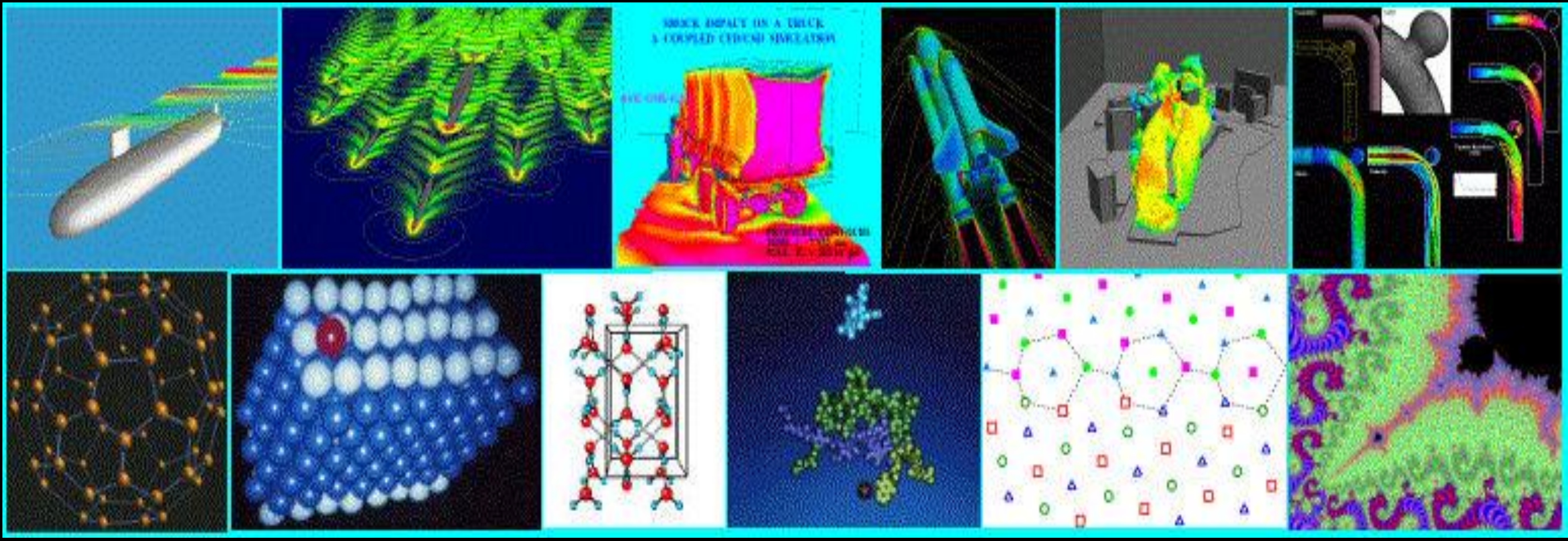
Computing for Scientists
Section I
Computer Fundamentals (CF)
(Jan. 24, 2013 - Feb. 12, 2013)



Jie Zhang

CDS 130 - 003
Spring, 2013

Computer Fundamentals (CF)

CH1. Binary Representation

- Binary numbers to/from Decimal numbers

CH2. Binary Operation

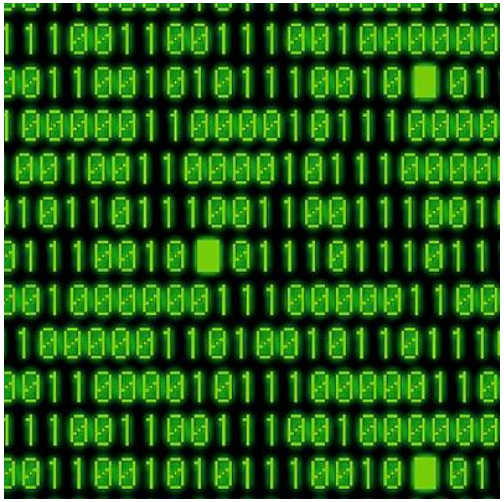
- Binary addition, subtraction, multiplication

CH3. Binary Encoding and Data Storage

- Binary numbers to/from Characters (ASCII code)

CH4. Logic Circuit and Logic Table

- Binary operation in hardware



Section 1 – Chapter 1

Computer Fundamentals: Binary Representation

(Jan. 24, 2013)

CH1. Objectives

•Understand the different ways of representing numbers

•Understand why the binary representation of numbers is important for computing machines

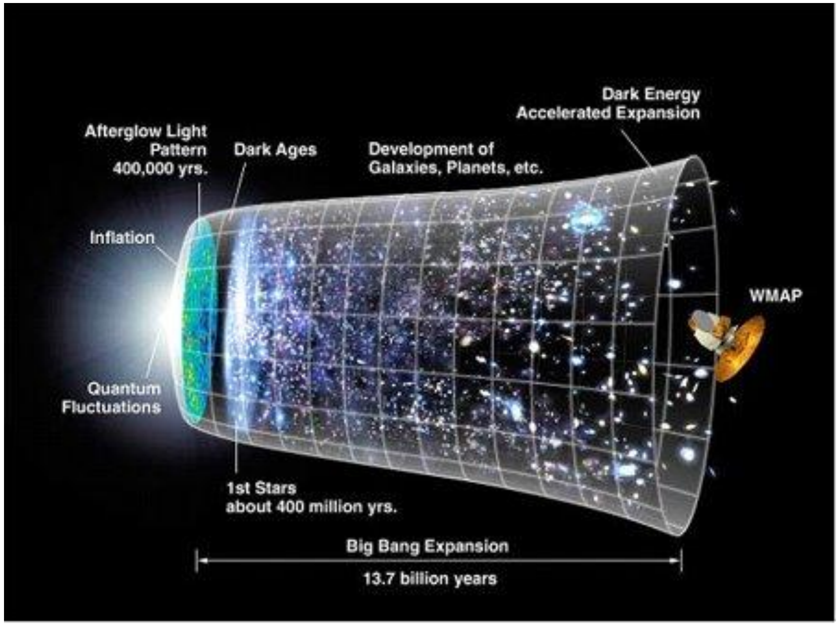
•Be able to convert from binary to decimal

•Be able to convert from decimal to binary

Power of 0 (The creation)

First God made heaven & earth. The earth was without form and void, and darkness was upon the face of the deep; and the Spirit of God was moving over the face of the waters. And God said, "Let there be light"; and there was light.
- Bible

无生有，有生二，二生三，三生万物
- The teach of Daoism



Big Bang Model of the creation of our universe
-Modern Astronomy

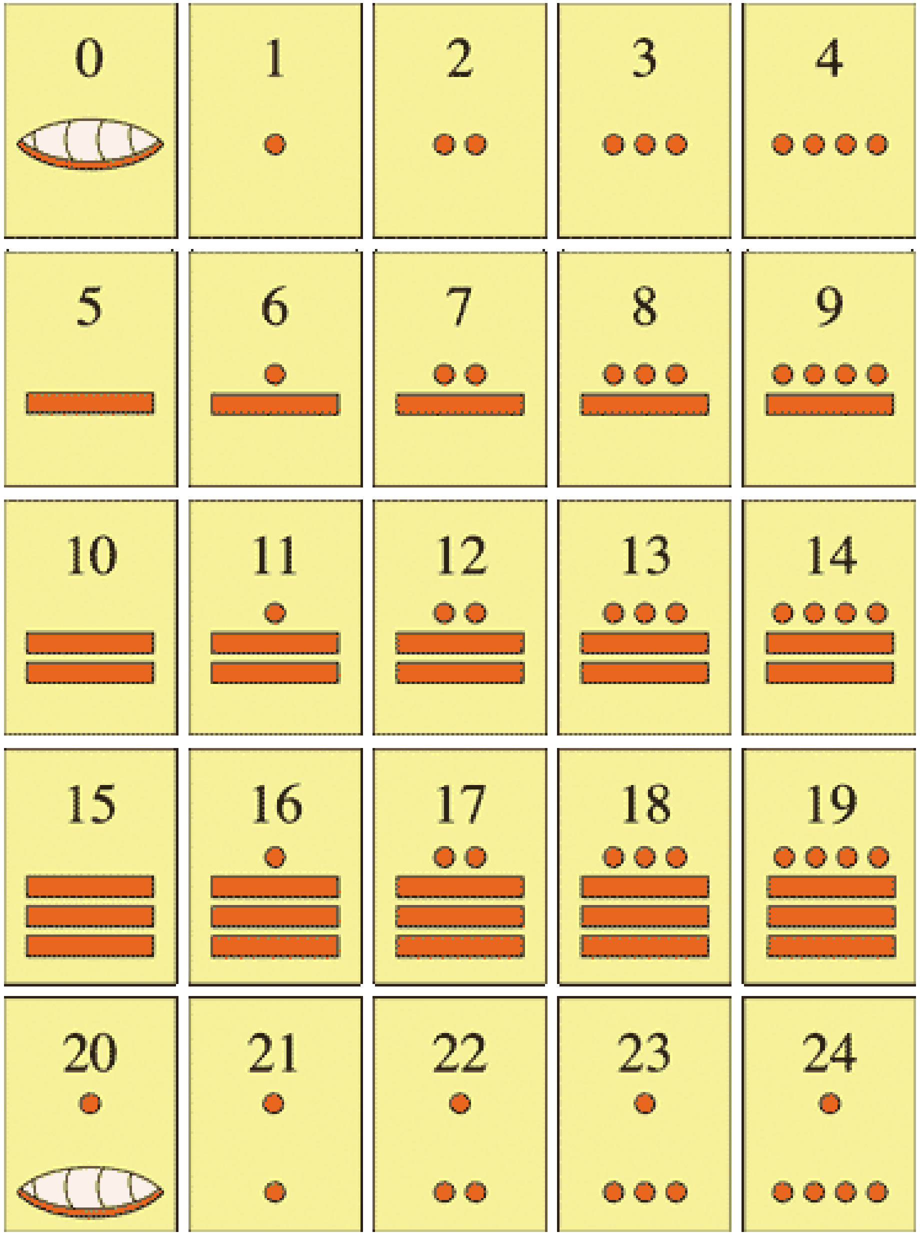
Arabic Numerals: base 10

0123456789

Arabic numerals or **Hindu numerals** are the ten digits (0, 1, 2, 3, 4, 5, 6, 7, 8, 9). They are descended from the Hindu-Arabic numeral system developed by Indian mathematicians. The Indian numerals were adopted by the Persian mathematicians in India, and passed on to the Arabs further west. From there they were transmitted to Europe in the Middle Ages. The use of Arabic numerals spread around the world through European trade, books and colonialism. Today they are the most common symbolic representation of numbers in the world

-http://en.wikipedia.org/wiki/Arabic_numerals

Mayan Numerals: base 20



©www.archimedes-lab.org

Maya Calendar:

Five digits, base-20

Re-count:
August 11, 3114 BC

December 21, 2012
(doomsday is just a one new digit in the calendar)

October 13, 4772

Decimal Numeral Numbers

- It is based on the Arabic numeral system.
- It uses **positional notation**
  - Use same symbols
  - but different orders of magnitude in different places, e.g., 1st place (ones place), 2nd place (tens place), 3rd place (hundreds place).

- Example: In decimal, 1478 means
  - 1 unit of the fourth place (1000)
  - 4 units of the third place (400)
  - 7 units of the second place (70)
  - 8 units of the first place (8)

1000+400+70+2=1478

Decimal Template

•The positional notation is represented by **power**, which relates the place and the value of one unit in that places

•**You need to know the value of one unit in that particular place**

1 unit of the first place: $10^0 = 1$

1 unit of the second place: $10^1 = 10$

1 unit of the third place: $10^2 = 100$

1 unit of the fourth place: $10^3 = 1000$

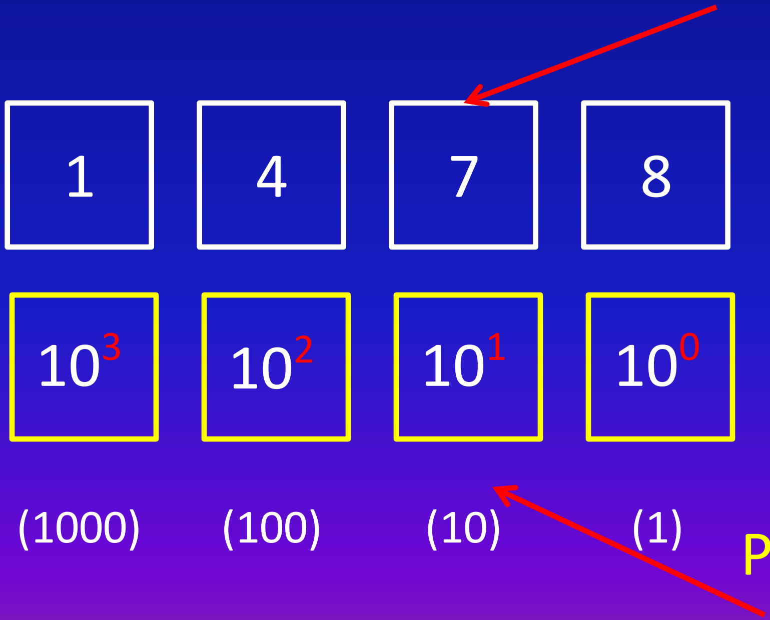
4 units of the fourth place: $4 \text{ X} 10^2 = 400$

Decimal Template

$$1 \text{ x } 10^{3} \quad + \quad 4 \text{ x } 10^{2} \quad + \quad 7 \text{ x } 10^{1} \quad + \quad 8 \text{ x } 10^{0}$$

Or, we could construct a template, like this, and just fill in the numbers :

How many units in the place



Power notation: the value of one unit in the place

Binary Numeral Numbers

- It is based on the binary representation (0, 1).

- Machines like binary representation
    - Much easier to create two states, instead of ten states

    - Open (electric current flows) versus Close (No electric current in the circuit)

    - High electric voltage versus low electric voltage

Binary to Decimal

Binary Template

			1	1	0	1
2^6	2^5	2^4	2^3	2^2	2^1	2^0
(64)	(32)	(16)	(8)	(4)	(2)	(1)

Binary Numeral Numbers

Binary	Decimal
0	0
1	1
10	2
100	4
1000	8
10000	16
100000	32
1000000	64

Decimal	Binary
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111

Let’s Do It Again!

Let’s say I have the following binary number: 1011011
Fill in the template, beginning from the right . . .

1	0	1	1	0	1	1
2^6	2^5	2^4	2^3	2^2	2^1	2^0
(64)	(32)	(16)	(8)	(4)	(2)	(1)

1 unit of 64
0 unit of 32
1 unit of 16
1 unit of 8

0 unit of 4
1 unit of 2
1 unit of 1

Again . . .

1 unit of 64
0 unit of 32
1 unit of 16
1 unit of 8
0 unit of 4
1 unit of 2
1 unit of 1

That is,

1 x 64 + 0 x 32 + 1 x 16 + 1 x 8 + 0 x 4 + 1 x 2 + 1 x 1

Or,

64 + 0 + 16 + 8 + 0 + 2 + 1

Or, 91 (written in base ten)

Practice Doing It a Few More Times

Convert the following binary numbers to base ten:

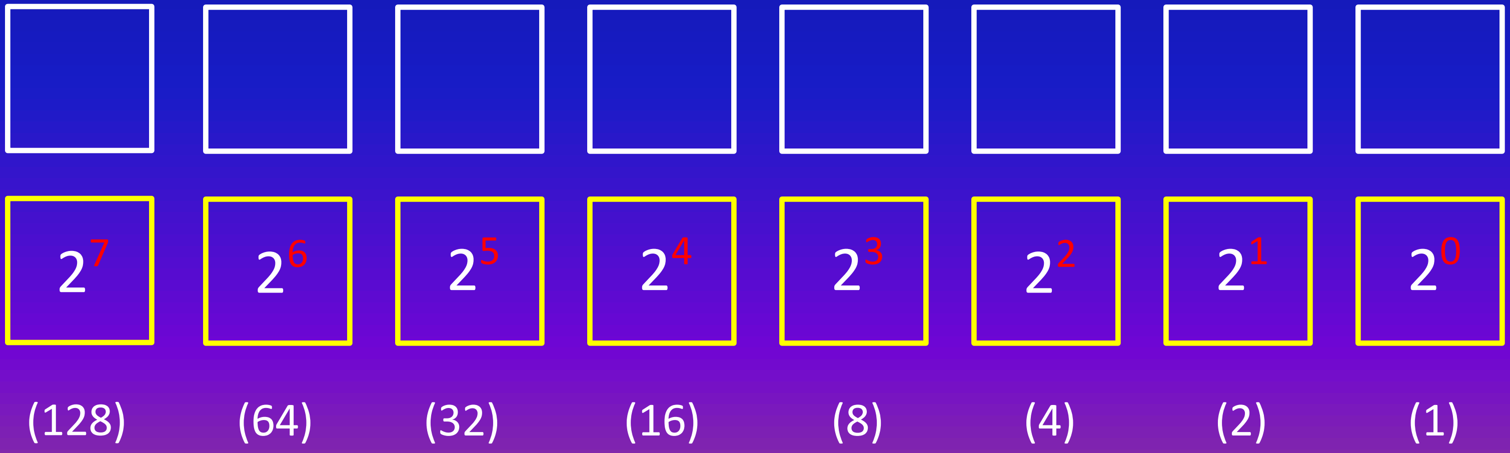
1. 11
2. 100
3. 10101
4. 100001
5. 111111

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

Practice Doing It a Few More Times -- ANSWERS

Converted binary numbers:

1. **3**
2. **4**
3. **21**
4. **33**
5. **63**



2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

Decimal to Binary

Backwards . . .

