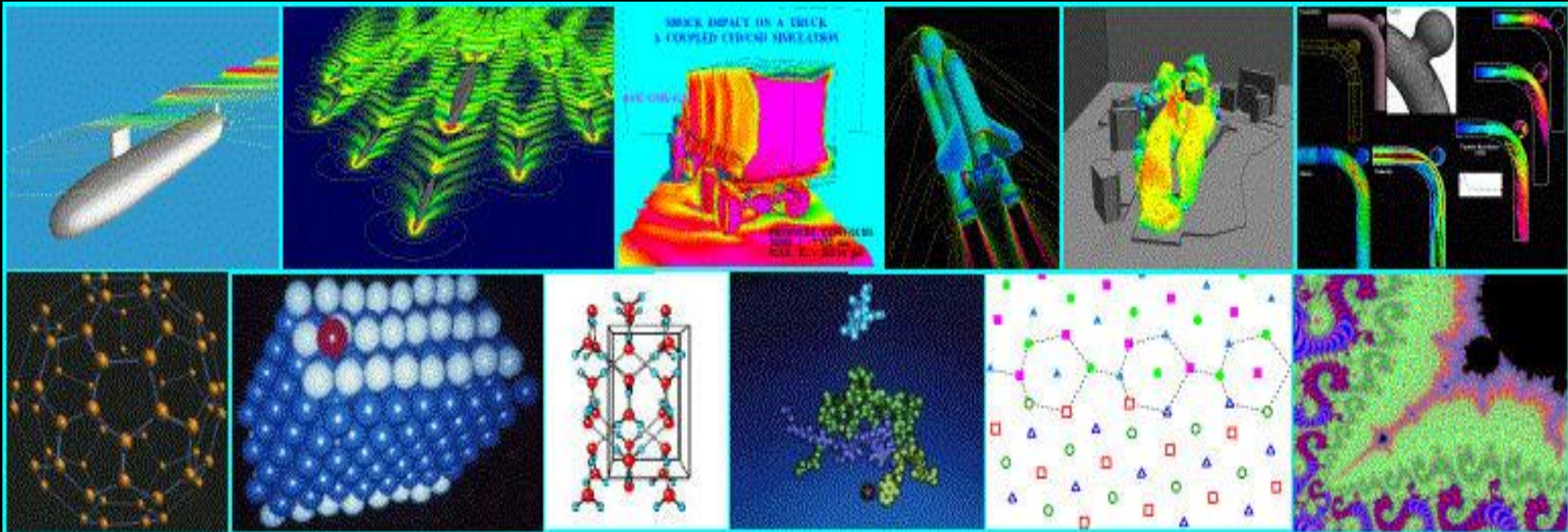


Computing for Scientists

Section I

Computer Fundamentals (CF)

(August 29, 2013 - September 24, 2013)



Jie Zhang

Copyright ©

CDS 130 - 001
Fall, 2013

Computer Fundamentals (CF)

CH1. Binary Representation

- Binary numbers to/from Decimal numbers

CH2. Binary Operation

- Binary addition and subtraction

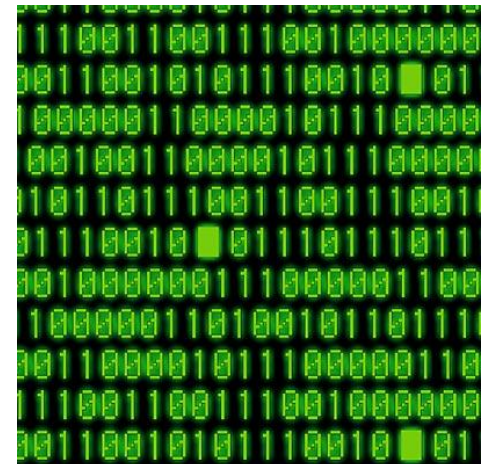
CH3. Binary Encoding/Decoding

- Binary numbers to/from Characters (ASCII code)

CH4. Logic Circuit and Logic Table

- Binary operation in hardware

CH5. Data Storage





Section 1

Chapter 1

Binary Representation

(Jan. 29, 2013)

Questions?

- Why do computers like binaries?
- Why do human like decimals?
- What is positional notation?
- What is the value (count) of 11?

