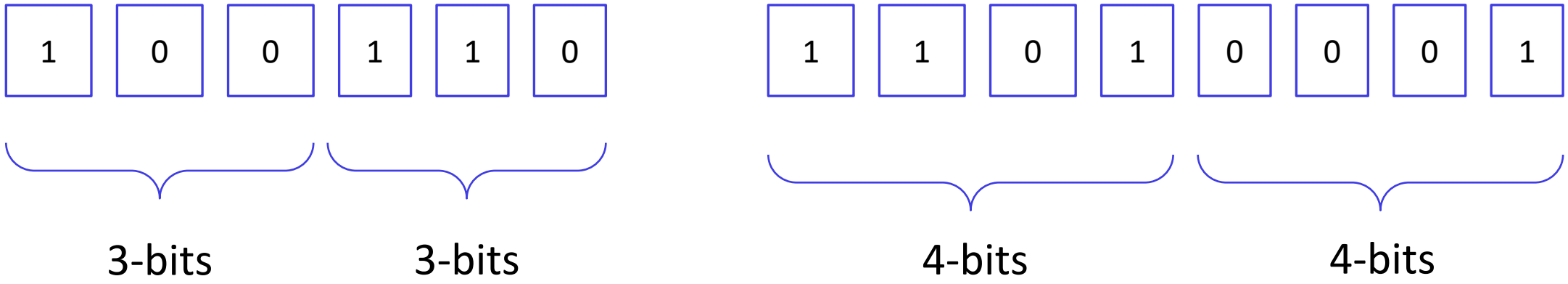
The letters of the alphabet are used to supplement the ten decimal digits when the base of the number is greater than 10. For example, in the hexadecimal (base 16) number system, the first ten digits are borrowed from the decimal system. The letters A, B, C, D, E, and F are used for digits 10, 11, 12, 13, 14, and 15, respectively.

An example of a hexadecimal number is:

$$(B65F)_{16} = 11 \times 16^3 + 6 \times 16^2 + 5 \times 16^1 + 15 = (46687)_{10}$$

Number Systems: base-r system (r<10 | r>10)

Why do we care about octal / hexadecimal number systems?



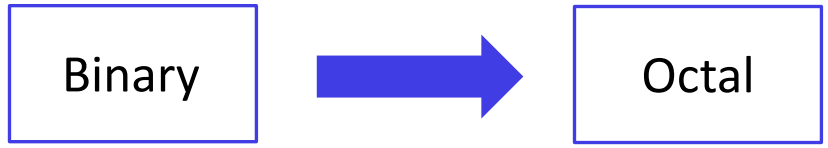
It's easy to look at binary numbers as a group.
(For Human)

Octal and Hexadecimal Numbers

The conversion from and to binary, octal, and hexadecimal plays an important part in digital computers. Since $2^3 = 8$ and $2^4 = 16$, each octal digit corresponds to three binary digits and each hexadecimal digit corresponds to four binary digits.

The conversion from binary to octal is easily accomplished by partitioning the binary number into groups of three digits each, The following example illustrates the procedure:

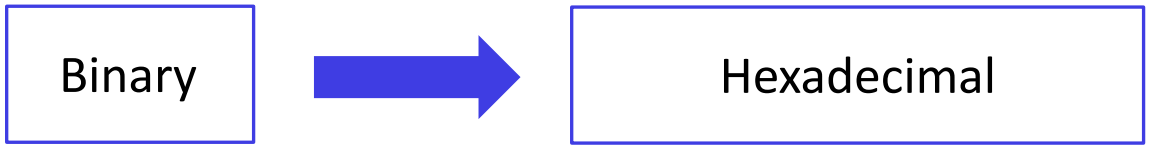
$$\left(\underbrace{10}_{2}\ \underbrace{110}_{6}\ \underbrace{001}_{1}\ \underbrace{101}_{5}\ \underbrace{011}_{3} \cdot \underbrace{111}_{7}\ \underbrace{100}_{4}\ \underbrace{000}_{0}\ \underbrace{110}_{6}\right)_2 = (26153.7406)_8$$



Octal and Hexadecimal Numbers

Conversion from binary to hexadecimal is similar, except that the binary number is divided into groups of four digits:

$$\left(\underbrace{10}_{2}\ \underbrace{1100}_{C}\ \underbrace{0110}_{6}\ \underbrace{1011}_{B}\ \bullet\ \underbrace{1111}_{F}\ \underbrace{0010}_{2}\right)_2 = (2C6B.F2)_{16}$$



Octal and Hexadecimal Numbers

Conversion from octal or hexadecimal to binary is done by a procedure reverse to the above. Each octal digit is converted to its three-digit binary equivalent. Similarly, each hexadecimal digit is converted to its four-digit binary equivalent. This is illustrated in the following examples:

Take out one by one

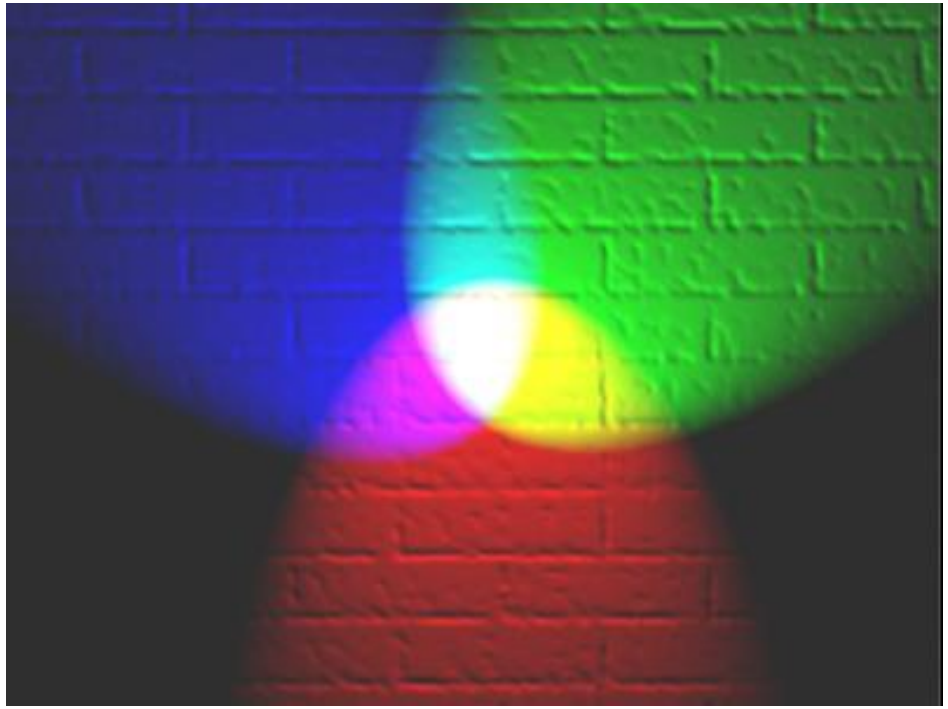
$$(673.124)_8 = \left(\underbrace{110}_{6} \underbrace{111}_{7} \underbrace{011}_{3} \bullet \underbrace{001}_{1} \underbrace{010}_{2} \underbrace{100}_{4} \right)_2$$

$$(306.\,D)_{16} = \left(\underbrace{0011}_{3} \underbrace{0000}_{0} \underbrace{0110}_{6} \bullet \underbrace{1101}_{D} \right)_2$$

Number Systems: base-r system (r<10 | r>10)

Web Color System

Name	Hex (RGB)	Red (RGB)	Green (RGB)	Blue (RGB)
White	#FFFFFF	100%	100%	100%
Silver	#C0C0C0	75%	75%	75%
Gray	#808080	50%	50%	50%
Black	#000000	0%	0%	0%
Red	#FF0000	100%	0%	0%
Maroon	#800000	50%	0%	0%
Yellow	#FFFF00	100%	100%	0%
Olive	#808000	50%	50%	0%
Lime	#00FF00	0%	100%	0%
Green	#008000	0%	50%	0%
Aqua	#00FFFF	0%	100%	100%
Teal	#008080	0%	50%	50%
Blue	#0000FF	0%	0%	100%
Navy	#000080	0%	0%	50%
Fuchsia	#FF00FF	100%	0%	100%
Purple	#800080	50%	0%	50%



A representation of additive color mixing. (256 levels)

16 x 16 = 256 ➔ FF

ระบบเลขฐาน 16

[x] ไล่เลขฐาน 16 ตามลำดับ

Arithmetic operations

base-r number systems

Arithmetic operations

Arithmetic operations with numbers in base r follow the same rules as for decimal numbers.
When other than the familiar base 10 is used, one must be careful to use only the r allowable digits.
Examples of addition, subtraction, and multiplication of two binary numbers are shown below:

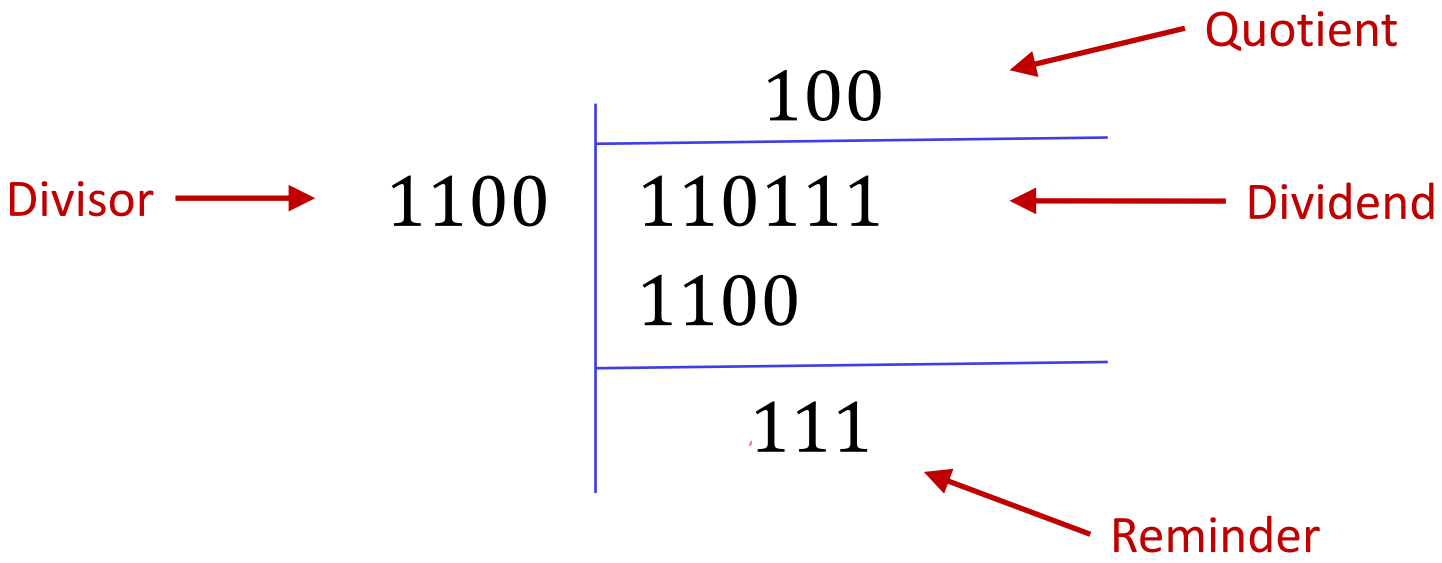
augend:	101101
addend:	+100111
sum:	1010100

minuend:	101101
subtrahend:	−100111
difference:	000110

multiplicand:	1011
multiplier:	× 101
	1011
	0000
	1011
product:	110111

Arithmetic operations

Arithmetic operations with numbers in base r follow the same rules as for decimal numbers.
When other than the familiar base 10 is used, one must be careful to use only the r allowable digits.
Examples of addition, subtraction, and multiplication of two binary numbers are shown below:

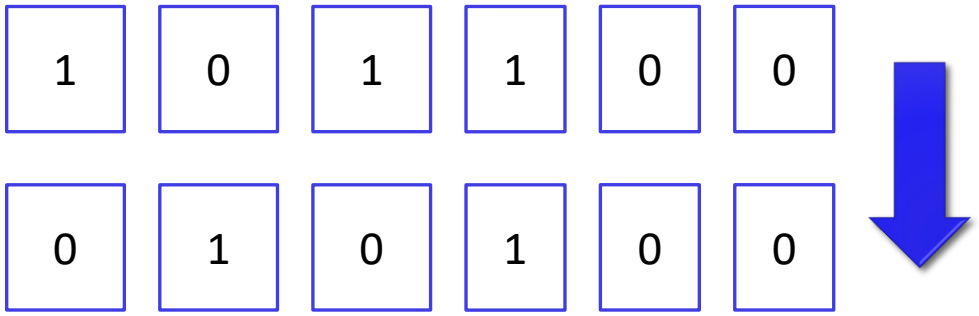


Take a Break

Complements (ส่วนเติมเต็ม)

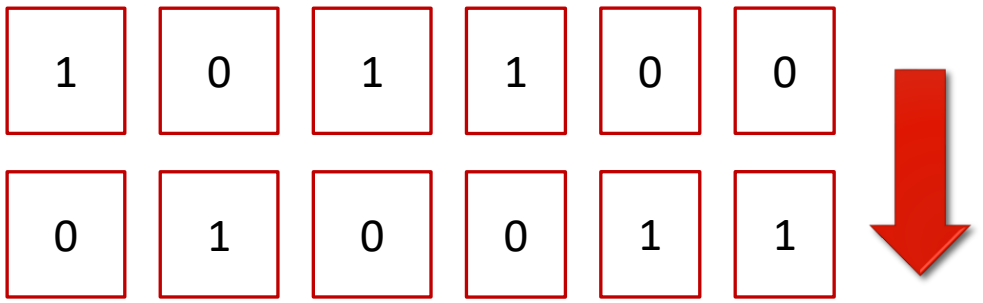
Complements are used in digital computers for simplifying the subtraction operation and for logical manipulations. There are two types of complements for each base-r system:

1. the r's complement
2. the (r – 1)'s complement.



r's complement

44 → 20



$(r-1)$'s complement

44 → 19

When the value of the base is substituted, the two types receive the names 2's and 1's complement for binary numbers, or 10's and 9's complement for decimal numbers.

Complement of Binary Systems

Binary systems $(r = 2)$

$r's$ complement

$2's$ complement

1	0	1
0	1	1

$(r-1)'s$ complement

$1's$ complement

1	0	1
0	1	0

Flip bits in Binary

Complement of Binary Systems

Binary number	1's complement	2's complement
000	111	000
001	110	111
010	101	110
011	100	101
100	011	100
101	010	011
110	001	010
111	000	001

ระบบเลขฐาน 2

[4] การ Complement (การหาส่วนเติมเต็ม) ของเลขฐาน 2

รูปแบบ r's complement  **รูปแบบ 2's complement**

ระบบเลขฐาน 2

[5] การ Complement (การหาส่วนเติมเต็ม) ของเลขฐาน 2

รูปแบบ (r-1)'s complement  **รูปแบบ 1's complement**

Subtraction with r's Complements

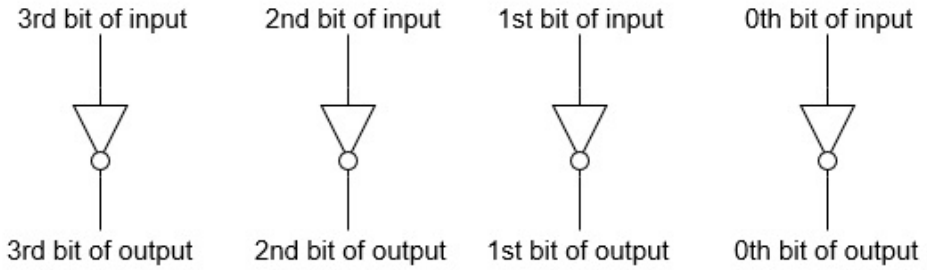
The direct method of subtraction taught in elementary schools uses the **borrow concept**.

When subtraction is implemented by means of digital components, this method is found to be less efficient than the method that uses complements and addition as stated below.

The subtraction of two positive numbers $(M - N)$, both of base r, may be done as follows:

1. Add the minuend M to the r's complement of the subtrahend N.
2. Inspect the result obtained in step 1 for an end carry:
   (a) If an end carry occurs, discard it.
   (b) If an end carry does not occur, take the r's complement of the number obtained in step 1 and place a negative sign in front.