45

?

Rule: Since we're working with binary, we have ONLY TWO SYMBOLS to work with:
0 and 1 .

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

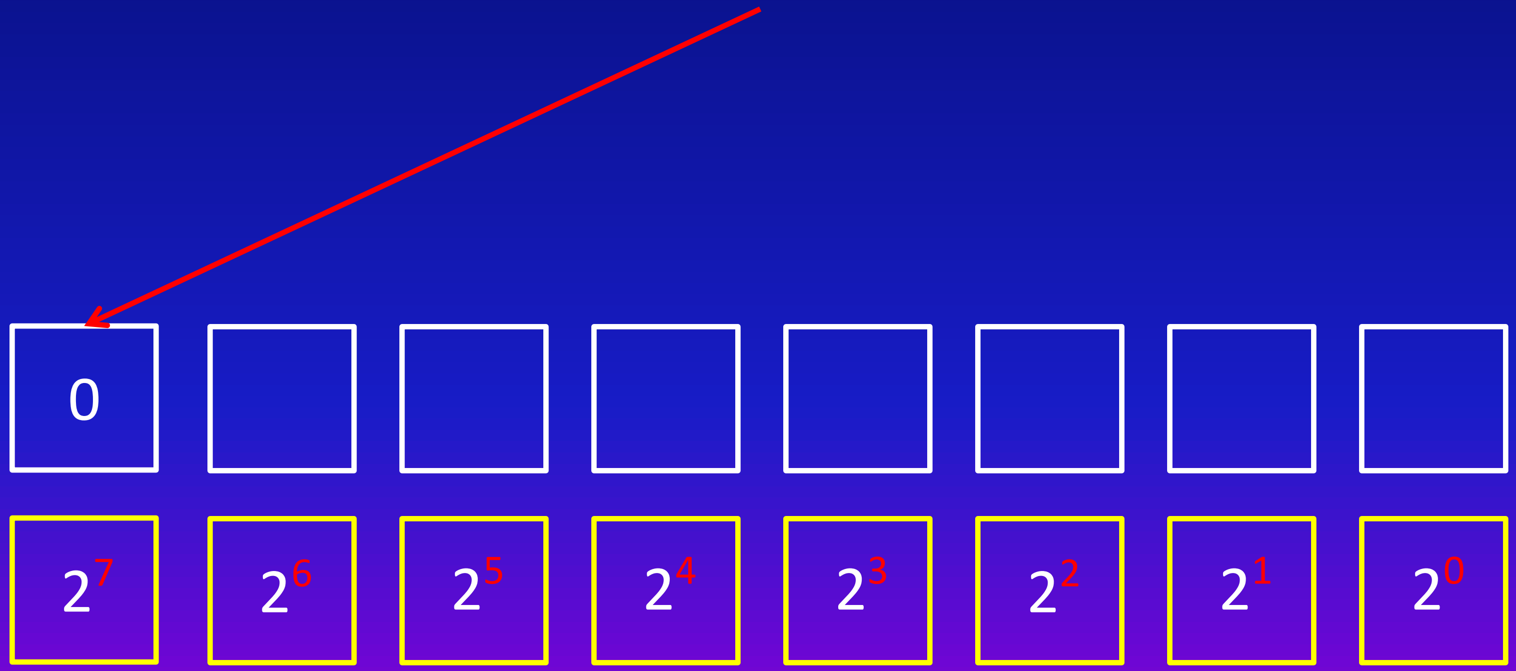
Backwards . . .

45

BEGIN FROM THE LEFT

Ask: "How many units of 128 are contained in 45?"

Zero! (None)



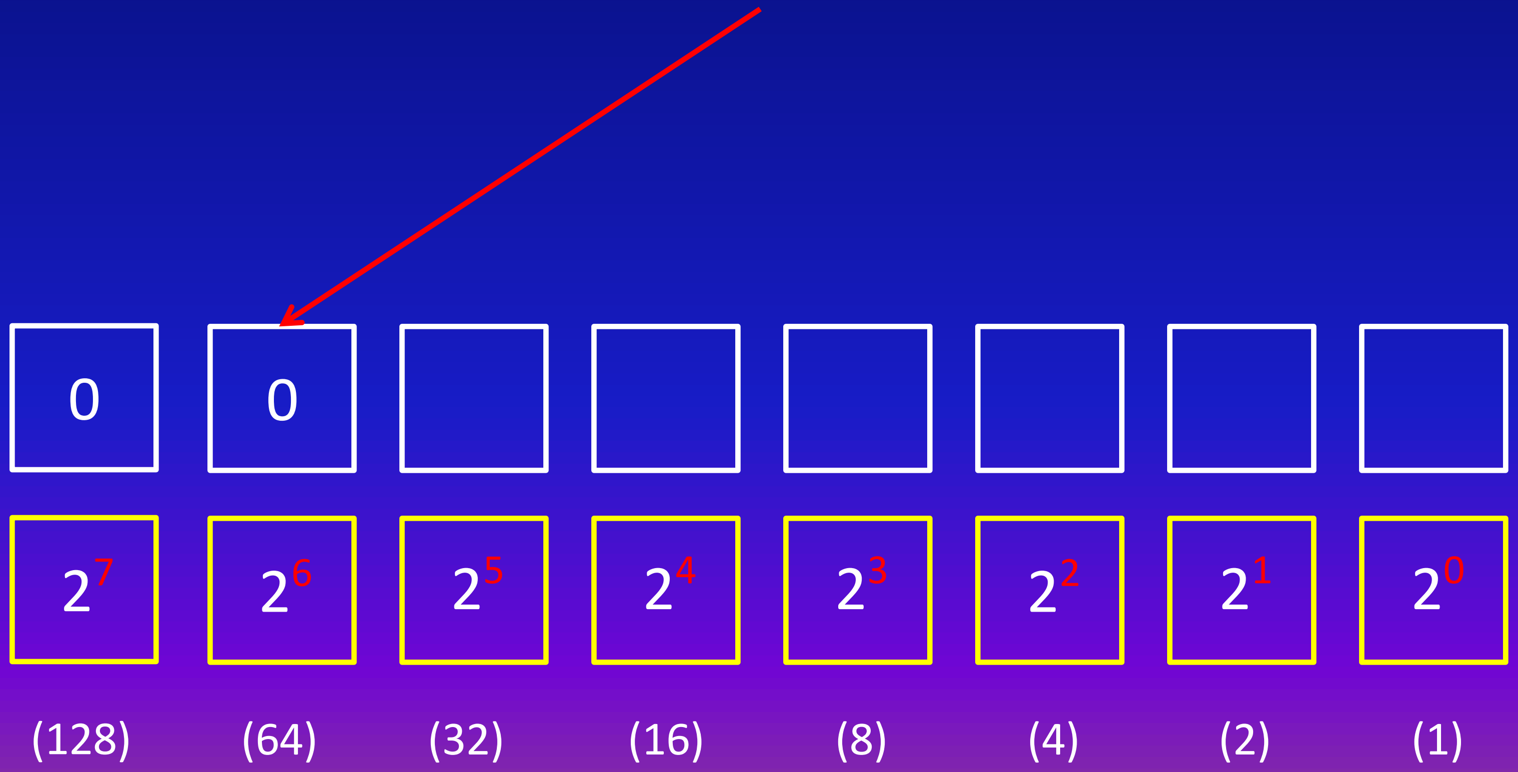
0							
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

Backwards . . .

45

Ask: “How many units of 64 are contained in 45?”

Zero! (None)



0	0						
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

Backwards . . .

$$\begin{array}{r}45\\ \underline{-32}\\ 13\end{array}$$

Ask: “How many units of 32 are contained in 45?”

One!

Then, *TAKE IT OUT . . .*

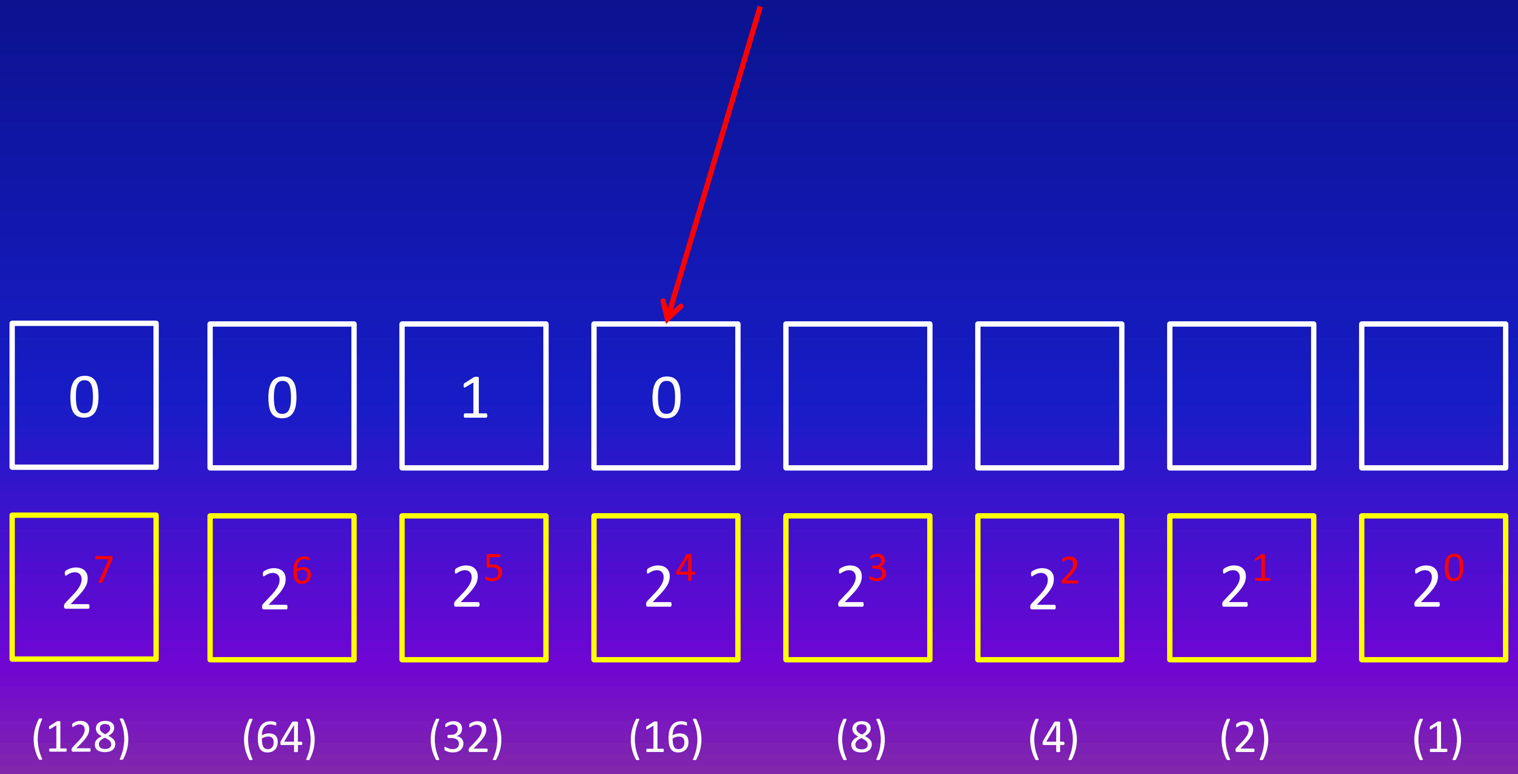
0	0	1					
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

Backwards . . .

$$\begin{array}{r}45\\ \underline{-32}\\ 13\end{array}$$

Ask: “How many units of 16 are contained in 13?”

Zero! (None)



0	0	1	0				
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

Backwards . . .

$$\begin{array}{r} 45 \\ \underline{-32} \\ 13 \\ \underline{-\ 8} \\ 5 \end{array}$$

Ask: “How many units of 8 are contained in 5?”

One!

Then, *TAKE IT OUT . . .*

0	0	1	0	1			
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

Backwards . . .

$$\begin{array}{r} 45 \\ \underline{-32} \\ 13 \\ \underline{-\ 8} \\ 5 \\ \underline{-\ \ 4} \\ 1 \end{array}$$

Ask: “How many units of 4 are contained in 5?”

One!

Then, *TAKE IT OUT* . . .

0	0	1	0	1	1		
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

Backwards . . .

$$\begin{array}{r} 45 \\ \underline{-32} \\ 13 \\ \underline{-\ 8} \\ 5 \\ \underline{-\ \ 4} \\ 1 \end{array}$$

Ask: “How many units of 2 are contained in 1?”

Zero! (None)

0	0	1	0	1	1	0	
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

Backwards . . .

$$\begin{array}{r} 45 \\ -32 \\ \hline 13 \\ -\ \ 8 \\ \hline 5 \\ -\ \ 4 \\ \hline 1 \\ -\ \ 1 \\ \hline 0 \end{array}$$

Ask: "How many units of 1 are contained in 1?"

One!

Then, *TAKE IT OUT* . . .

0	0	1	0	1	1	0	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

CONGRATULATIONS!!

45 IN BASE TEN IS 00101101 IN BASE TWO, OR, IN BINARY!

$$45_{10} = 00101101_2$$

$$45_{10} = 101101_2$$

0	0	1	0	1	1	0	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

Long Division Method

- Always divided by 2 (the divisor is always 2)
- Divide each new quotient by 2 and write the remainders to the right. Stop when the quotient is 0
- **The top remainder is the least important, and the bottom remainder is the most important.**
- Start from the bottom remainder, read the sequence of remainders upwards to the top.

- Exp: what is decimal 13 in binary?

2)13	1
2)6	0
2)3	1
2)1	1
0	

The answer: 1101

Long Division Method

For decimals: the trivial case, for illustration purpose only

10)1478   8
10)147   7
10)14   4
10)1   1
0

The answer: 1478

Long Division Method

2)45   1

2)22   0

2)11   1

2)5   1

2)2   0

2)1   1

0

The answer: $45_{10}=101101_2$

One More Time!

(Base Two, or Binary ← Base Ten)

Problem: Convert 97_{10} to binary

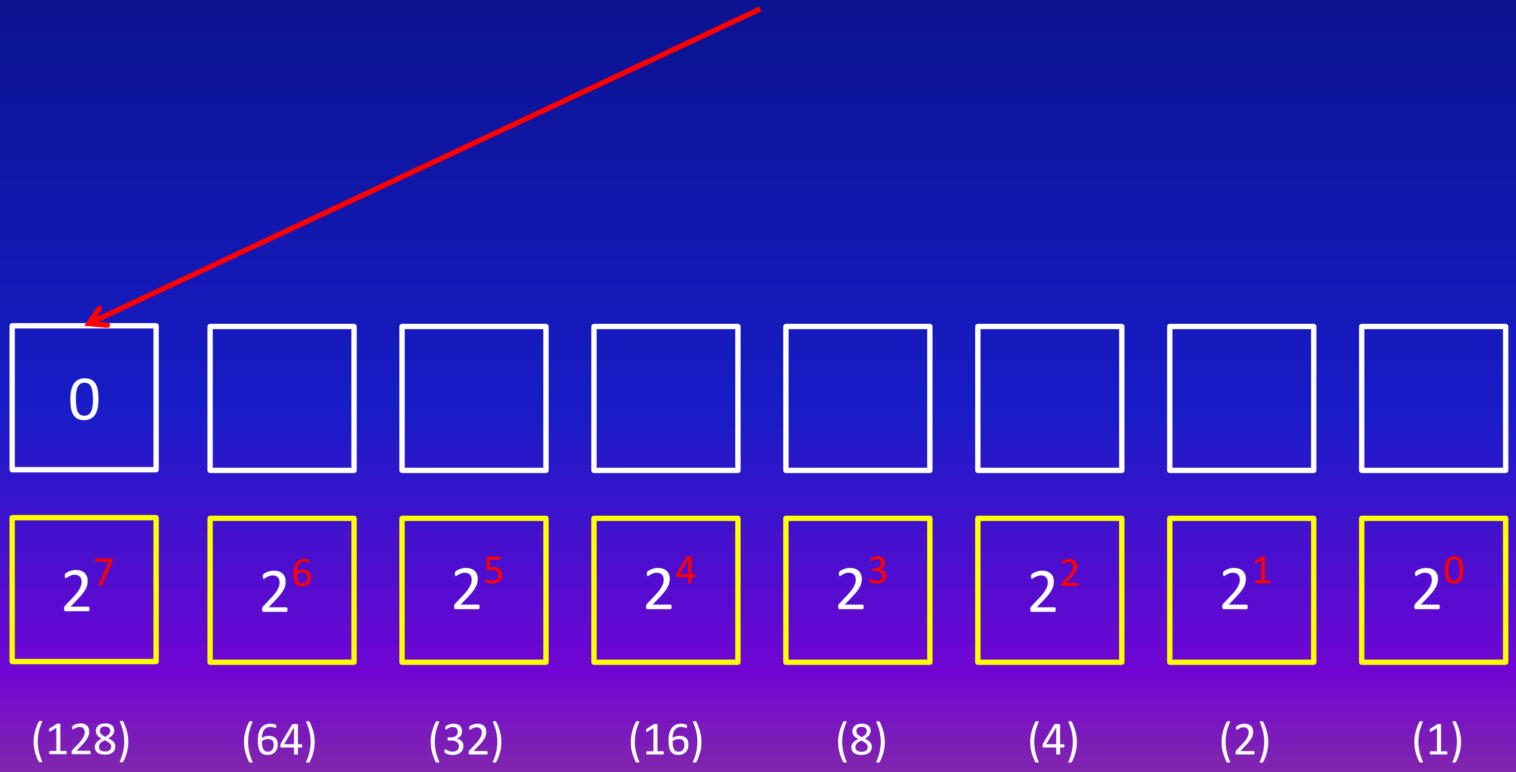
97

One More Time!