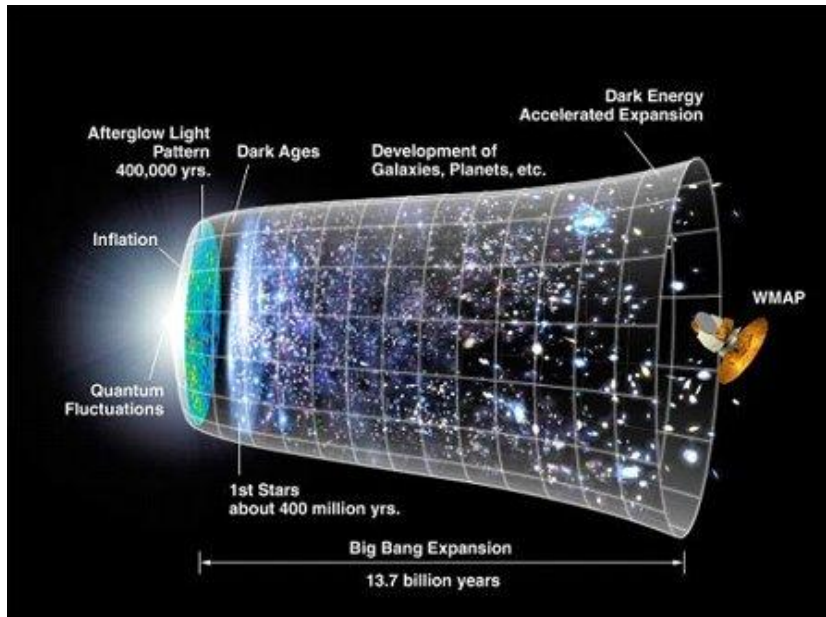
Objectives

- 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
















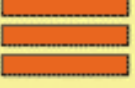







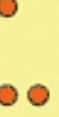

Arabic Numerals: base 10

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

0 	1 	2 	3 	4 
5 	6 	7 	8 	9 
10 	11 	12 	13 	14 
15 	16 	17 	18 	19 
20 	21 	22 	23 	24 

**Maya Calendar:
expressed by five
digits**

**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 had ten different digit
- Why if the counts are more than nine, the largest number?
 - Well, this is the power of zero

Positional Notation

It uses **positional notation to represent big numbers**

- Use same symbols
- but different position represent different values in orders of magnitude

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)

$$1000+400+70+8=1478$$

Power

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

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

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

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

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

4 units of the fourth place: value = $4 \times 10^3 = 4000$

Decimal Template

$$1 \times 10^3 + 4 \times 10^2 + 7 \times 10^1 + 8 \times 10^0$$

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

1	4	7	8
---	---	---	---

How many units in the place

10^3	10^2	10^1	10^0
--------	--------	--------	--------

(1000)	(100)	(10)	(1)
--------	-------	------	-----

Power notation of the value of one unit in the place

Binary Numeral Numbers

- It is based on digits only: 0 and 1
- Machines like binary representation
 - Much easier to create two states in hardware, or easier in engineer, instead of creating ten states
 - Open (electric current flows) versus Close (No electric current in the circuit)
 - High electric voltage versus low electric voltage

CH1. Binary Representation

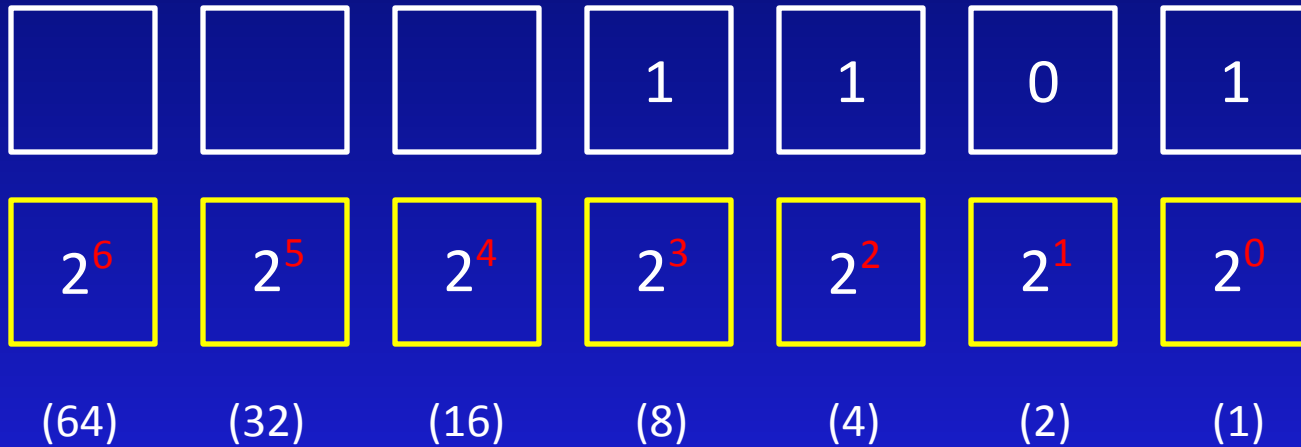
Topic: Binary to Decimal

Binary - Decimal

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

Binary Template



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,

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

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

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
-------	-------	-------	-------	-------	-------	-------	-------