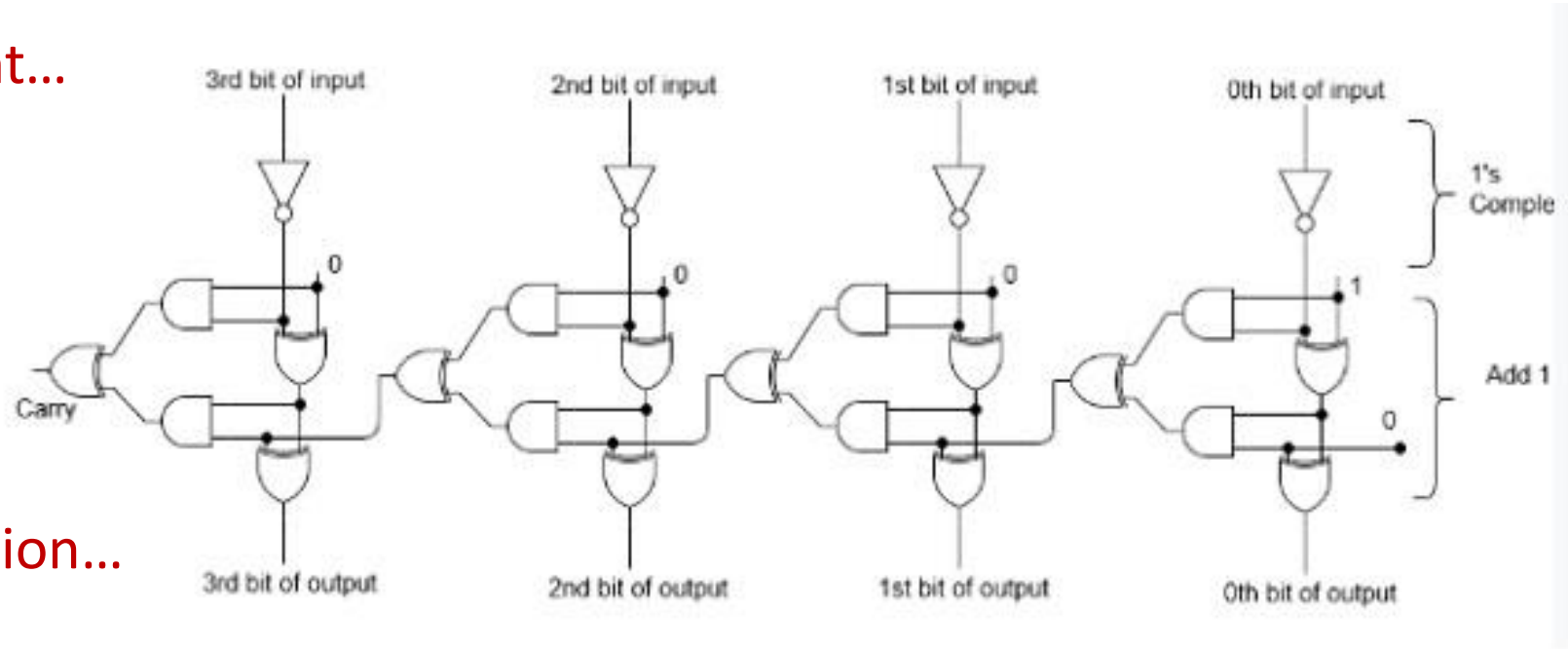
Why we need to do something difficult like this?



How can we do arithmetic with logic circuit?

We may do a complement…

We may do an addition…



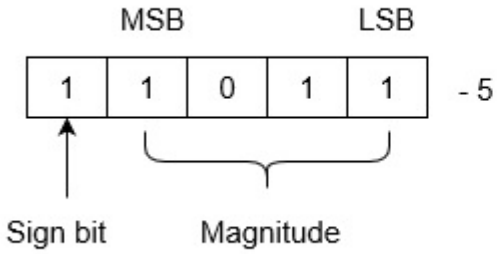
ระบบเลขฐาน 2

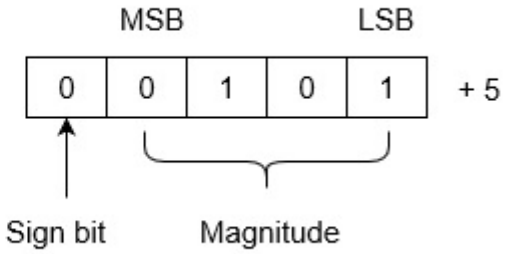
[6] การ Subtraction ด้วย **r's complement**

ระบบเลขฐาน 2

[7] การ Subtraction ด้วย **(r-1)'s complement**

Signed Binary number Representation

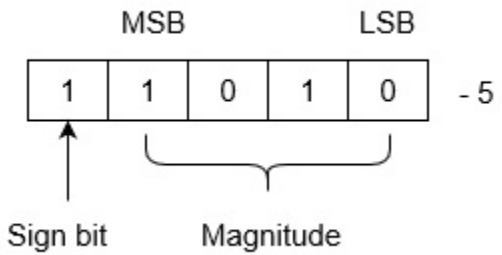


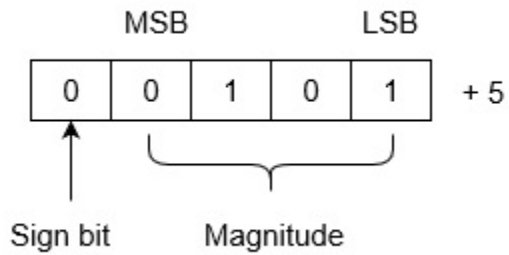


$+5 = 0\ 0101$

$-5 = 1\ 1011$

$2's$ complement



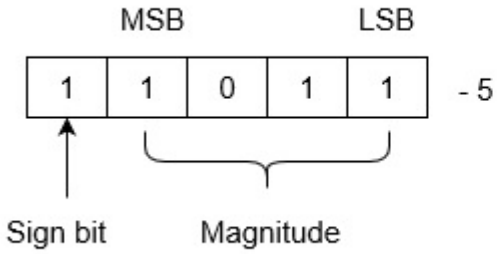


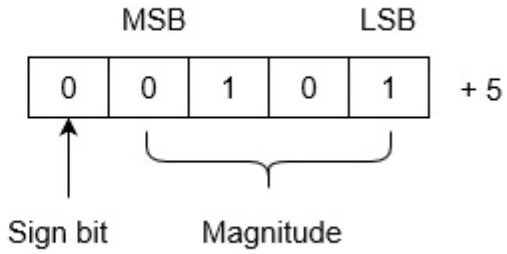
$+5 = 0\ 0101$

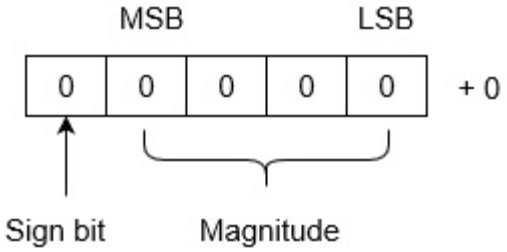
$-5 = 1\ 1010$

$1's$ complement

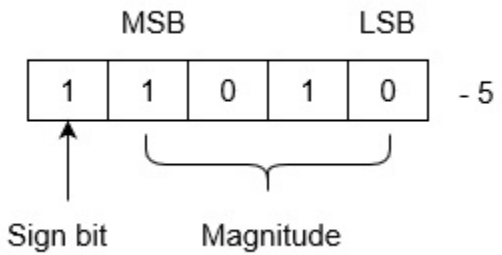
Signed Binary number Representation

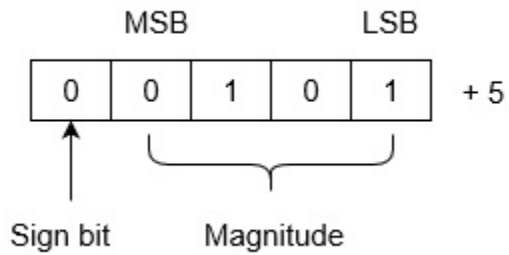




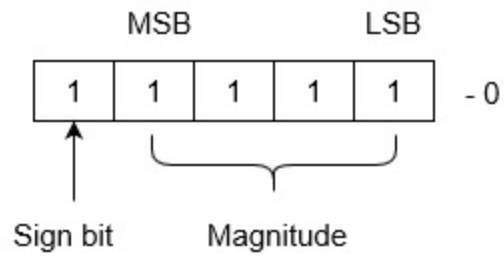


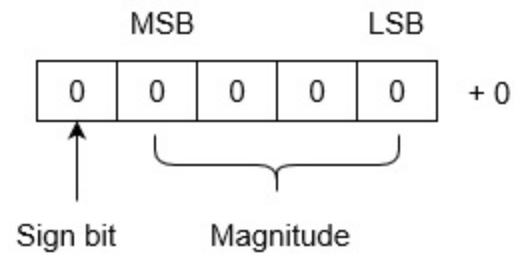
$2's$ complement





Drawback of 1's complement





$1's$ complement

ระบบเลขฐาน 2

[8] Signed Binary Number Representation ด้วย **r's complement**

ระบบเลขฐาน 2

[9] Signed Binary Number Representation ด้วย **(r-1)'s complement**

Take a Break

Binary Code

Binary Code

A bit, by definition, is a binary digit. When used in conjunction with a binary code, it is better to think of it as denoting a binary quantity equal to 0 or 1. To represent a group of 2^n distinct elements in a binary code requires a minimum of n bits.