BEGIN FROM THE LEFT

Ask: “How many units of 128 are contained in 97?”

Zero! (None)



$$\begin{array}{r} 97 \\ -\ 64 \\ \hline 33 \end{array}$$

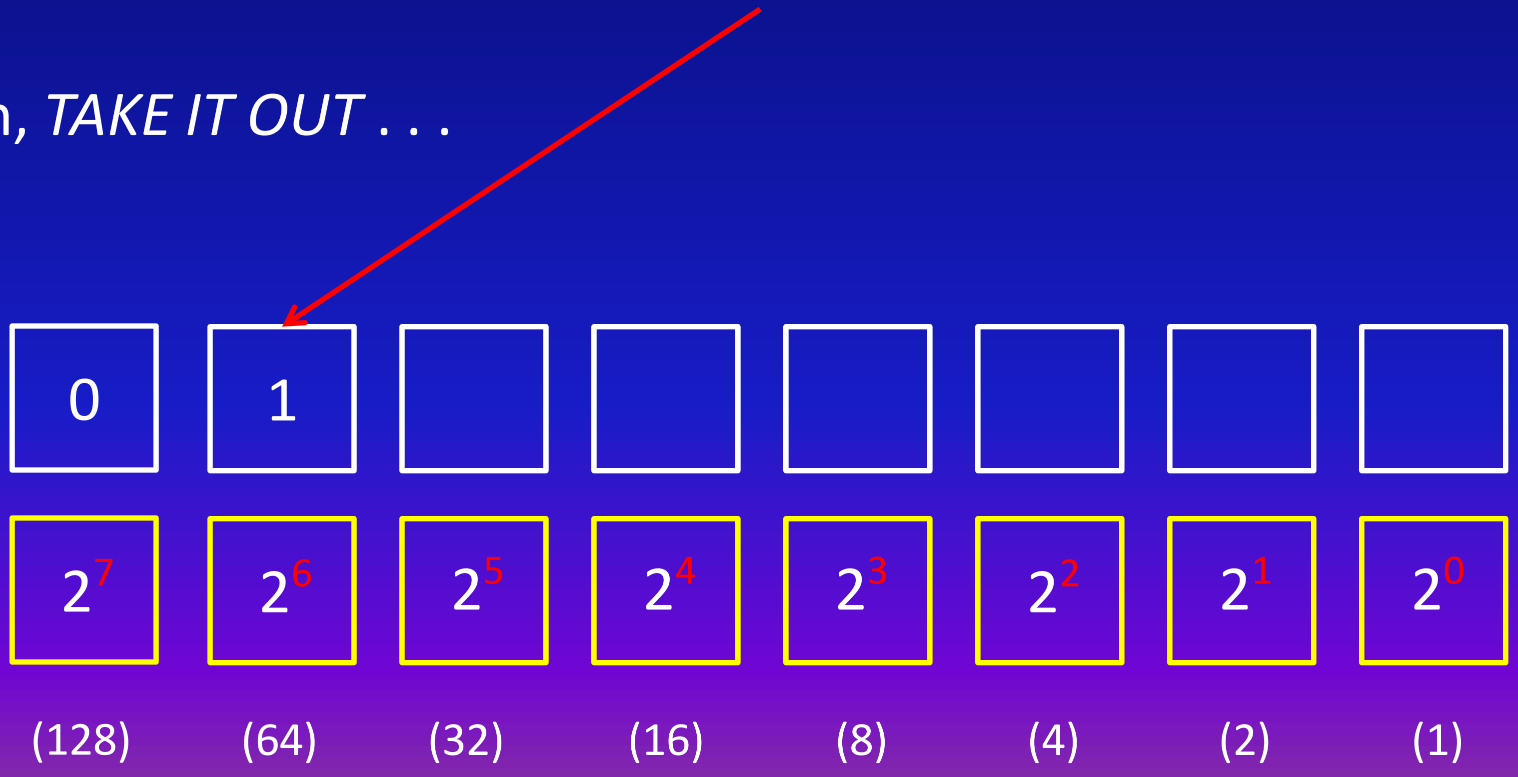
One More Time!

BEGIN FROM THE LEFT

Ask: “How many units of 64 are contained in 97?”

One!

Then, *TAKE IT OUT . . .*



0	1						
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

$$\begin{array}{r} 97 \\ \underline{-\ 64} \\ 33 \\ \underline{-\ 32} \\ 1 \end{array}$$

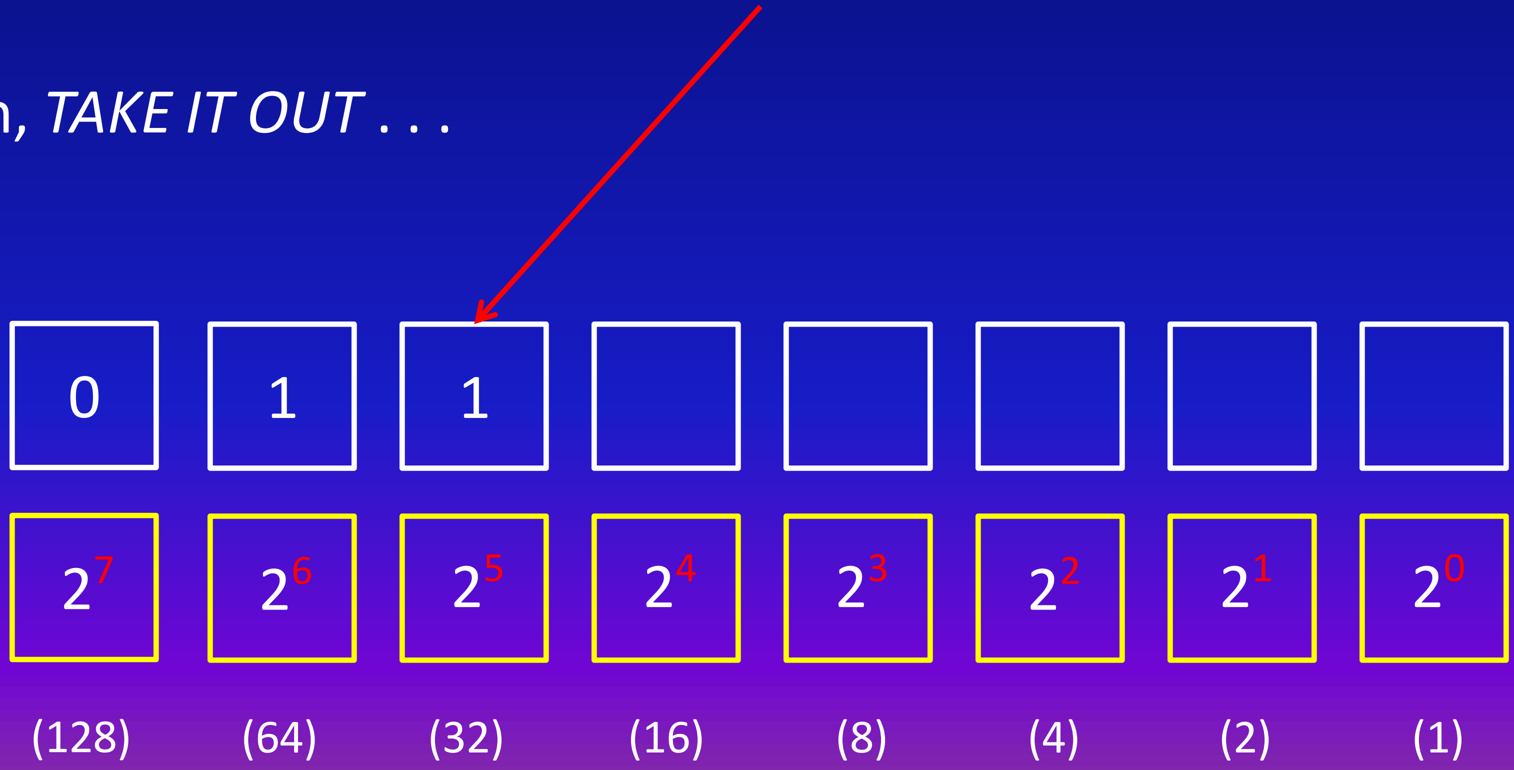
One More Time!

BEGIN FROM THE LEFT

Ask: “How many units of 32 are contained in 33?”

One!

Then, *TAKE IT OUT . . .*



0	1	1					
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

$$\begin{array}{r} 97 \\ \underline{-\ 64} \\ 33 \\ \underline{-\ 32} \\ 1 \end{array}$$

One More Time!

BEGIN FROM THE LEFT

Ask: “How many units of 16 are contained in 1?”

Zero! (none)

0	1	1	0				
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

$$\begin{array}{r} 97 \\ \underline{-\ 64} \\ 33 \\ \underline{-\ 32} \\ 1 \end{array}$$

One More Time!

BEGIN FROM THE LEFT

Ask: “How many units of 8 are contained in 1?”

Zero! (none)

0	1	1	0	0			
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

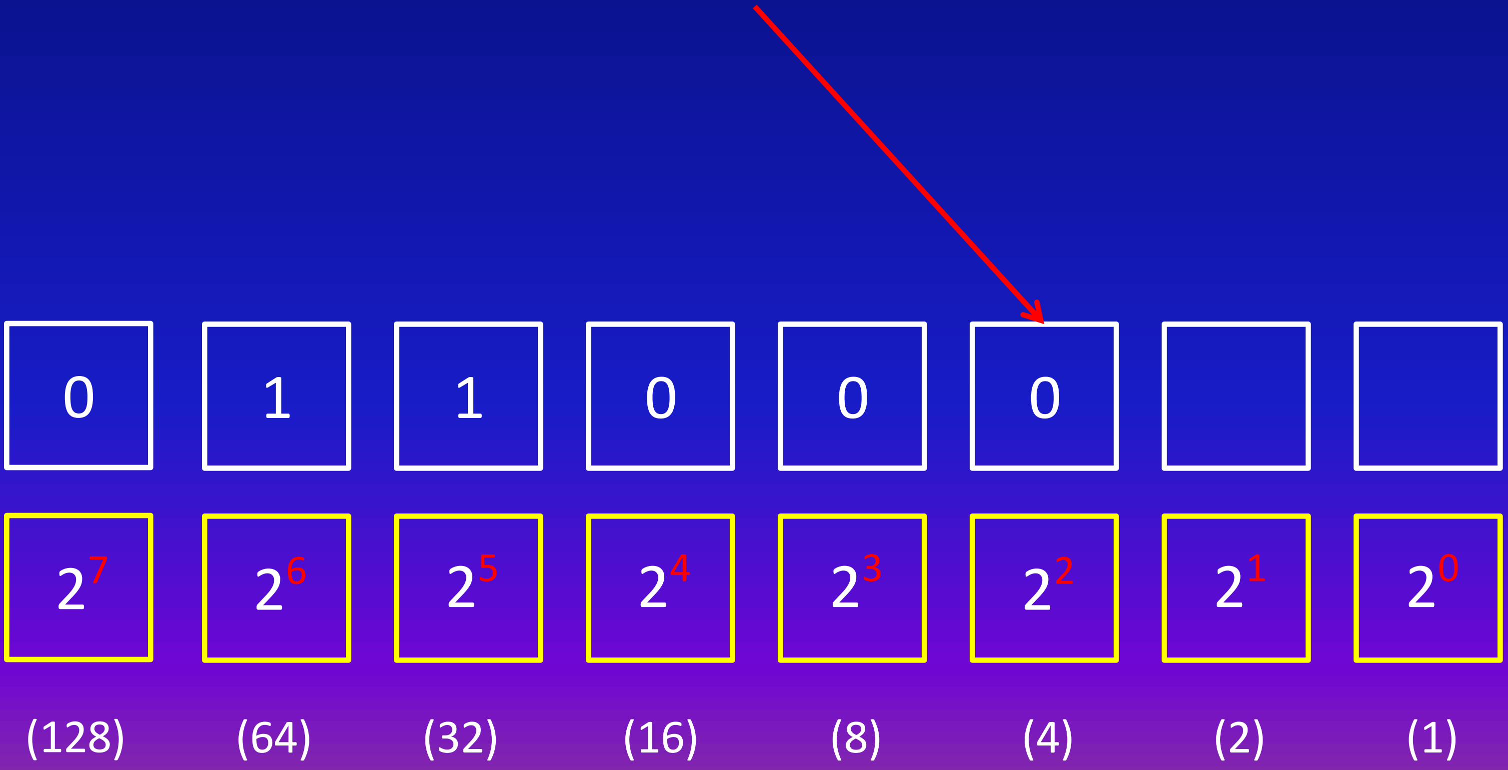
$$\begin{array}{r} 97 \\ \underline{-\ 64} \\ 33 \\ \underline{-\ 32} \\ 1 \end{array}$$

One More Time!

BEGIN FROM THE LEFT

Ask: “How many units of 4 are contained in 1?”

Zero! (none)



0	1	1	0	0	0		
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

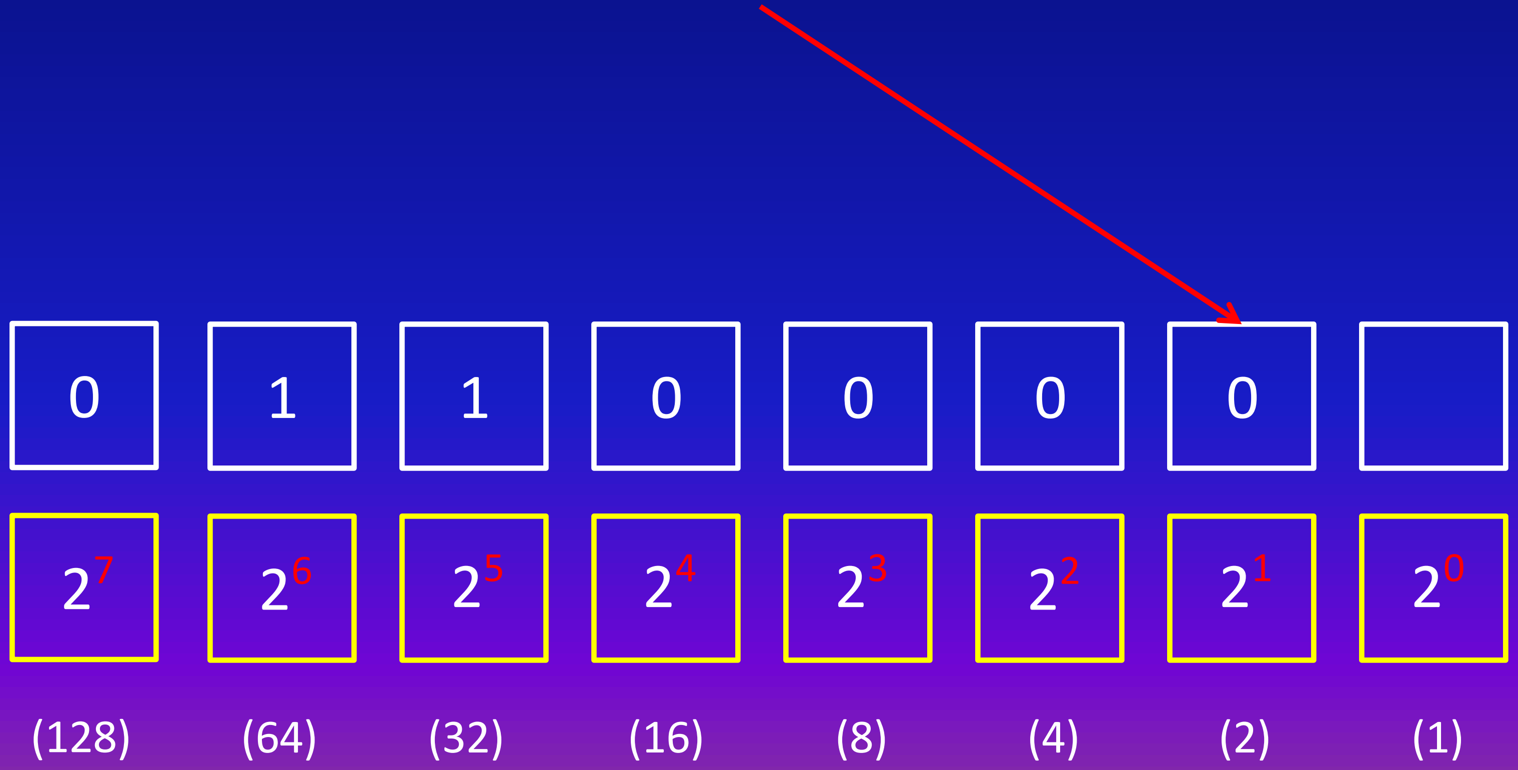
One More Time!

$$\begin{array}{r} 97 \\ \underline{-\ 64} \\ 33 \\ \underline{-\ 32} \\ 1 \end{array}$$

BEGIN FROM THE LEFT

Ask: “How many units of 2 are contained in 1?”