(128)

(64)

(32)

(16)

(8)

(4)

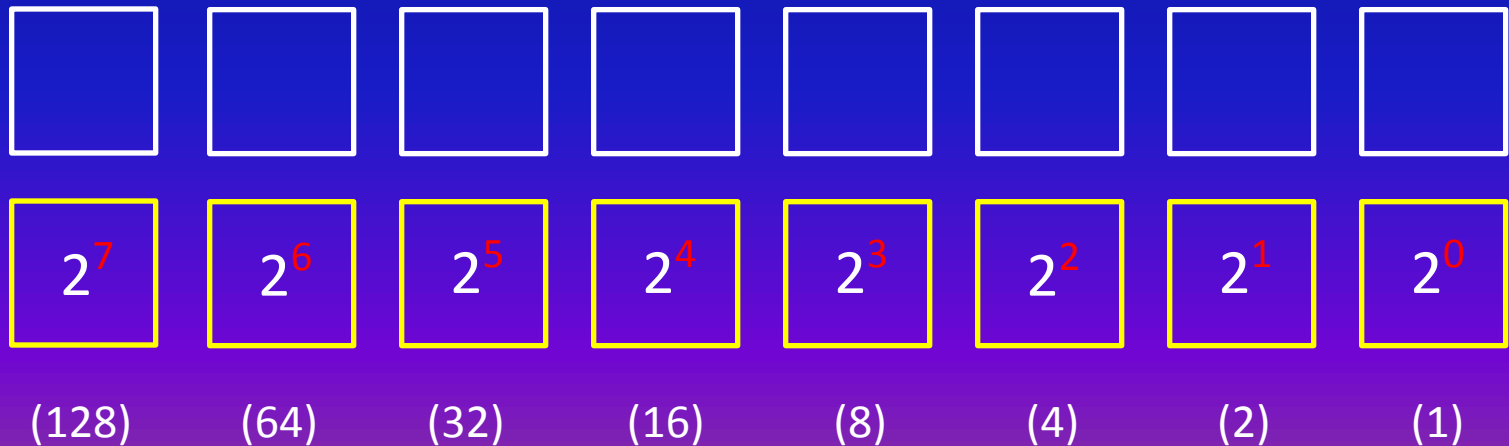
(2)

(1)

Practice Doing It a Few More Times -- ANSWERS

Converted binary numbers:

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



CH1. Binary Representation

Topic: Decimal to Binary

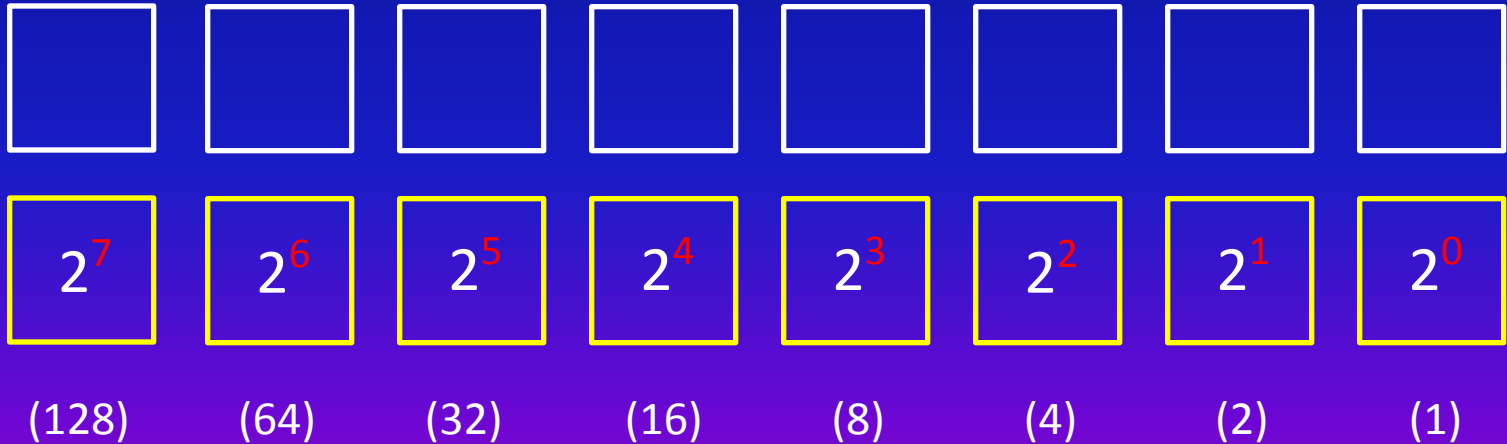
Backwards . . .