Binary number of n digits

1	0	0	1	0

A

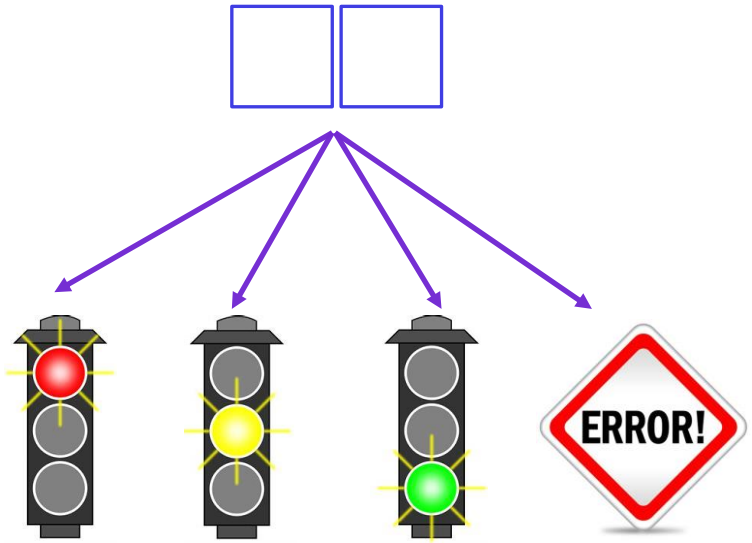
Digital systems represent and manipulate not only binary numbers, but also many other discrete elements of information. Any discrete element of information distinct among a group of quantities can be represented by a binary code.

Binary Code

The meaning of binary code is a group of binary numbers that can be used to represent discrete information.

The number of bits indicates the amount of data that can be collected.

Binary code 2 bits



Units derived from bit

- ❑ Nibble: group of 4 bits
- ❑ Byte: group of 8 bits
- ❑ Word / Block / Page

32-bit 64-bit

Computers usually manipulate bits in groups of a fixed size, conventionally called words. The number of bits in a word is usually defined by the size of the registers in the computer's CPU, or by the number of data bits that are fetched from its main memory in a single operation

Binary Code

Digital codes can be classified into two types

1. **Weighted Code**: The weighted codes are those that obey the position weighting principle, which states that the position of each number represent a specific weight.
2. **Non – weighted Code**: The non-weighted codes are not positionally weighted. In other words, codes that are not assigned with any weight to each digit position.

Binary Code: Binary-Coded Decimal (BCD)

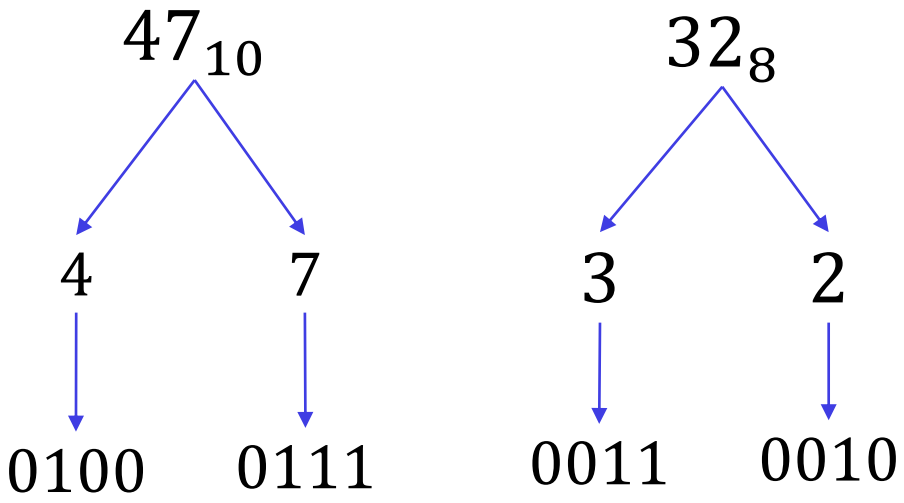
4-bit BCD codes and pseudo-tetrades

Bit	Weight	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Comment
4	8	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	Binary
3	4	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	
2	2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
Name		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Decimal

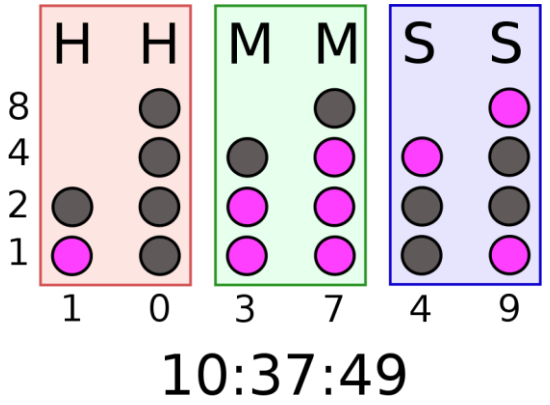
Binary-coded decimal (BCD) is a class of binary encodings of decimal numbers where each digit is represented by a fixed number of bits, usually four or eight. Sometimes, special bit patterns are used for a sign or other indications.

Binary Code: Binary-Coded Decimal (BCD)

Example: How to convert number to BCD



01000111_{BCD} 00110010_{BCD}

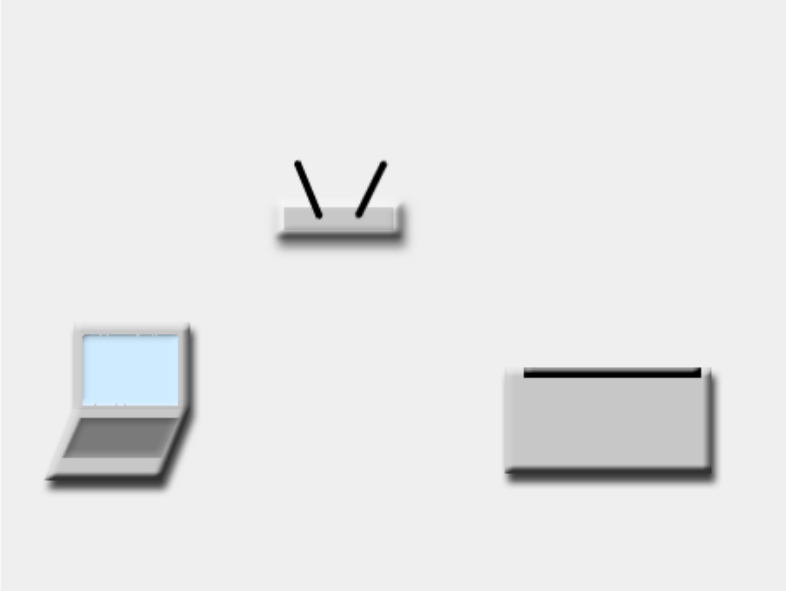


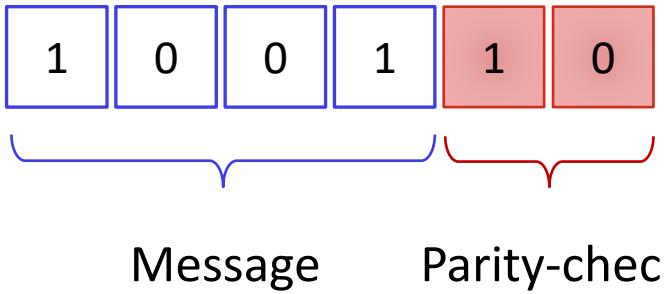
Digital Clock with 20 bits

Challenge !!!

How to design a logic circuit that makes a clock running.