Zero! (none)



0	1	1	0	0	0	0	
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

One More Time!

$$\begin{array}{r} 97 \\ -\ 64 \\ \hline 33 \\ -\ 32 \\ \hline 1 \\ -\ \ 1 \\ \hline 0 \end{array}$$

BEGIN FROM THE LEFT

Ask: “How many units of 1 are contained in 1?”

One!

Then, *TAKE IT OUT* . . .

0	1	1	0	0	0	0	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

WALA!

97 IN BASE TEN IS 01100001 IN BASE TWO, OR, IN BINARY!

$$97_{10} = 01100001_{2}$$

$$97_{10} = 1100001_{2}$$

0	1	1	0	0	0	0	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

Long Division Method

$2\underline{)97}$ 1
$2\underline{)48}$ 0
$2\underline{)24}$ 0
$2\underline{)12}$ 0
$2\underline{)6}$ 0
$2\underline{)3}$ 1
$2\underline{)1}$ 1
0

The answer: 1100001

YOUR TURN…

1. Convert 17_{10} to binary
2. Convert 22_{10} to binary
3. Convert 31_{10} to binary
4. Convert 88_{10} to binary
5. Convert 168_{10} to binary

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

YOUR TURN – ANSWERS

1. Convert 17_{10} to binary: **10001 or 1 0001**
2. Convert 22_{10} to binary: **10110 or 1 0110**
3. Convert 31_{10} to binary: **11111 or 1 1111**
4. Convert 88_{10} to binary: **1011000 or 101 1000**
5. Convert 168_{10} to binary: **10101000 or 1010 1000**

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

The End of Chapter 1

(Jan. 24, 2013 Stops Here)

Jan. 29, 2013

Review

Question:

Convert binary number 11011011 to a decimal number?

Note: use the power-notation-based template method

Review

Answer:

$(1101\ 1011)_2 = (\ 219\ )_{10}$

Note: For a binary number, it is a convention to group four digits together as a sub-group

e.g. 1101 1011

e.g. 1000 0000 1001 0011

1	1	0	1	1	0	1	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

= 128 + 64 + 0 + 16 + 8 + 0 + 2 + 1

= 192 + 24 + 3

= 219

Review

Question:

Convert decimal number 251 to a binary number, using the power-notation-based template method?

Review

Answer:

$(251)_{10} = (1111\ 1011)_2$

251
-128
123
-64
59
- 32
27
- 16
11
- 8
3
- 2
1

1	1	1	1	1	0	1	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

Review

Question:

Convert decimal number 251 to a binary number, using the long division method?

Review

Answer:

$$(251)_{10} = (1111\ 1011)_2$$

2)251	1
2)125	1
2) 62	0
2) 31	1
2) 15	1
2) 7	1
2) 3	1
2) 1	1
0	

Review

How many digits do we need in the template?

$(15)_{10} = (1111)_2$
$(16)_{10} = (1\ 0000)_2$
Therefore, four digits if less than 16

$(127)_{10} = \quad (111\ 1111)_2$
$(128)_{10} = (1000\ 0000)_2$
Therefore, seven digits if less than 128

$(255)_{10} = \quad (1111\ 1111)_2$
$(256)_{10} = (1\ 0000\ 0000)_2$
Therefore, eight digits if less than 256

Section 1 – Chapter 2

Computer Fundamentals: Binary Operation

(Jan. 29, 2013)

CH2: Objectives

- Binary Addition

- Binary Subtraction

- Binary Multiplication

- (Binary Division) (omitted)

- Other numeral systems
    - Octal numeral system – 8-based
    - Hexadecimal system – 16-based

Binary Addition

Use the same principle as adding two decimal numbers

- Add the two digits
- Carry the result to the next position if necessary

Question:

Find the answer of 10110 + 00111?

Binary Addition

Answer: 10110 + 00111 = 11101

Carries

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

Check the answer:

$10110_2 = 22$

$00111_2 = 7$

$11101_2 = 29$

Binary Addition

Question:

Find the answer of 1101 + 1001?

Binary Addition

Answer: 1101+1001 = 10110

Carries

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

Check the answer:

$1101_2 = 13$

$1001_2 = 9$

$10110_2 = 22$

Binary Addition

Question (your turn):

Find the answer of 1110 + 1011?

Binary Subtraction

Use the same principle as subtracting two decimal numbers: borrow from the next place if necessary

Question:

Find the answer of 1110 - 1011?

Binary Subtraction

Answer: 1110-1011=0011

$$\begin{array}{cccccc} & & 0 & 2 & 2 & \leftarrow \text{borrowed} \\ & 1 & \cancel{1} & \cancel{1} & 0 & \\ - & 1 & 0 & 1 & 1 & \\ \hline & 0 & 0 & 1 & 1 & \end{array}$$

Check the answer:

$1110_2 = 14$

$1011_2 = 11$

$0011_2 = 3$

Binary Subtraction

Question:

Find the answer of 10110 - 00111?

Binary Subtraction

Answer: 10110-00111=01111

$$\begin{array}{rccccc} & 0 & 1 & 2 & 2 & 2 \\ & \cancel{1} & \cancel{0} & \cancel{1} & \cancel{1} & 0 \\ - & 0 & 0 & 1 & 1 & 1 \\ \hline & 0 & 1 & 1 & 1 & 1 \end{array}$$

borrowed

Check the answer:

$10110_2 = 22$

$00111_2 = 7$

$01111_2 = 15$

Binary Subtraction

Question (your turn):

Find the answer of 1101 - 1001?

Binary Multiplication

- Multiplication in binary is also similar to its decimal counterpart

Question:

Find the answer of 11 X 10 ?

Binary Multiplication

Answer: 11 X 10 = 110

$$\begin{array}{rrrr} & & 1 & 1 \\ \text{X} & & 1 & 0 \\ \hline & & 0 & 0 \\ + & 1 & 1 & \\ \hline & 1 & 1 & 0 \end{array}$$

Check the answer:

$11_2 = 3$

$10_2 = 2$

$110_2 = 6$

Binary Multiplication

Question:

Find the answer of 1101 X 1001 =?

Binary Multiplication

Answer: 1101 X 1001 = 1110101

$$
\begin{array}{rr}
 & 1\,1\,0\,1 \\
\text{X} & 1\,0\,0\,1 \\
\hline
 & 1\,1\,0\,1 \\
 & 0\,0\,0\,0 \\
 & 0\,0\,0\,0 \\
+ & 1\,1\,0\,1 \\
\hline
 & 1\,1\,1\,0\,1\,0\,1
\end{array}
$$

Check the answer: 13 x 9 = 117

1101_2 = 13
1001_2 = 9
1110101_2 = 117

Binary Multiplication

Question (your turn):

Find the answer of 101 X 010?

Other Systems of Numeration

- Octal: 8-based numeral system
  - Numerals: 0 1 2 3 4 5 6 7

- Hexadecimal: 16-based numeral system
  - Numerals: 0 1 2 3 4 5 6 7 8 9 A B C D E F

Octal Template

Similar to binary template, use 8-base instead of 2-base

Question: What is the decimal value of 456_8?

$$4 \text{ x } 8^2 \quad + \quad 5 \text{ x } 8^1 \quad + \quad 6 \text{ x } 8^0$$

$$= \quad 256 \quad + \quad 40 \quad + \quad 6 \quad = \ 302$$

4	5	6
8^2	8^1	8^0
(64)	(8)	(1)

Octal Numerals

Question (your turn):

What is the decimal value of 123_8?

Hexadecimal Numerals

Answer

$(123)_8 = (83)_{10}$

Hexadecimal Template

Similar to binary template, use 16-base instead of 2-base

Question: What is the decimal value of 456_{16}?

$$4 \times 16^{2} + 5 \times 16^{1} + 6 \times 16^{0}$$

$$= 4 \times 256 + 5 \times 16 + 6 \times 1 = 1110$$

4	5	6
16^{2}	16^{1}	16^{0}
(256)	(16)	(1)

Hexadecimal Numerals

Question (your turn):

What is the decimal value of 123_{16}?

Hexadecimal Numerals

Answer

$(123)_{16} = (291)_{10}$

Hexadecimal Numerals

Question (your turn):

What is the decimal value of FF$_{16}$?

What is the binary value of FF$_{16}$?

Hexadecimal Numerals

FF_{16} = 15 X 16 + 15 = $(255)_{10}$

$(255)_{10}$ = $(1111\ 1111)_2$

$(FF)_{16}$ = $(1111\ 1111)_2$

$(F)_{16}$ = $(15)_{10}$ = $(1111)_2$

The End of Chapter 2

(Jan. 29, 2013 Stopped Here)

Jan. 31, 2013

Review

- Binary Operations
  - Addition
  - Subtraction
  - Multiplication

- Octal Numeral System

- Hexadecimal Numeral System

Why do programmers always mix up Halloween and Christmas?

Because Oct 31 equals Dec 25.

Section 1 – Chapter 3

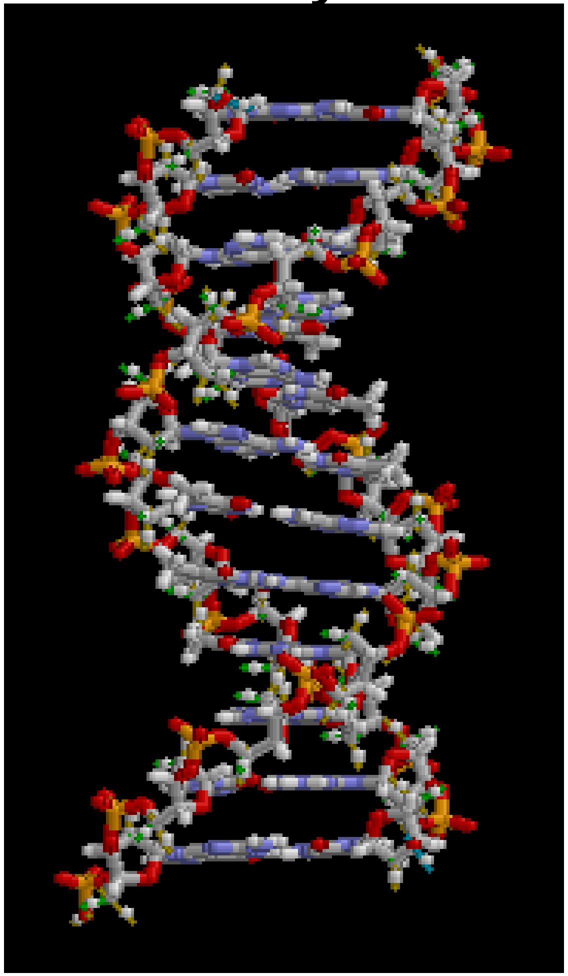
Computer Fundamentals: Binary Encoding and Data Storage

(Jan. 31, 2013)

Objectives

- Understand how a computer stores binary data and data

- Understand the data encoding



DNA: deoxyribonucleic acid

Storing Data

There are many ways to store binary data

- On papers with holes (punch card)
- On papers with inks (scantron)
- Using magnetized material
  - Magnetic tape (audio, video, data)
  - Computer Hard Disk Drive (HDD)
- Storing data as “pits” or “lands” on aluminum sheet (CD-ROM)
- Using electronic charge-coupled device (CCD)

HDD

CCD

Storing Data (In Computers)

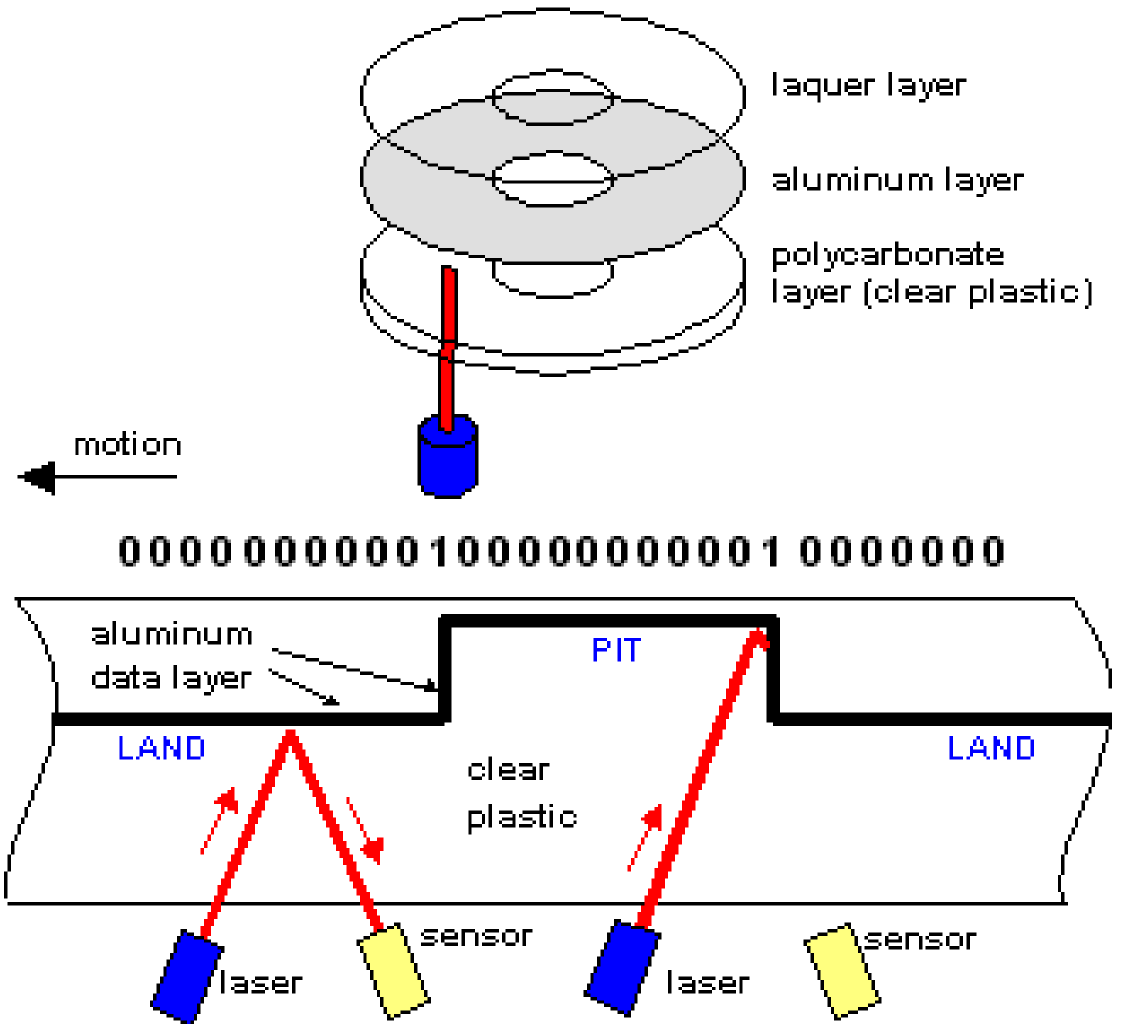
- Using silicon-based semi-conductor material
  - Building block: transistors (details in next class)
  - For CPU (Central Processing Unit)
  - For RAM (Random Access Memory)
  - For flash memory (memory stick)
    - Flash memory is not volatile, keeping the state of the data when power is off. It is portable (~1998)

CPU

RAM

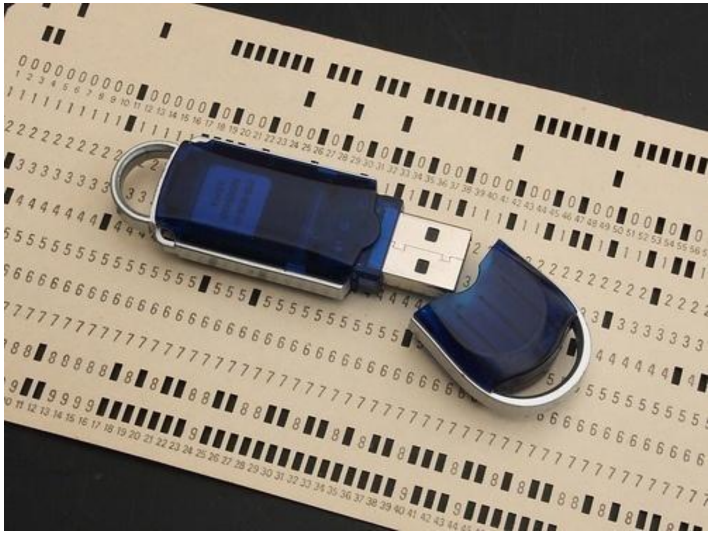
How CD-ROM work?

From Computer Desktop Encyclopedia
© 1998 The Computer Language Co. Inc.



Data Density (character/inch^2)

1. **Human readable**
2. **Machine readable**
3. **Electronic computer readable**



1. Standard A4 paper, 8 inch by 11 inch, 50 lines and 30 columns (2nd century)

2. Standard IBM punch card, 7-3/8 inch by 3-1/4 inch, 80 columns (~1950)

3. 1 GB memory stick, 2 inch by 0.5 inch (~2000)

Data Density

Answer:

Paper: 17 characters per square inch

Scantron: 3.3 characters per square inch

Memory stick: one billion characters per square inch

Data Density

Question:

How many books does an 1 GB memory stick hold?

Assuming, on average, one book has 200 pages, and 1500 characters on each page.

Data Density

Answer:

3300 books

Bits

- Bits are the individual zeros and ones that are stored by computer hardware
- A Bit can have two different states (0 or 1), and is a single digit long.
- Any system that has two states can be thought of as one bit:
    - 0 or 1
    - off or on
    - yes or no
    - true or false
    - high or low
    - open or closed

Bytes

- A Byte is a group of eight Bits grouped together.

- One Byte (8 Bits) can be used to represent:
  - Numbers from 00000000 to 11111111 in binary
  - Numbers from 0 to 255 in decimal
  - Numbers from 00 to FF in Hexadecimal

Other Types

Short Integer: 8 bits (max: 255 = 256 - 1)
Integer: 16 bits (max: 65535 = 256 X 256 - 1)
Long Integer: 32 bits (max = 65536 x 65536 -1)

Floating Point: 32 bits (max: ~ $10^{38.53}$)
Double Floating Point: 64 bits (max: ~ $10^{308.3}$)

ASCII Character: 8 bits

Bit Patterns

Given **N bits**, there are **2^N different ways** to write these N bits. That is, there are **2^N different BIT PATTERNS**.

Question:

Given 7 bits, how many different patterns can it represent?

Bit Patterns

Answer: 128

2^7=128 different patterns

Bit Patterns

Question:

What is the largest number an 8-bit binary number can represent?

Bit Patterns

Answer: 255

2^8=256 different patterns.

However, since the integer set has a zero in it, the maximum number is 256-1=255.

Question?

Explain in basic terms the meaning of the following: "The old monitor only supports 8-bit color, my monitor supports 24-bit color".

Bit Patterns

Answer:

8-bit monitor: support 256 different colors

$2^8 = 256$

24-bit monitor: support 1.7 million different colors, the true color

$2^{24} = 256 X 256 x 256 = 1677216$

Encoding

One of key ideas of understanding computing

•Encoding text involves transforming it into an "encoded" number (e.g., ASCII code)

•Music (e.g, au), video (e.g., mpg), pictures (e.g, gif) and almost everything else can be represented as "encoded" numbers

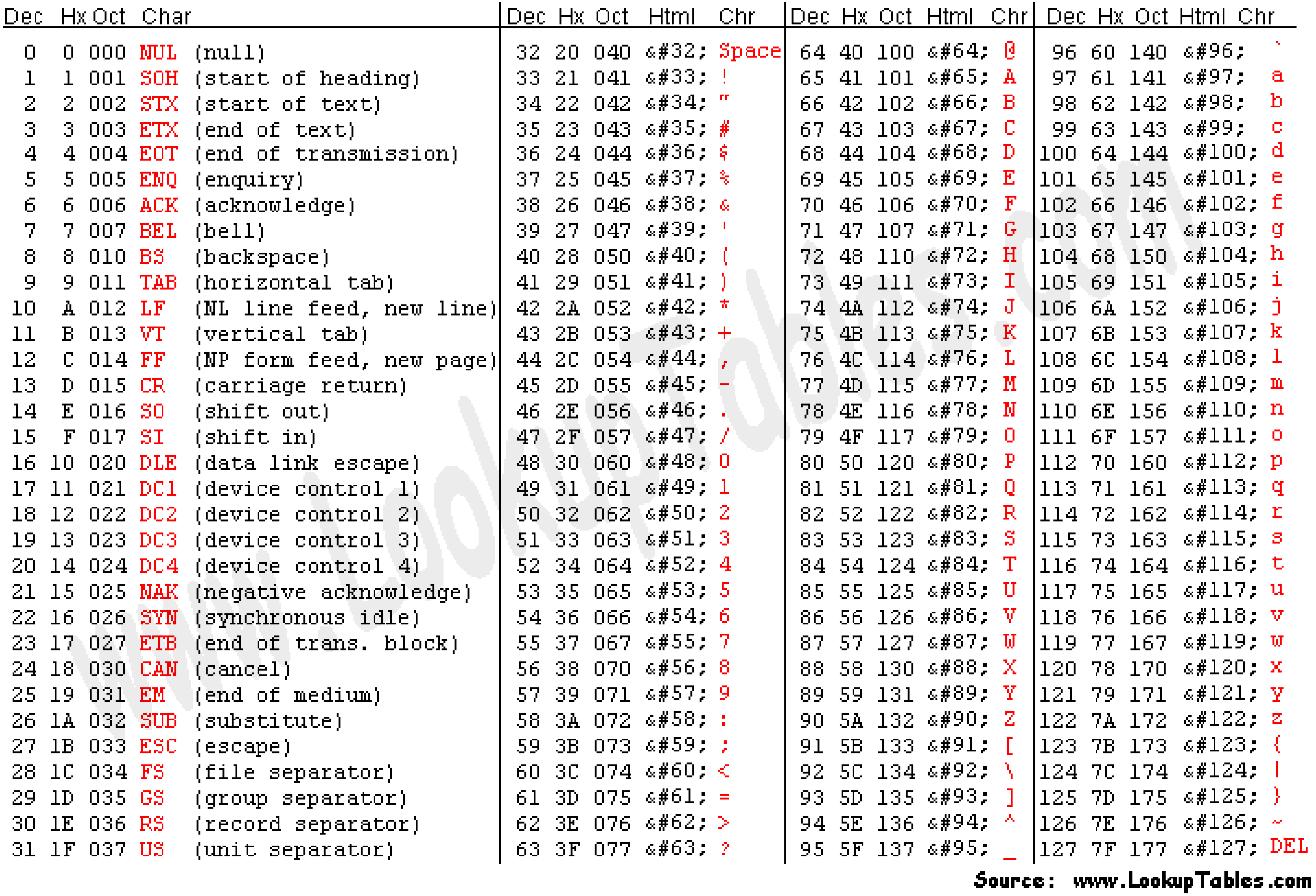
•Once the data are transformed into encoded binary sequence, it can then be stored, processed or transmitted by digital devices

ASCII Code

ASCII: American Standard Code for Information Interchange
--http://www.asciitable.com/

- When a string of characters is stored in a computer's memory, each character is given a binary number representation which is assigned to it in the ASCII Table

ASCII Table



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 Table - Extended

128	Ç	144	É	160	á	176	░	192	└	208	╨	224	α	240	≡
129	ü	145	æ	161	í	177	▒	193	┴	209	╤	225	ß	241	±
130	é	146	Æ	162	ó	178	▓	194	┬	210	╥	226	Γ	242	≥
131	â	147	ô	163	ú	179	│	195	├	211	╙	227	π	243	≤
132	ä	148	ö	164	ñ	180	┤	196	─	212	╘	228	Σ	244	⌠
133	à	149	ò	165	Ñ	181	╡	197	┼	213	╒	229	σ	245	⌡
134	å	150	û	166	ª	182	╢	198	╞	214	╓	230	µ	246	÷
135	ç	151	ù	167	º	183	╖	199	╟	215	╫	231	τ	247	≈
136	ê	152	ÿ	168	¿	184	╕	200	╚	216	╪	232	Φ	248	°
137	ë	153	Ö	169	⌐	185	╣	201	╔	217	┘	233	Θ	249	∙
138	è	154	Ü	170	¬	186	║	202	╩	218	┌	234	Ω	250	·
139	ï	155	¢	171	½	187	╗	203	╦	219	█	235	δ	251	√
140	î	156	£	172	¼	188	╝	204	╠	220	▄	236	∞	252	ⁿ
141	ì	157	¥	173	¡	189	╜	205	═	221	▌	237	φ	253	²
142	Ä	158	₧	174	«	190	╛	206	╬	222	▐	238	ε	254	■
143	Å	159	ƒ	175	»	191	┐	207	╧	223	▀	239	∩	255	

Source: www.LookupTables.com

(Jan. 31, 2013 Stopped Here)

February 05, 2013

Review

- What is a bit (1 b)?

- What is a byte (1 B)?

- How many different patterns in a 6-bit digit device?

- What is encoding?

- What is ASCII?

Two bytes meet.

The first byte asks, “Are you ill?”

The second byte replies,

“No, just feeling a bit off.”

Example of Encoding

Question:

Encoding “Hello!” to binary sequences

String broken to characters (Bytes)

Example of Encoding

Answer:

ASCII	H	e	l	l	o	!
Decimal	72	101	108	108	111	33
Binary	0100 1000	0110 0101	0110 1100	0110 1100	0110 1111	0010 0001

- There are six characters, including the exclamation mark. Note: “space” is also a character
- Look up the ASCII table, find the **decimal number** of the character
- Convert the decimal into **8-bit binary**, padding “0” in the front if necessary

Exercise

Question:

Encoding “cool GMU”

Exercise

Answer:

ASCII	c	o	o	l		G	M	U
Decimal	99	111	111	108	32	71	77	85
Binary	0110 0011	0110 1111	0110 1111	0110 1100	0010 0000	0100 0111	0100 1101	0101 0101

- There are 8 characters, including one space

Exercises

Question: You are asked to decode a binary sequence to ASCII characters.

What is 01001010 01011010?

This is a process of decoding, an inverse process of encoding.

Exercises

1. **Pair students to form 2-person groups**
2. **On a piece of paper, encoding your name initial to a binary sequence, without revealing your name**
3. **Give the paper to your partner, and ask your partner to decode the binary sequence into ASCII characters.**
4. **Cross-check the answers. Make sure that all of you get the right answer**

The End of Chapter 3

Section 1 – Chapter 4

Computer Fundamentals: Logic Circuits and Logic Tables

(February 05, 2013)

Motivation

So far, we have learned how binary numbers work arithmetically.

Now we need to learn how a machine operates on binary numbers

Objectives

- Transistors - the building block of modern electronic device - **“Ubiquitous”**

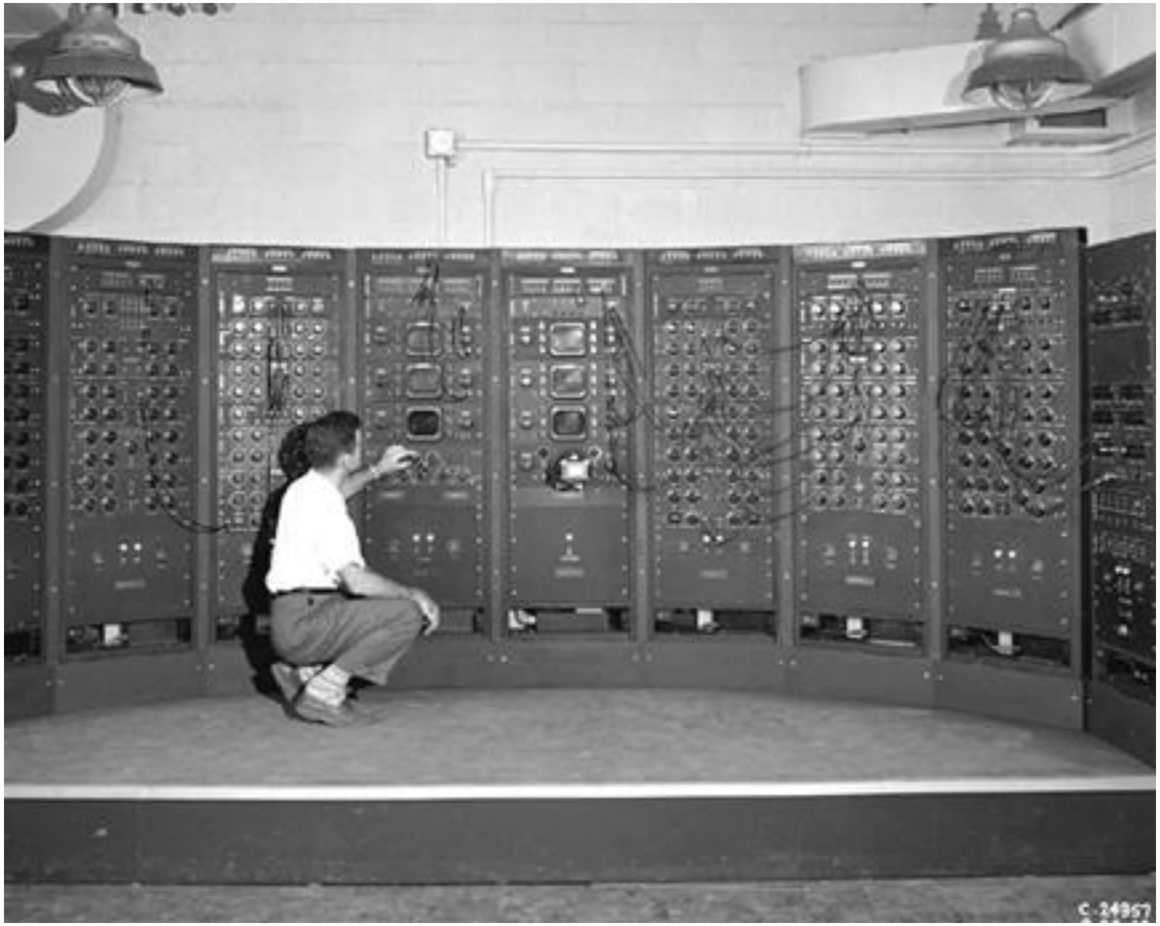
- Explain what a logic circuit is.

- Explain what a logic table is.

- Explain how logic circuits can be combined to manipulate binary numbers and do calculations.

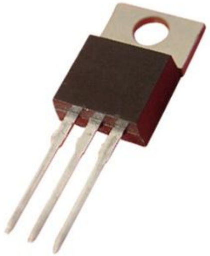
Transistor: Prior Digital Era

Vacuum Tube – Old fashioned analog transistor to amplify electric signals



Analog Computing Machine (1949)
http://www.centennialofflight.gov/essay/Evolution_of_Technology/Computers/Tech37G1.htm

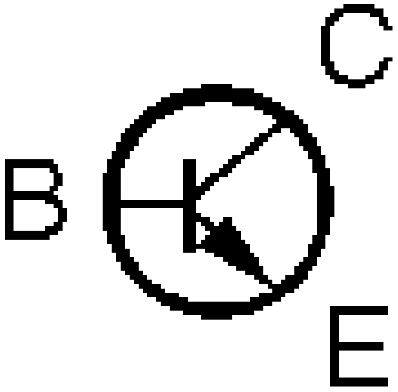
Transistors



- A transistor is a semiconductor device used to amplify and switch electronic signals

- Semiconductor: a material has dual properties of a pure conductor and a pure insulator. It is foundation of modern electronic device.

- A transistor has three pins
  - Collector (C): acting as source
  - Base (B): acting as gate
  - Emitter (E): acting as drain



Transistors

“About 60 million transistors were built this year [2002] ... for [each] man, woman, and child on Earth.”

--

http://en.wikipedia.org/wiki/Transistor#cite_note-11

Moore’s Law

Moore's law is the observation that over the history of computing hardware, the number of transistors on integrated circuits doubles approximately every two years.

Gordon E. Moore, Intel co-founder (1965)

Question:

Estimate the number of transistors built for each person on the Earth in 2012?

(February 05, 2013 Stops Here)

February 07, 2013

Review

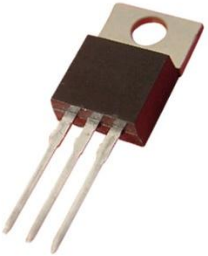
- Encoding
- Decoding
- Transistor (to be continued)

Review

Question:

Encoding “CDS 130”

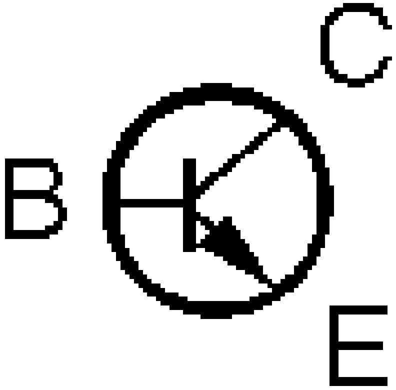
Transistors



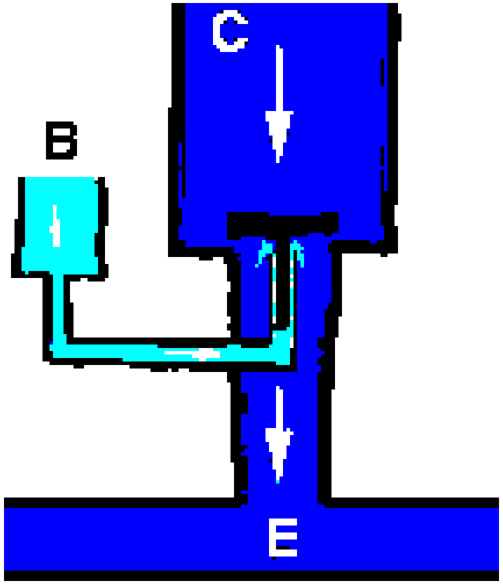
- A transistor is a semiconductor device used to amplify and switch electronic signals

- Semiconductor: a material has dual properties of a pure conductor and a pure insulator. It is foundation of modern electronic device.

- A transistor has three pins
  - Collector (C): acting as source
  - Base (B): acting as gate
  - Emitter (E): acting as drain



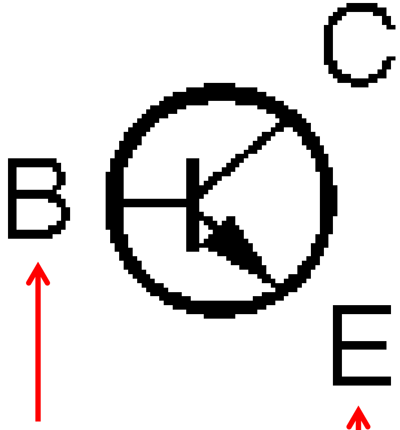
The Hydraulic Analogy



http://www.satcure-focus.com/tutor/page4.htm

- It is often easier to think of electronic current flow in terms of water flow.