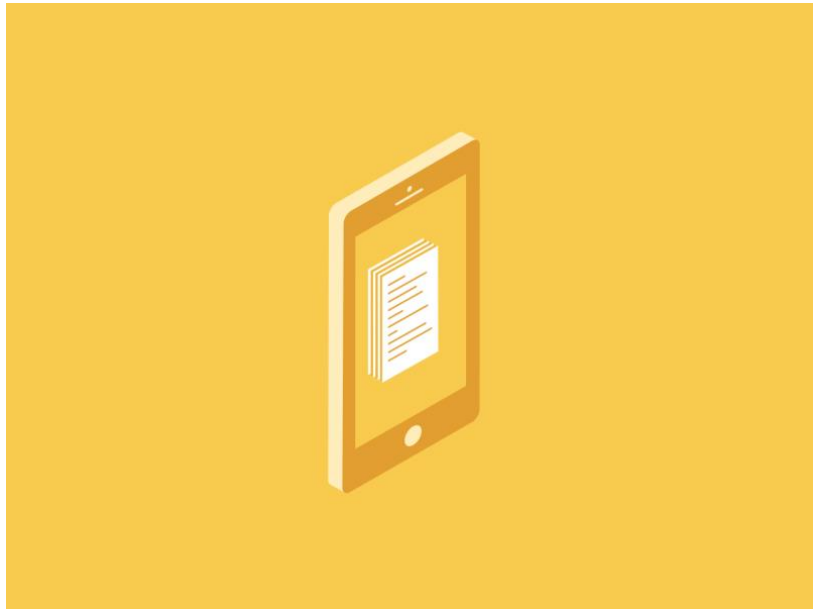
Binary Code: Error Detection Code (Parity-check)

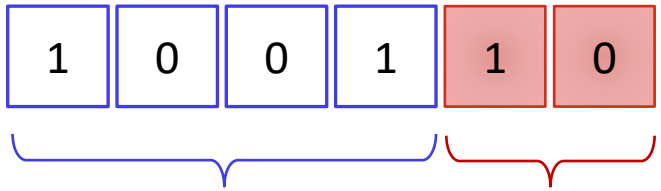


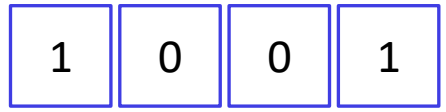


(a) Message	P(odd)	(b) Message	P (even)
0000	1	0000	0
0001	0	0001	1
0010	0	0010	1
0011	1	0011	0
0100	0	0100	1
0101	1	0101	0
0110	1	0110	0
0111	0	0111	1
1000	0	1000	1
1001	1	1001	0
1010	1	1010	0
1011	0	1011	1
1100	1	1100	0
1101	0	1101	1
1110	0	1110	1
1111	1	1111	0

Binary Code: Error Detection Code (Parity-check)







Count of 1-bits

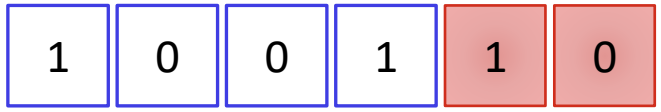
2

Odd parity-check
(including the parity bit)

1

Even parity-check
(including the parity bit)

0



Data ready to send

The Reflected Code (Gray Code)

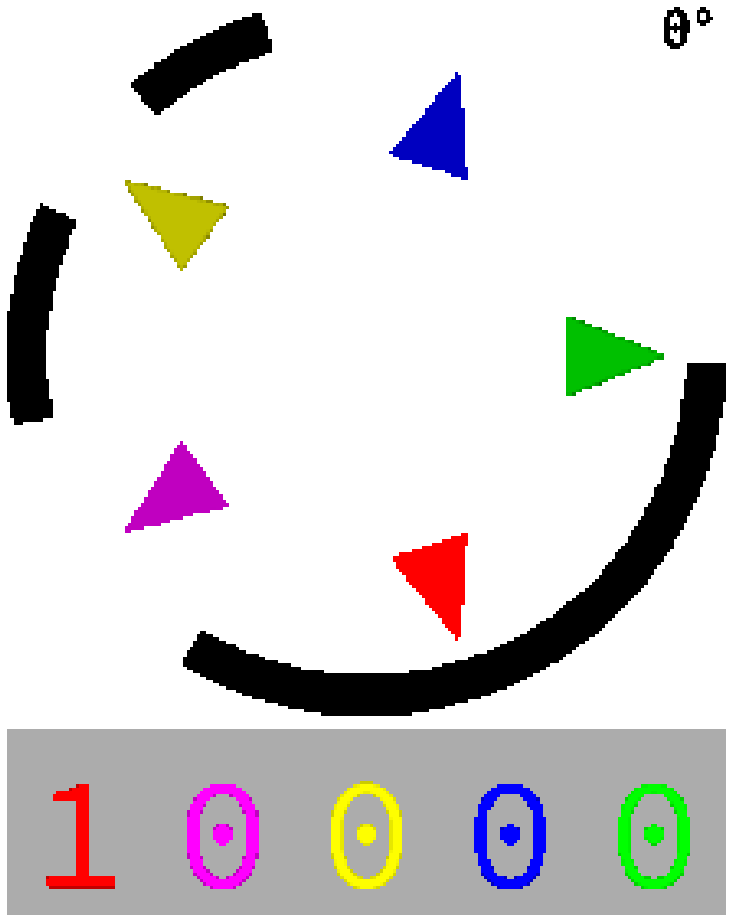
The reflected binary code or Gray code is an ordering of the binary numeral system such that two successive values differ in only one bit (binary digit). Gray codes are very useful in the normal sequence of binary numbers generated by the hardware that may cause an error or ambiguity during the transition from one number to the next. So, the Gray code can eliminate this problem easily since only one bit changes its value during any transition between two numbers.

Decimal	Binary	Gray Code
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

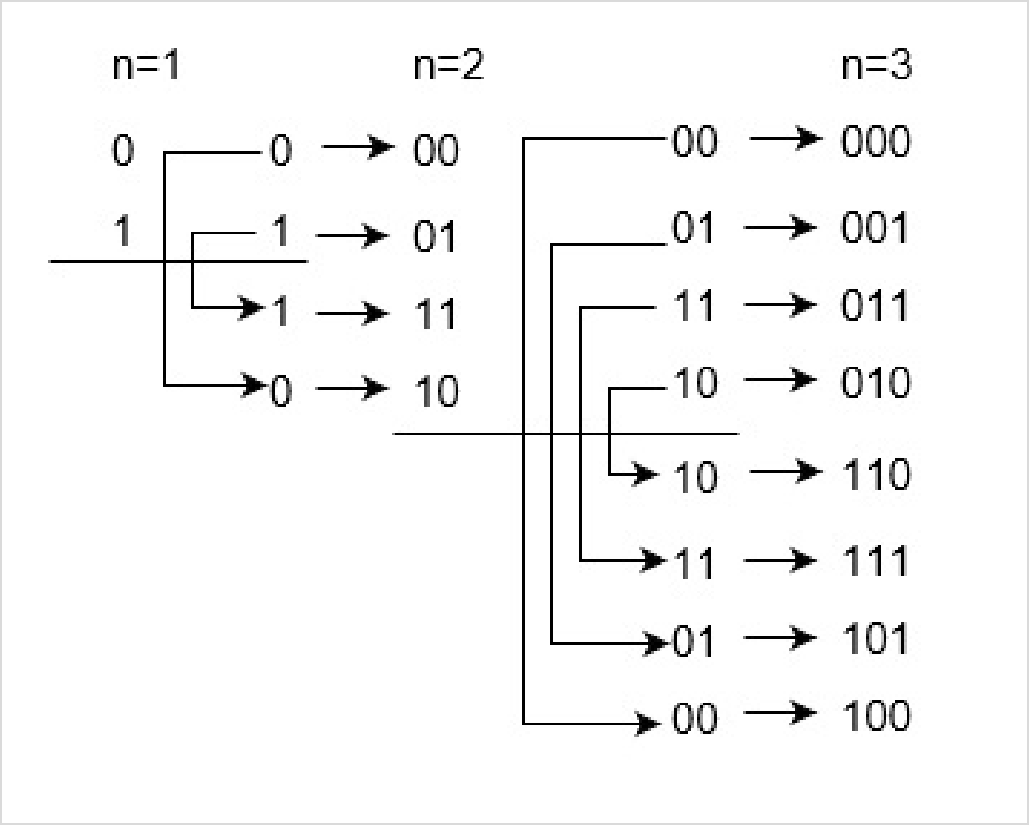
The Reflected Code (Gray Code)

Single-track Gray code for 30 positions

Angle	Code	Angle	Code	Angle	Code	Angle	Code	Angle	Code
0°	10000	72°	01000	144°	00100	216°	00010	288°	00001
12°	10100	84°	01010	156°	00101	228°	10010	300°	01001
24°	11100	96°	01110	168°	00111	240°	10011	312°	11001
36°	11110	108°	01111	180°	10111	252°	11011	324°	11101
48°	11010	120°	01101	192°	10110	264°	01011	336°	10101
60°	11000	132°	01100	204°	00110	276°	00011	348°	10001



The Reflected Code (Gray Code)



How to create Gray code from Binary code

Alphanumeric Codes

Since most of the information that humans perceive is in the form of letters, numbers, and symbols, so for the convenience of saving these data in the form of binary code we must have a standard for this information.

- **ASCII:** American Standard Code for Information Interchange, is a character encoding standard for electronic communication.

Control code chart

Binary	Oct	Dec	Hex	Abbreviation			[b]	[c]	[d]	Name (1967)
				1963	1965	1967				
000 0000	000	0	00	NULL	NUL		$^{N}_{L}$	^@	\0	Null
000 0001	001	1	01	SOM	SOH		$^{S}_{H}$	^A		Start of Heading
000 0010	002	2	02	EOA	STX		$^{S}_{T_X}$	^B		Start of Text
000 0011	003	3	03	EOM	ETX		$^{E}_{T_X}$	^C		End of Text
000 0100	004	4	04	EOT			$^{E}_{T}$	^D		End of Transmission

Example of ASCII code

Alphanumeric Codes

- **ASCII:** American Standard Code for Information Interchange, is a character encoding standard for electronic communication.