- A small water flow from B pushes the black plunger upward, allowing a large flow from C to E. This is the amplification.



Main Power

Input: Small Current

Output: Large Current

Transistors

- When a small current is applied between Base and Emitter (bit 1), a large current is generated between Collector and Emitter; the gain or amplification is about 100 times.

- **The large current is necessary for cascading the state through other transistors for complex calculations.**

Transistors

- Small size and minimal weight, allowing the development of miniaturized electronic devices – integrated circuits

- Highly automated manufacturing processes, resulting in low per-unit cost.

- Lower possible operating voltages, making transistors suitable for small, battery-powered applications.

- Lower power dissipation and generally greater energy efficiency.

- Extremely long life. Some transistorized devices have been in service for more than 50 years.

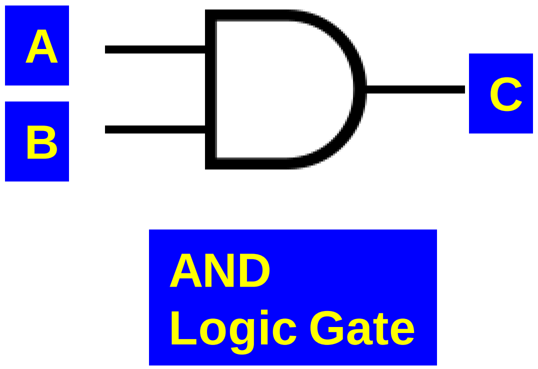
Logic Operation

Logic Circuit
Logic Table

Logic Circuit and Logic Operation

- Using transistors, we can create a logic circuit, or logic gate

- A logic circuit performs a logic operation
- e.g., AND operation: A AND B = C



Example: C= A AND B. If A is true, B is true, what is C?

Logic Circuit and Logic Operation

- It has two inputs and only one output

- Each input and output has two possible states
  - 1 or 0
  - true or false

- Logic operation is also called Boolean operation

Boolean Algebra:

AND is a multiplication

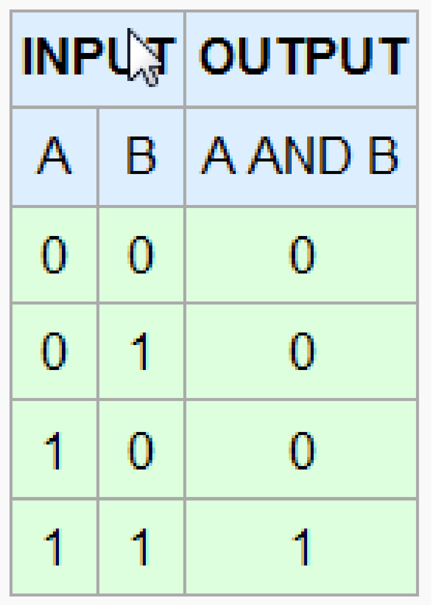
$$\boxed{C = A \cdot B}$$

Logic Table

- A logic table is used to define the inputs and output of a logic circuit or a logic operation

$$A \ \mathrm{AND} \ \mathrm{B}$$

$$A \cdot B$$



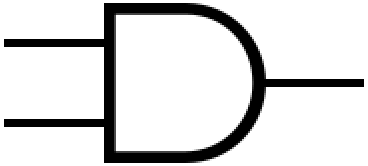
INPUT		OUTPUT
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

Boolean Algebra:

AND is a multiplication

AND operation logic table

Type - AND gate

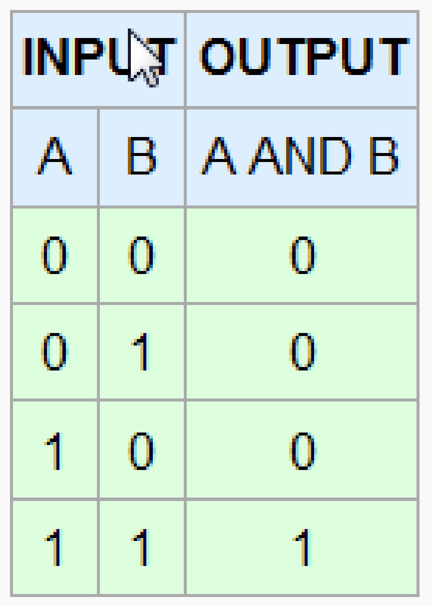


Symbol

$$A \cdot B$$

Boolean Algebra:

multiplication

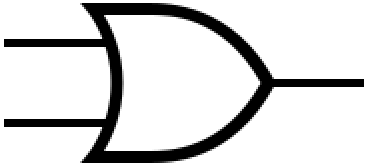


INPUT		OUTPUT
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

Logic table

- The diagram of the AND gate looks like a capital letter D with two "prongs" on the left (the inputs) and one "prong" on the right (the output).
- if either of the inputs is 0, then the output of the AND gate is 0. Thus, in order to get an AND gate to output 1, ***both*** inputs to it must be 1

Type - OR gate

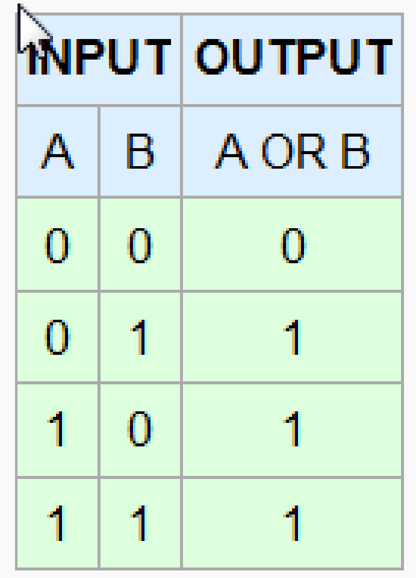


Symbol

$$A + B$$

Boolean Algebra:

plus

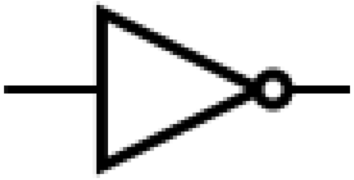


INPUT		OUTPUT
A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

Logic table

- if either of the inputs is 1, then the output of the OR gate is 1. Thus, in order to get an OR gate to output 0, ***both*** inputs to it must be 0

Type - NOT gate

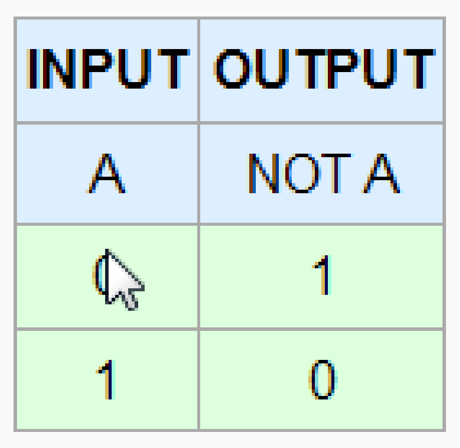


Symbol

$\overline{A}$

Boolean Algebra:

Negated,
Reversed

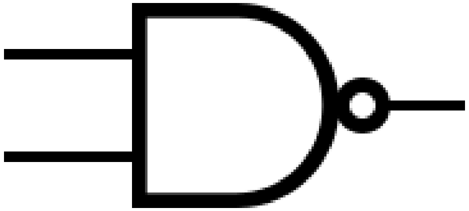


INPUT	OUTPUT
A	NOT A
0	1
1	0

Logic table

- The operation of reversing the input state

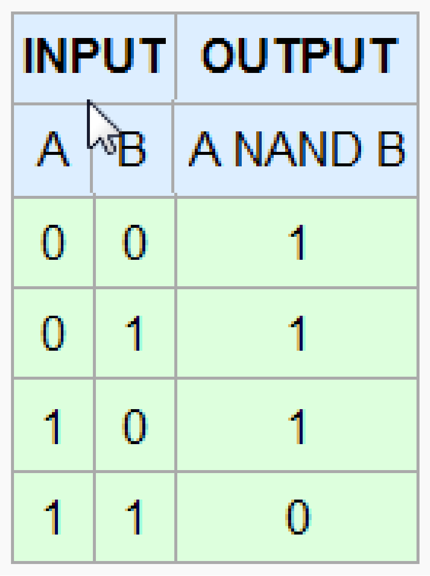
Type - NAND gate



Symbol

$$\overline{A \cdot B}$$

Boolean Algebra:



INPUT		OUTPUT
A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

Logic table

- The output of the NAND (negated AND) gate is the negation, or ***reverse*** of the output of an AND gate.

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

- Negated AND of A and B, equals OR of negated A and negated B: multiplication -> plus

Type - NAND gate

Exercise: Prove the following logic operation using logic tables

$$\boxed{\overline{A \cdot B} = \overline{A} + \overline{B}}$$

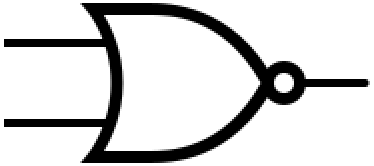
Type - NAND gate

Prove: $$\boxed{\overline{A \cdot B} = \overline{A} + \overline{B}}$$

A	B	$\boxed{A \cdot B}$	$\boxed{\overline{A \cdot B}}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

A	B	$\boxed{\overline{A}}$	$\boxed{\overline{B}}$	$\boxed{\overline{A} + \overline{B}}$
0	0	1	1	1
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0

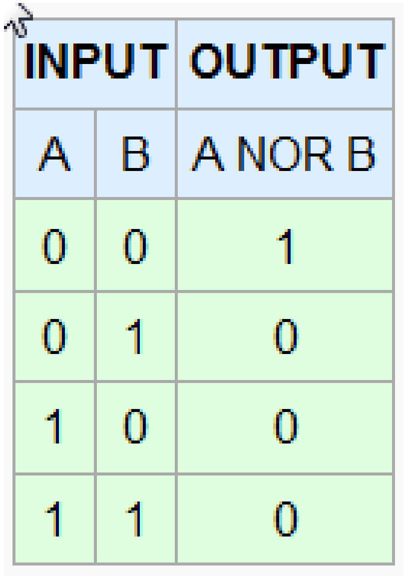
Type - NOR gate



Symbol

$$\overline{A+B}$$

Boolean Algebra



INPUT		OUTPUT
A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

Logic table

- NOR stands for "Negated OR". Thus, the output of the NOR gate is the negation, or reverse of the output of an OR gate with the same inputs.

$$\overline{A+B} = \overline{A} \cdot \overline{B}$$

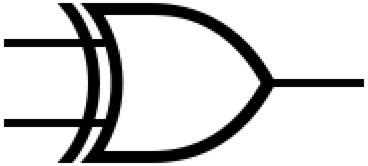
- Negated OR of A and B, equals AND of negated A and negated B: plus -> multiplication

Type - NOR gate

Exercise: Prove the following logic operation using logic tables

$$\boxed{\overline{A+B}=\overline{A}\cdot\overline{B}}$$

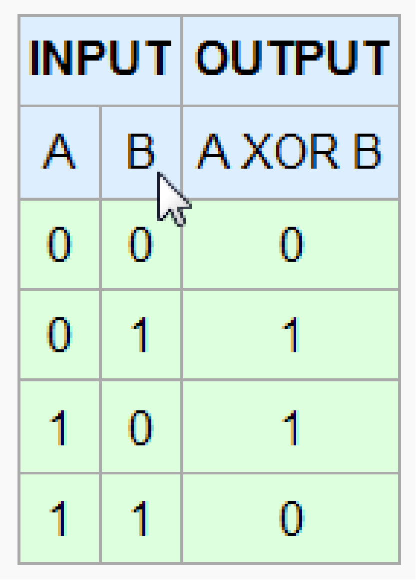
Type - XOR gate



Symbol

$$A \oplus B$$

Boolean Algebra



INPUT		OUTPUT
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Logic table

- EOR stands for "Exclusive OR". The thing to remember about EOR gates is this: An EOR gate will output 1 ***only*** if one of the inputs is 1 and the other input 0. If both inputs are the same (1 and 1, or 0 and 0), then EOR outputs 0

Logic Gate Mnemonics

MEANINGS:

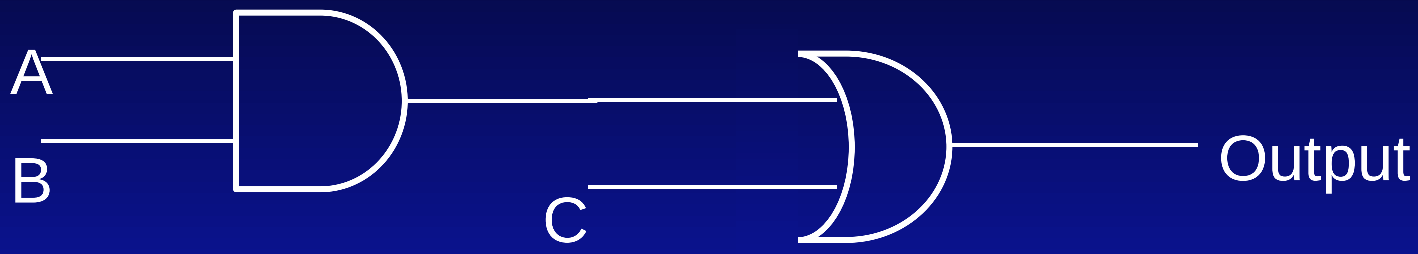
AND: If ANY input to the AND gate is 0, then it’s output is 0.

NAND: If ANY input to the NAND gate is 0, then it’s output is 1

OR: If ANY input to the OR gate is 1, then it’s output is 1

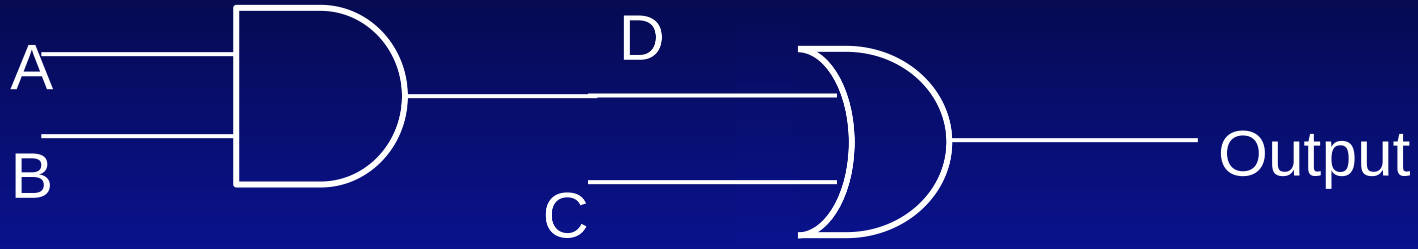
NOR: If ANY input to the NOR gate is 1, then it’s output is 0

Example: Two Gates



A	B	C	Output
0	0	0	?
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?

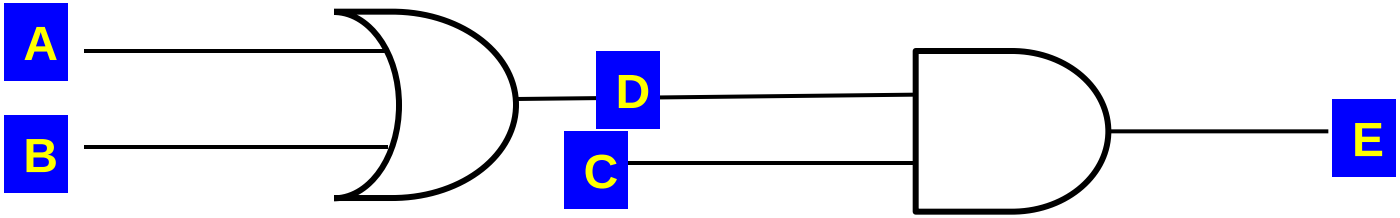
Answer



A	B	C	D	Output
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

Logic Operation

Exercise: Find the answer of the following logic circuit?



A	B	C	D	E
0	0	0	?	?
0	0	1	?	?
0	1	0	?	?
0	1	1	?	?
1	0	0	?	?
1	0	1	?	?
1	1	0	?	?
1	1	1	?	?

Adding Binary Numbers Using Logic Gates

- When adding two binary numbers, the most complex operation for one-bit addition is 1+1+1=11
  - This only happens when we carry data from a previous place

- You can think of the problems as

Digit A + Digit B + carry in= result + carry out

- If you can do this operation, you can repeat it to add any two binary numbers

Binary Addition

Answer: 1101+1001 = 10110

Carries

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

Digital A

Digital B

Check the answer:

$1101_2 = 13$

$1001_2 = \ \ 9$

$10110_2 = 22$

(Feb. 07, 2013 Stopped Here)

February 12, 2013

Review

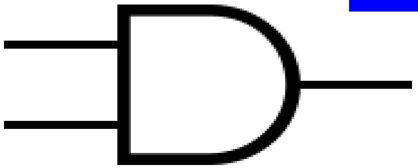
- Logic Operation
  - (1) AND
  - (2) OR
  - (3) NOT
  - (4) NAND
  - (5) NOR
  - (6) XOR

- Logic Circuit

- Logic Table

Review

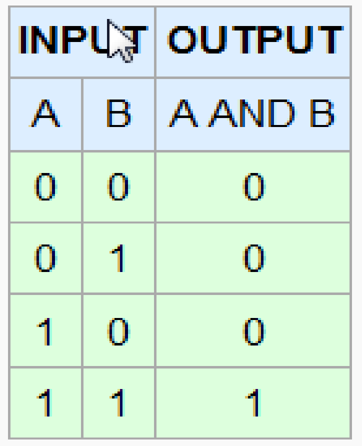
AND



$$A \cdot B$$

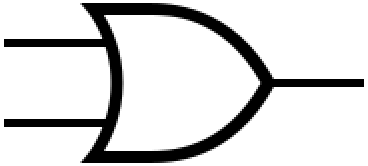
Symbol

Boolean Algebra: multiplication



INPUT		OUTPUT
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

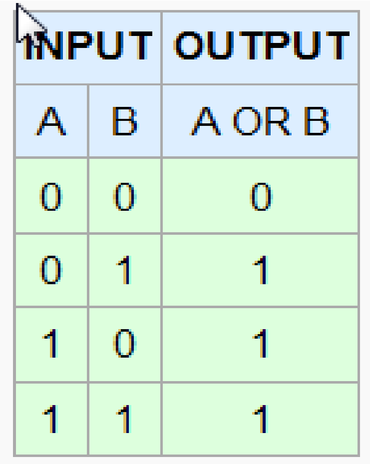
OR



$$A + B$$

Symbol

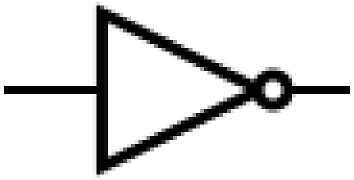
Boolean Algebra: plus



INPUT		OUTPUT
A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

Review

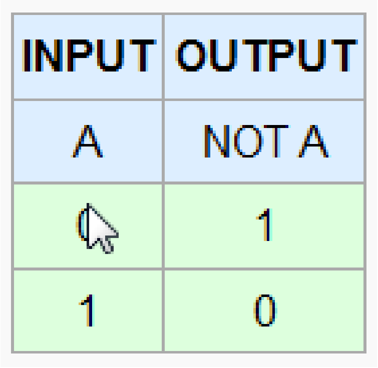
NOT



Symbol

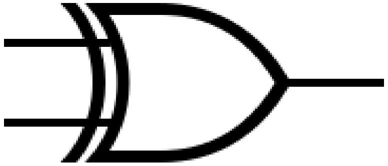
$$\overline{A}$$

Boolean Algebra: Negated, Reversed



INPUT	OUTPUT
A	NOT A
0	1
1	0

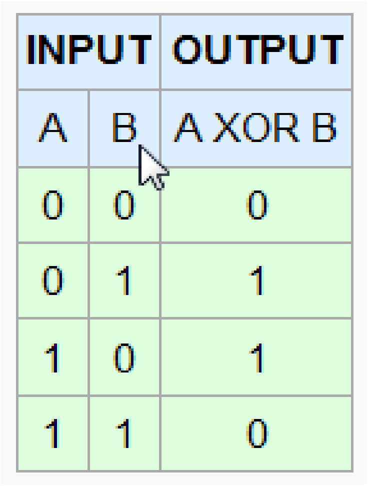
XOR



Symbol

$$A \oplus B$$

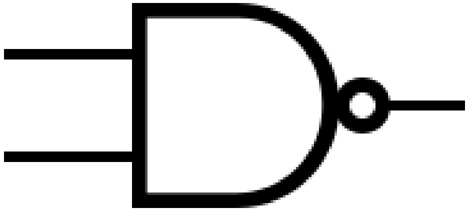
Boolean Algebra: Exclusive OR



INPUT		OUTPUT
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Review

NAND



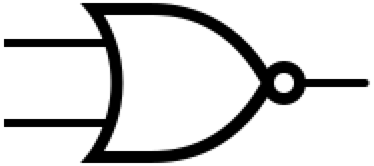
Symbol

$$\overline{A \cdot B}$$

Boolean Algebra

INPUT		OUTPUT
A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

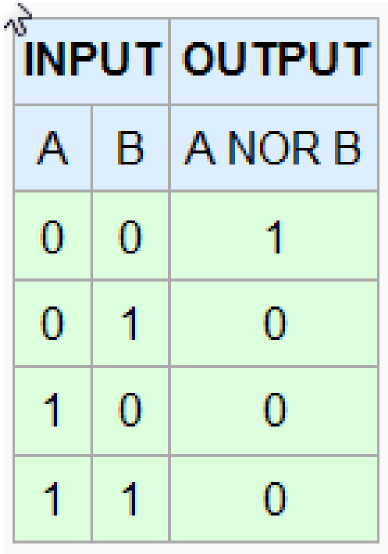
NOR



Symbol

$$\overline{A + B}$$

Boolean Algebra



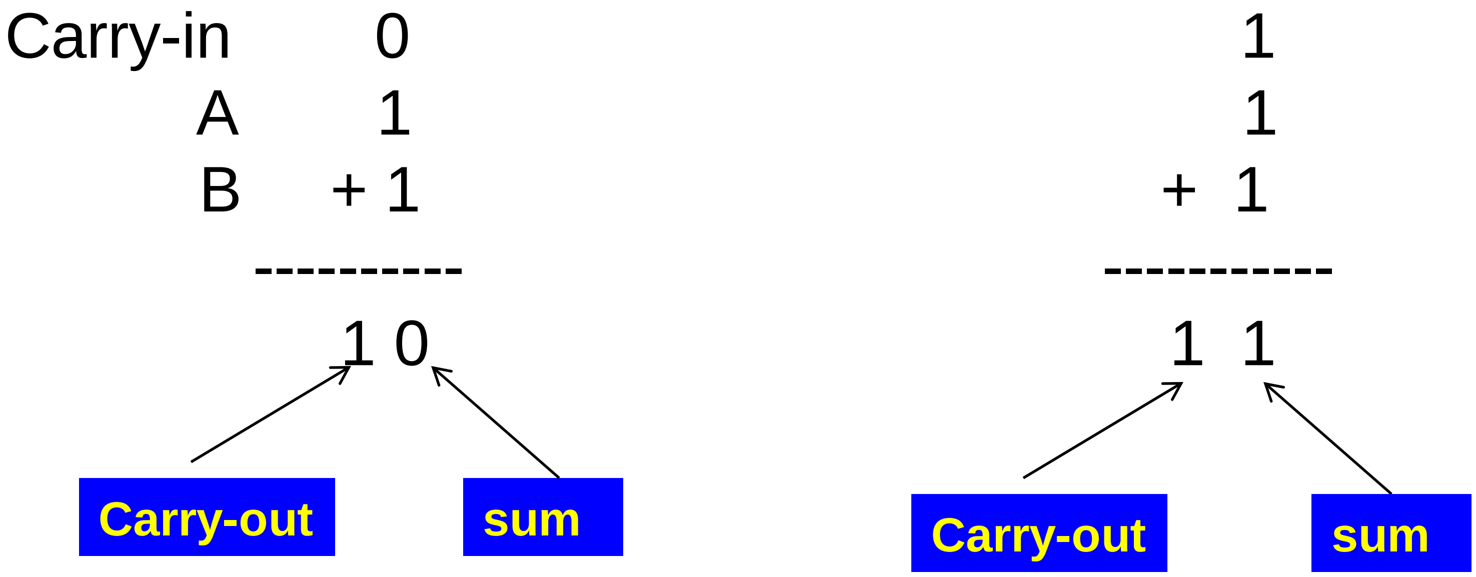
INPUT		OUTPUT
A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

Adding Binary Numbers

Create an adding machine using logic circuits: one-bit addition

You can think of the problem as

Digit A + Digit B + carry-in= sum + carry-out



Carry-in 0
A 1
B + 1
----
1 0

Carry-out **sum**

1
1
+ 1
----
1 1

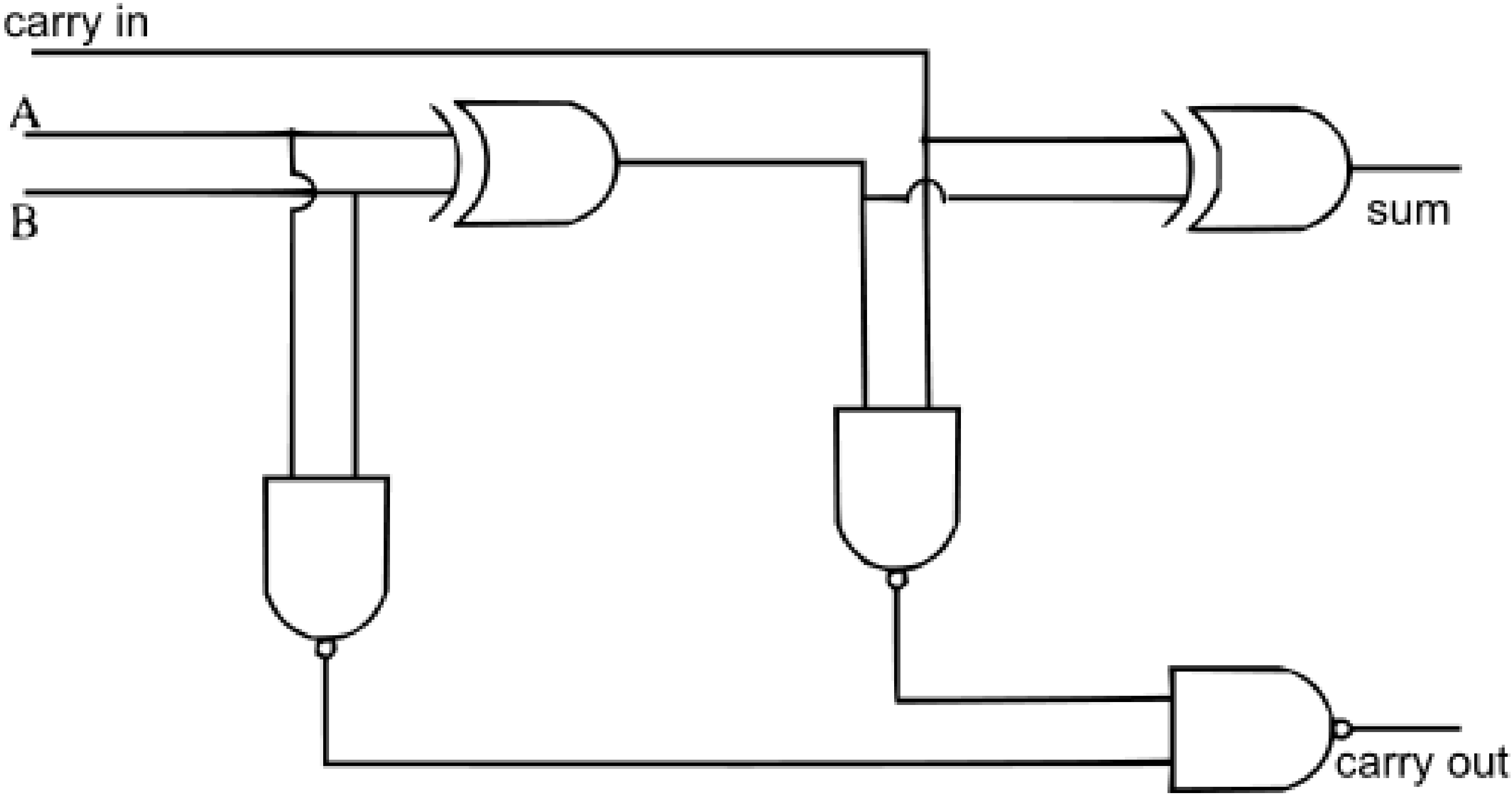
Carry-out **sum**

1-bit Adding Machine

The Logic Table

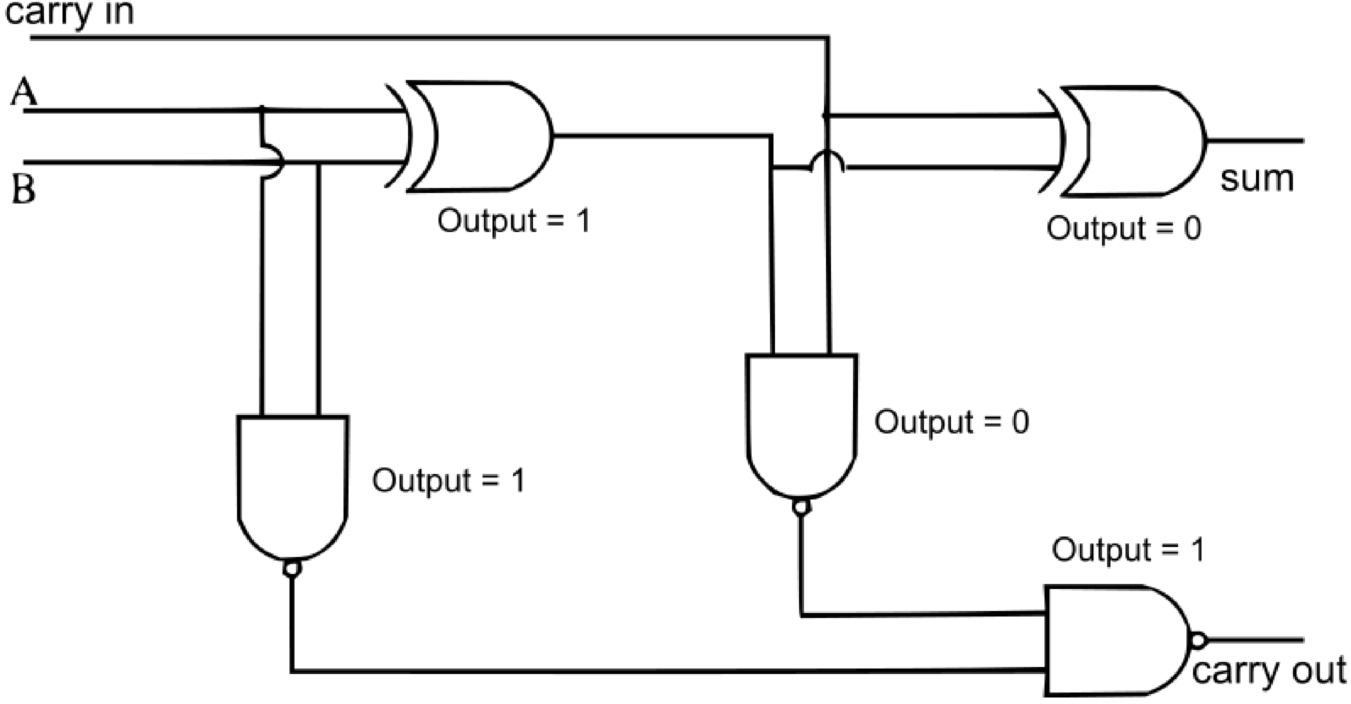
Carry-in	A	B	SUM	Carry-out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

1-bit Adding Machine



Two XOR gates and three NAND gates

1-bit Adding Machine

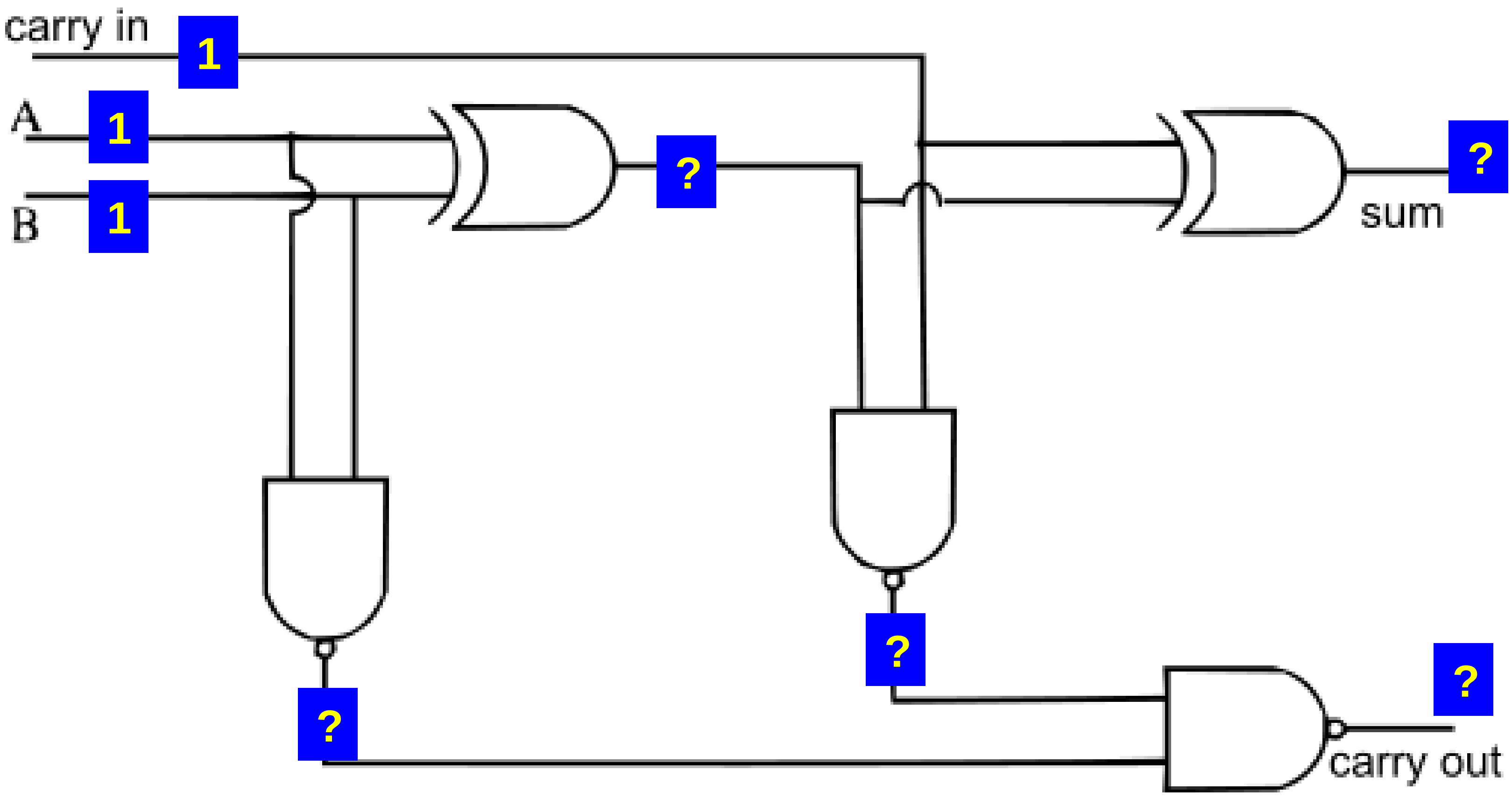


If A = 1, B = 0, and carry in = 1:

- The upper left EOR gate output is 1
- The lower left NAND gate output is 1
- The center NAND gate output is 0
- The upper right EOR gate (sum) output is 0
- The lower right NAND gate (carry out) output is 1

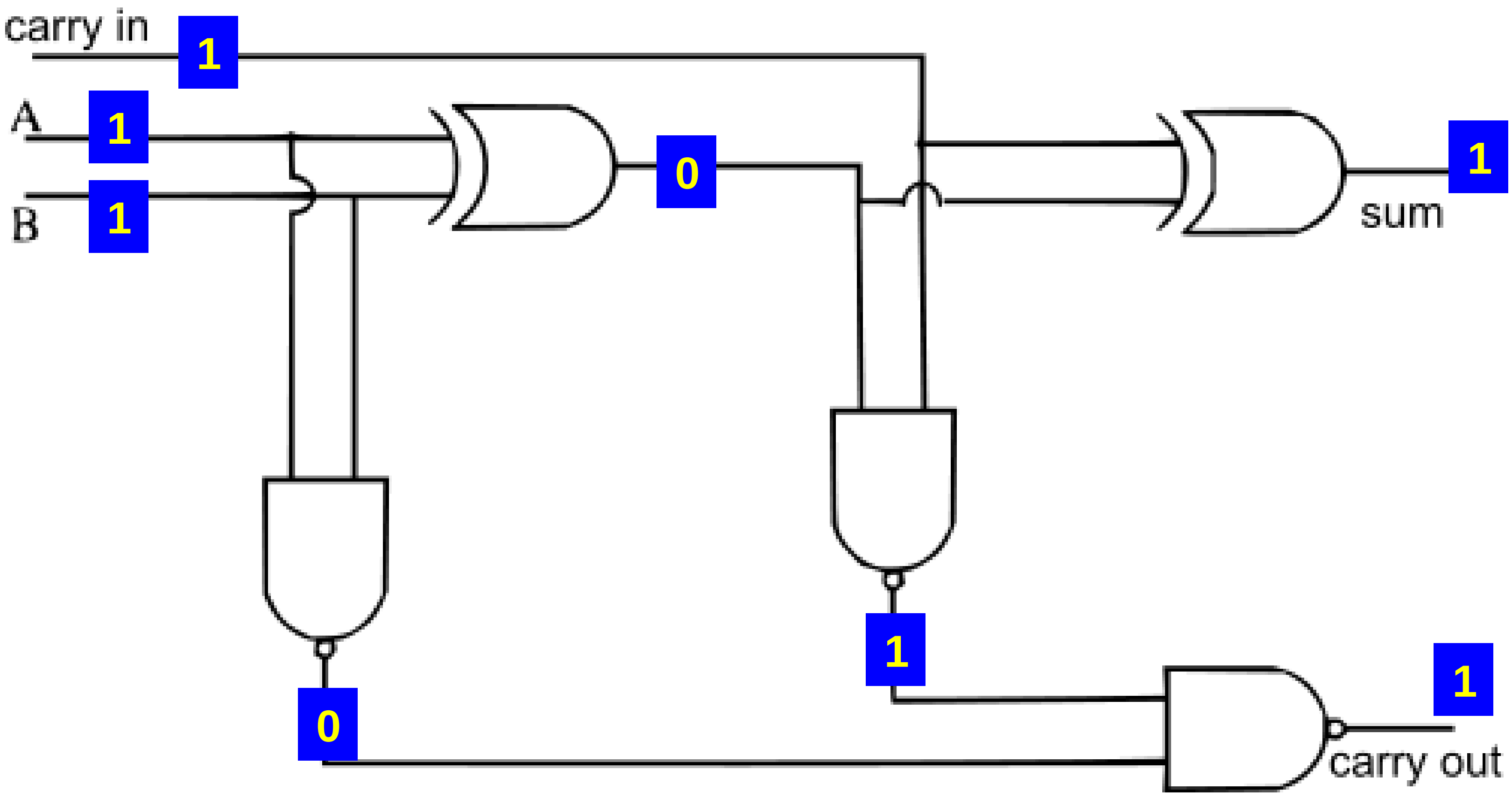
1-bit Adding Machine

Exercise: A=1, B=1, Carry-in=1, what are the outputs of the gates?



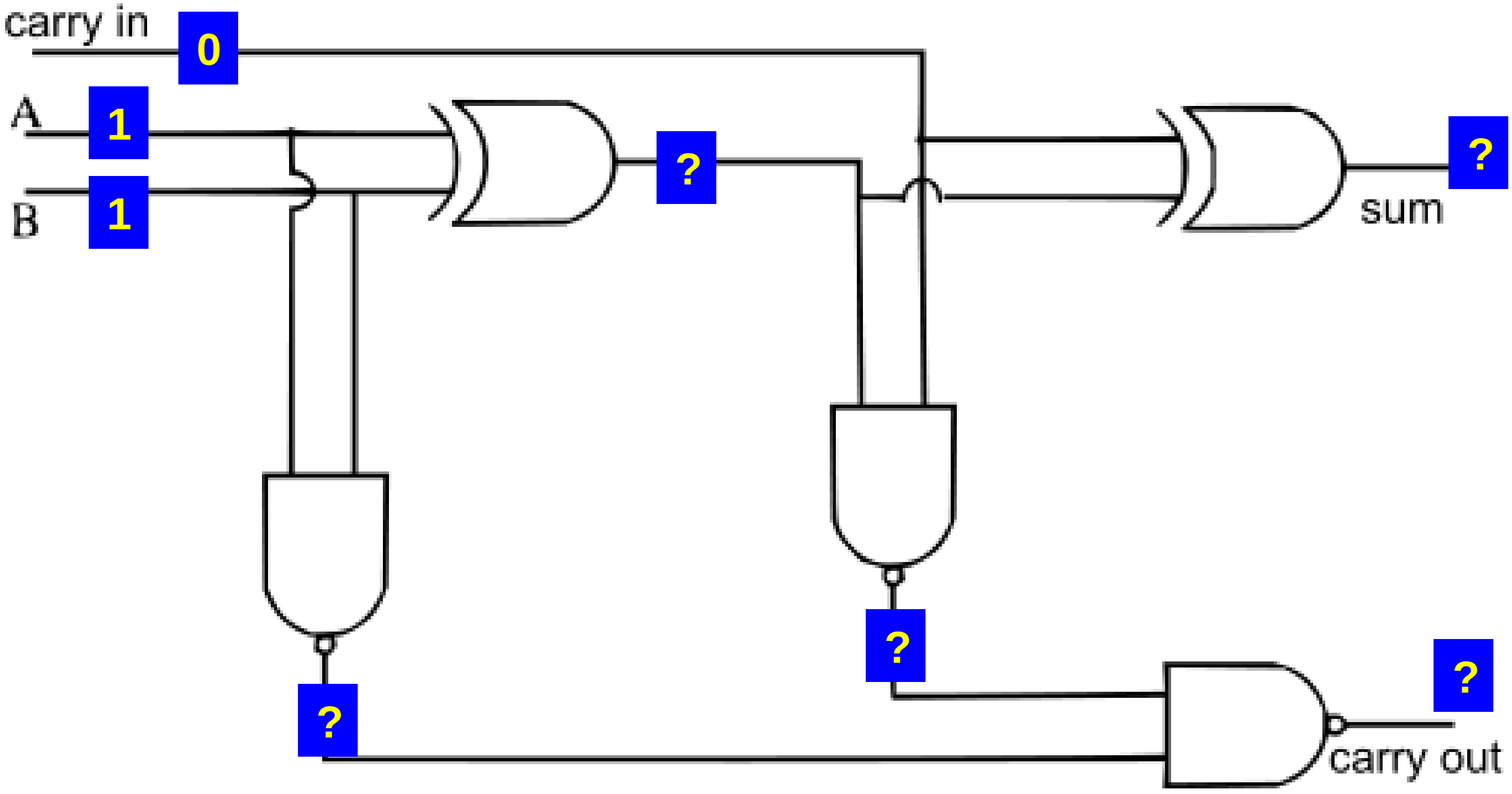
1-bit Adding Machine

Answer:



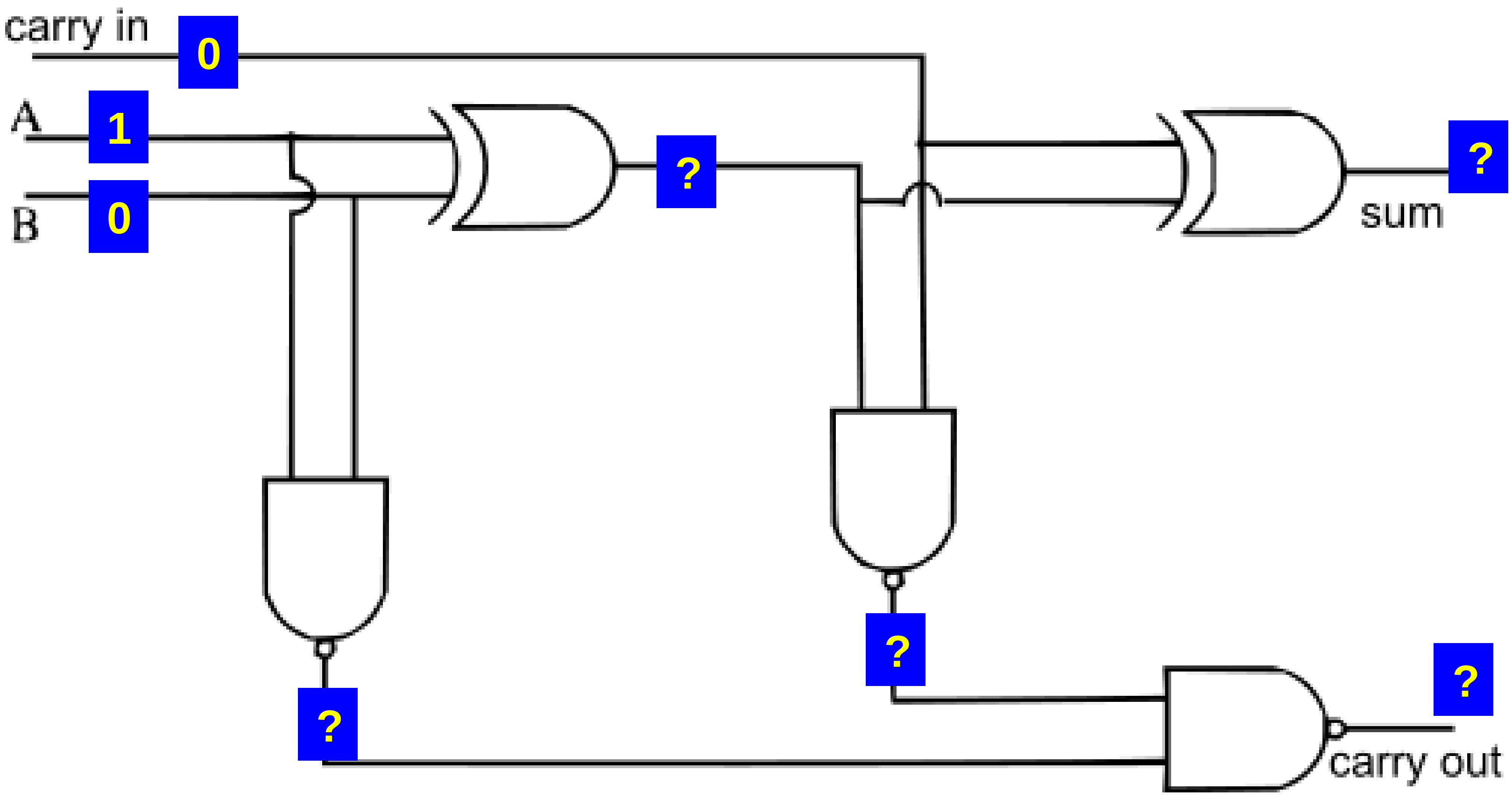
1-bit Adding Machine

Exercise: A=1, B=1, Carry-in=0, what are the outputs of the gates?

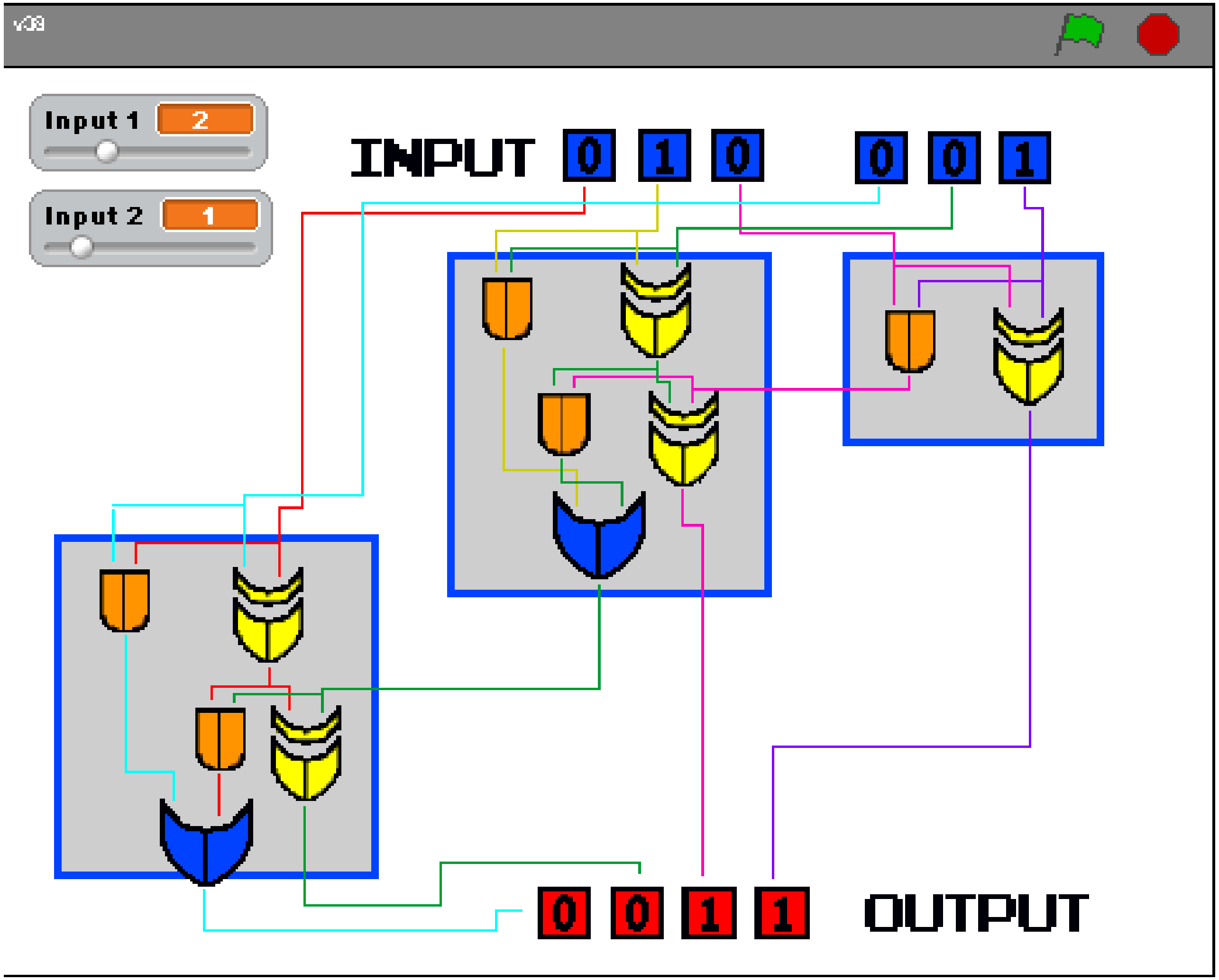


1-bit Adding Machine

Exercise: A=1, B=0, Carry-in=0, what are the outputs of the gates?



A 3-bit Adding Machine



http://scratch.mit.edu/projects/bla/192263

Computer

A computer is a machine that manipulates data according to a set of instructions called a computer program

--http://en.wikipedia.org/wiki/Computing

By now, we shall know the basis of a computer:

- Machine: electronic machine built upon electronic logic circuits
- Manipulate: the operation of logic circuits
- Data: encoded into binary data

Next, we need to learn how to "computer program": the instruction of manipulating data

The End of Section 1

Computer Fundamentals

(February 12, 2013, Final)