100 0001	101	65	41	A
100 0010	102	66	42	B
100 0011	103	67	43	C
100 0100	104	68	44	D
100 0101	105	69	45	E
100 0110	106	70	46	F
100 0111	107	71	47	G
100 1000	110	72	48	H
100 1001	111	73	49	I
100 1010	112	74	4A	J
100 1011	113	75	4B	K
100 1100	114	76	4C	L
100 1101	115	77	4D	M
100 1110	116	78	4E	N
100 1111	117	79	4F	O

101 0000	120	80	50	P
101 0001	121	81	51	Q
101 0010	122	82	52	R
101 0011	123	83	53	S
101 0100	124	84	54	T
101 0101	125	85	55	U
101 0110	126	86	56	V
101 0111	127	87	57	W
101 1000	130	88	58	X
101 1001	131	89	59	Y
101 1010	132	90	5A	Z

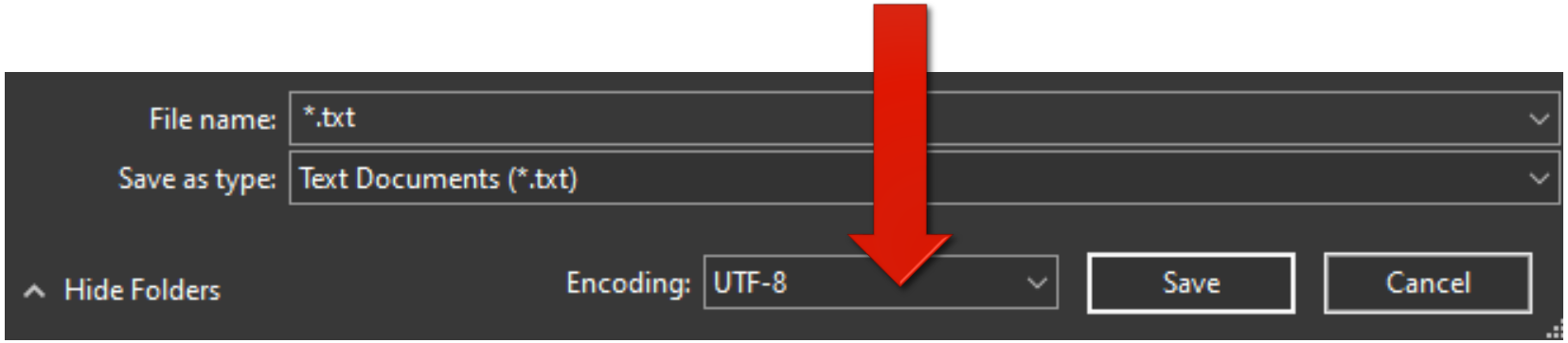
Example of ASCII code (Printable Character)

Alphanumeric Codes

- **Unicode**: Unicode can be implemented by different character encodings. The Unicode standard defines Unicode Transformation Formats (UTF): UTF-8, UTF-16, and UTF-32, and several other encodings

π Я 音 æ∞

When we trying to save the file with Notepad (.txt)
there is an option to encode the file

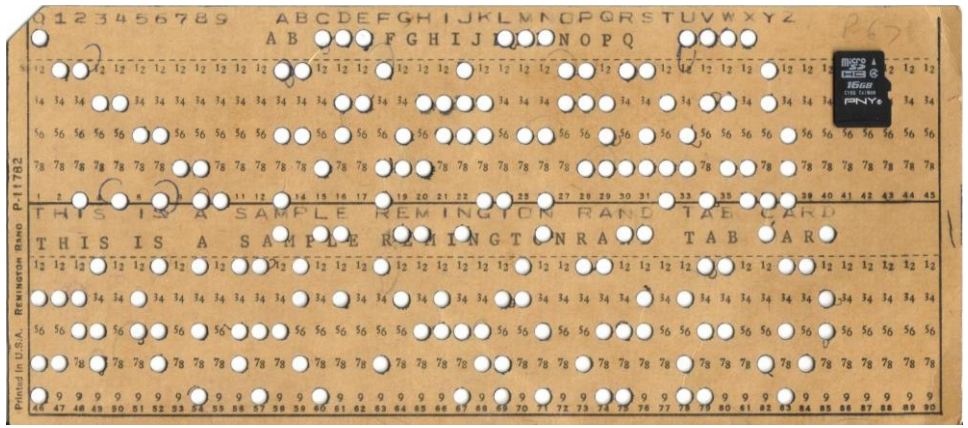


Binary Storage & Registers

The discrete elements of information in a digital computer must have a physical existence in some information storage medium. Furthermore, when discrete elements of information are represented in binary form, the information storage medium must contain binary storage elements for storing individual bits.

A binary cell is a device that possesses two stable states and can store one bit of information.

Examples of binary cells are electronic **flip-flop circuits**, ferrite cores used in memories, and positions punched with a hole or not punched in a card.



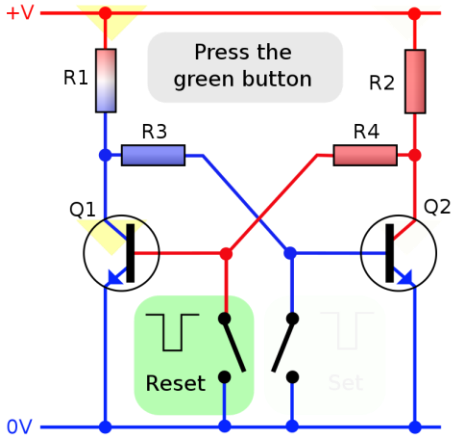
Punched Card: a piece of stiff paper that holds digital data represented by the presence or absence of holes in predefined positions

Binary Storage & Registers

The discrete elements of information in a digital computer must have a physical existence in some information storage medium. Furthermore, when discrete elements of information are represented in binary form, the information storage medium must contain binary storage elements for storing individual bits.

A binary cell is a device that possesses two stable states and can store one bit of information.

Examples of binary cells are electronic **flip-flop circuits**, ferrite cores used in memories, and positions punched with a hole or not punched in a card.



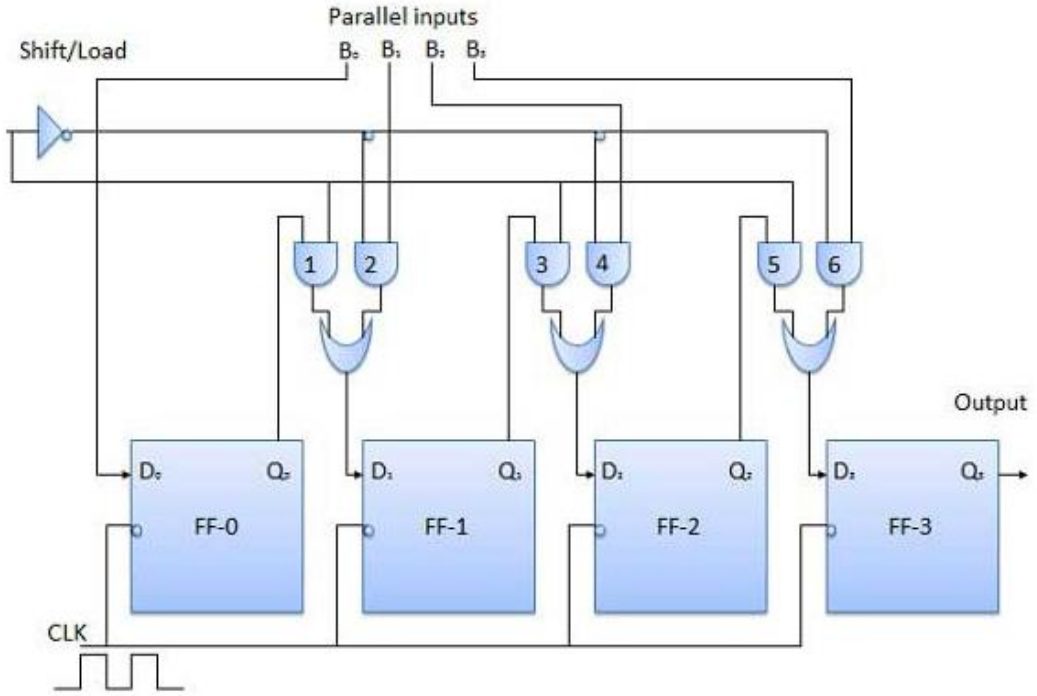
Flip-Flop Circuit (Latch): a circuit that has two stable states and can be used to store state information

Registers

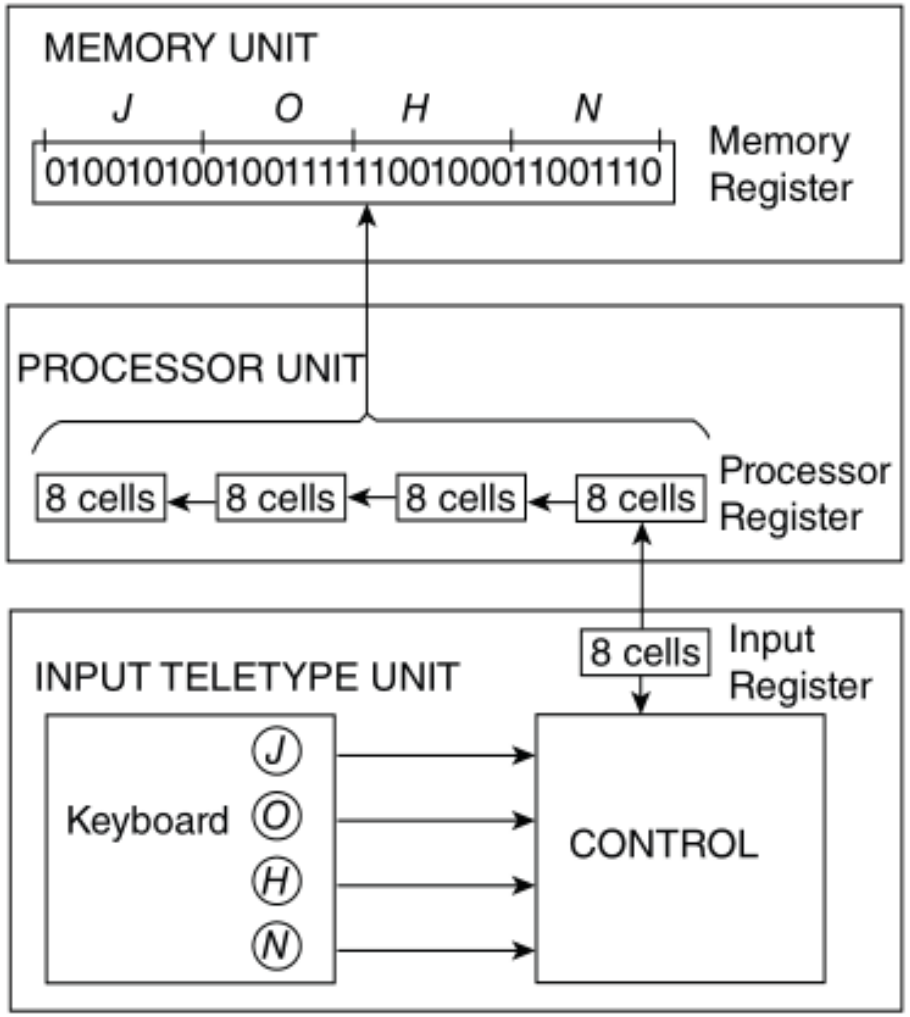
A register is a **group of binary cells**. Since a cell stores one bit of information, it follows that a register with n cells can store any discrete quantity of information that contains n bits. The state of a register is an n-tuple number of 1's and 0's, with each bit designating the state of one cell in the register. The content of a register is a function of the interpretation given to the information stored in it. Consider, for example, the following 16-cell register:

1	1	0	0	0	0	1	1	1	1	0	0	1	0	0	1
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Don't be panic if you still don't understand this picture, we will gradually learn about this step by step.



Register Transfer



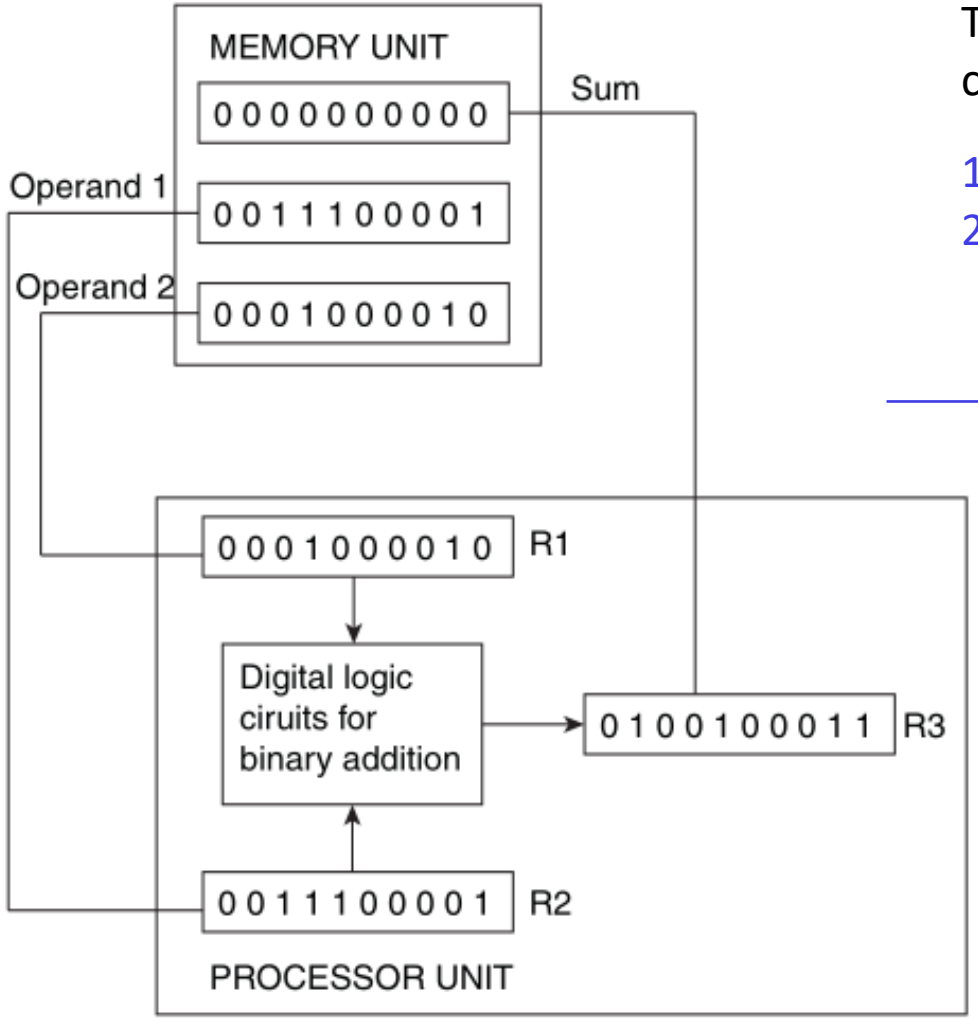
A digital computer is characterized by its registers.

The memory unit is merely a collection of thousands of registers for storing digital information.

The processor unit is composed of various registers that store operands upon which operations are performed.

The control unit uses registers to keep track of various computer sequences, and every input or output device must have at least one register to store the information transferred to or from the device.

Register Transfer: Binary information processing



To process discrete quantities of information in binary form, a computer must be provided with

1. Devices that hold the data to be processed (Register)
2. Circuit elements that manipulate individual bits of information

---

A. Memory unit stored two data but cannot do anything.

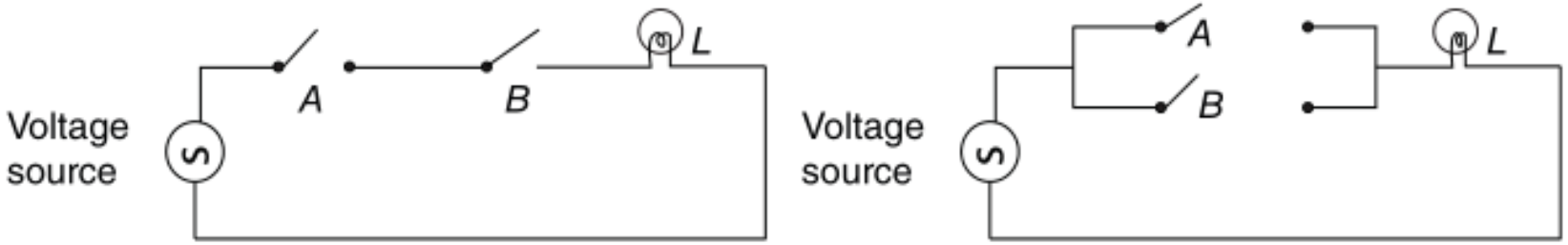
B. Processor unit received two data (registers) R1 and R2

C. Digital logic circuit do an addition then store a result to register (R3)

D. Result of summation send back to kept in Memory unit.

E. Processor unit clear registers of itself.

Binary Logic

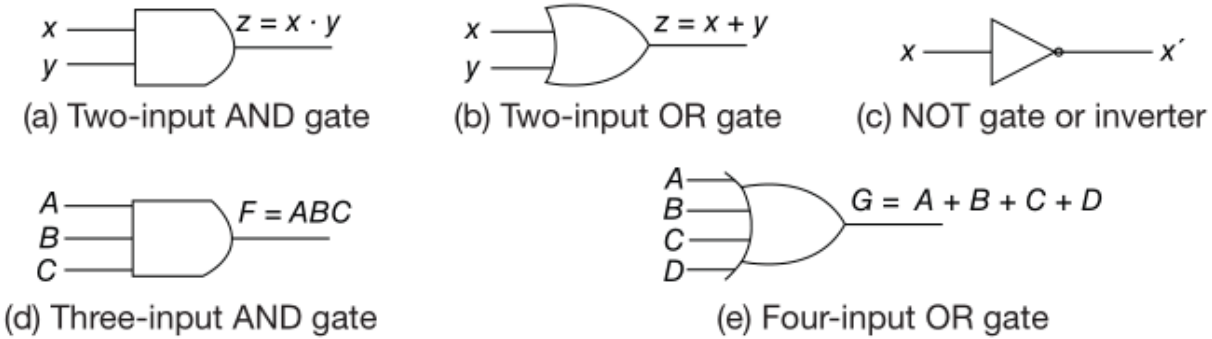


(a) Switches in series – logic AND (b) Switches in parallel – Logic OR

Logic circuits

AND			OR			NOT	
x	y	$x \cdot y$	x	y	$x + y$	x	x'
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

Truth tables of logical operations



(a) Two-input AND gate (b) Two-input OR gate (c) NOT gate or inverter

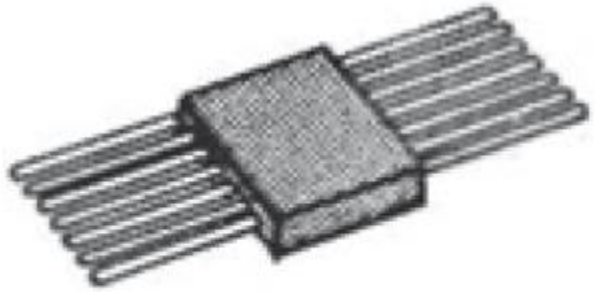
(d) Three-input AND gate (e) Four-input OR gate

Symbols for digital logic circuits

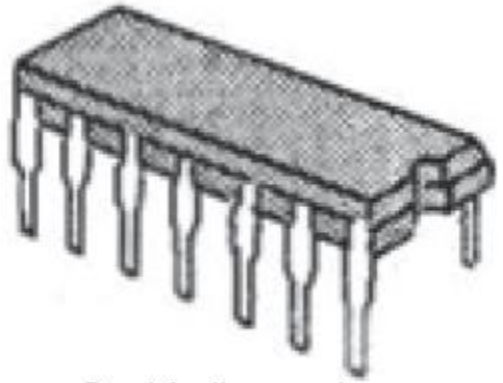
Binary Logic: Integrated Circuits (IC)

Digital circuits are invariably constructed with integrated circuits. An integrated circuit (abbreviated IC) is a small silicon semiconductor crystal, called a chip, containing electrical components such as transistors, diodes, resistors, and capacitors. The various components are interconnected inside the chip to form an electronic circuit.

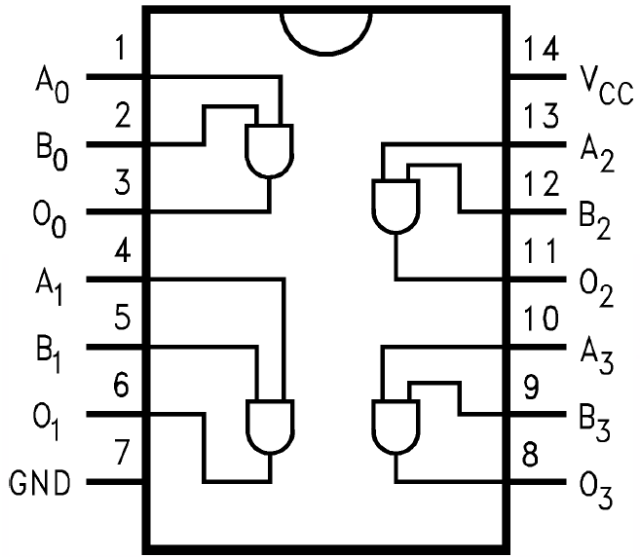
Integrated circuits differ from other electronic circuits composed of detachable components in that individual components in the IC cannot be separated or disconnected and the circuit inside the package is accessible only through the external pins.



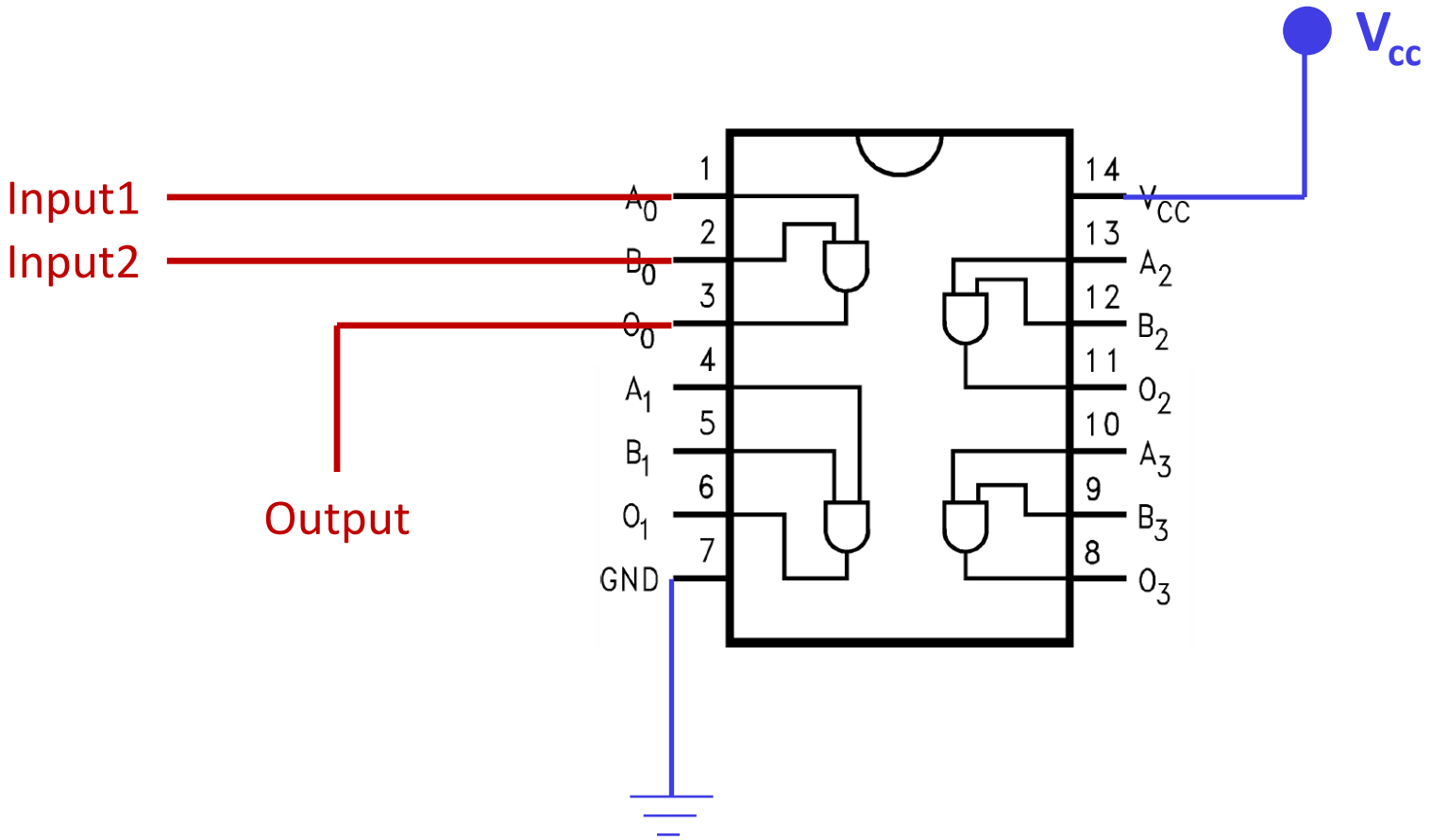
Flat package



Dual-in-line package



Binary Logic: Integrated Circuits (IC)




When you can measure what you are speaking about, and express it in numbers, you know something about it.
Lord Kelvin
quotefancy