45

?



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



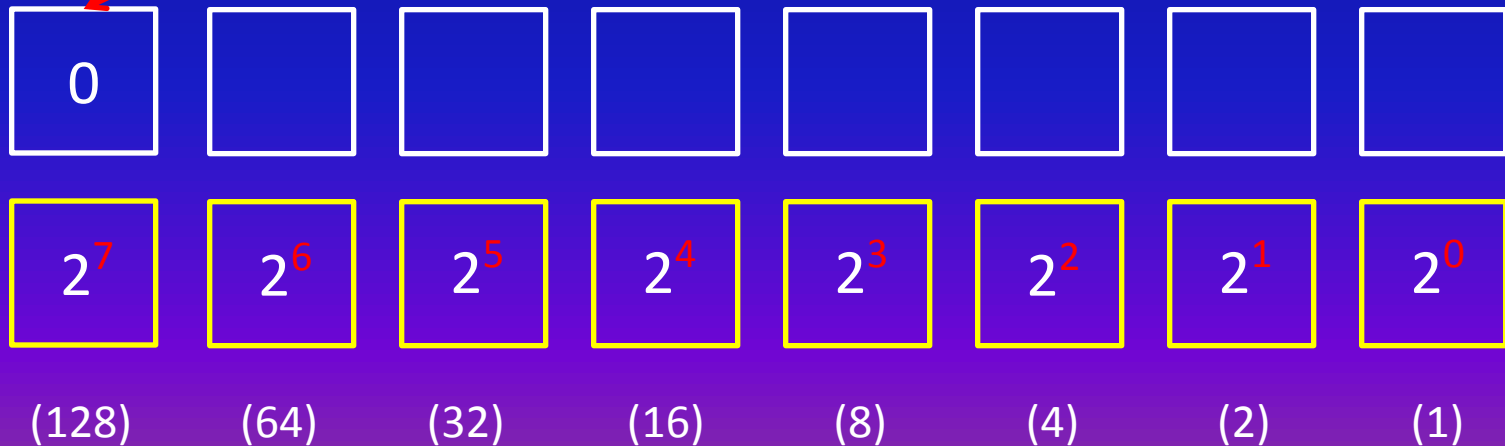
Backwards . . .

45

BEGIN FROM THE LEFT

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

Zero! (None)

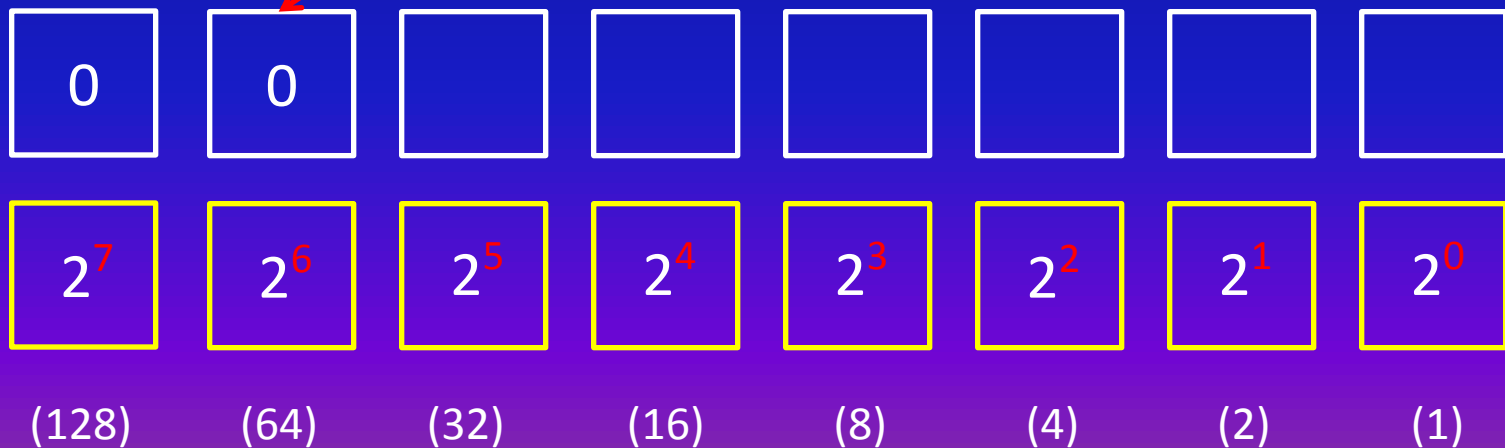


Backwards . . .

45

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

Zero! (None)



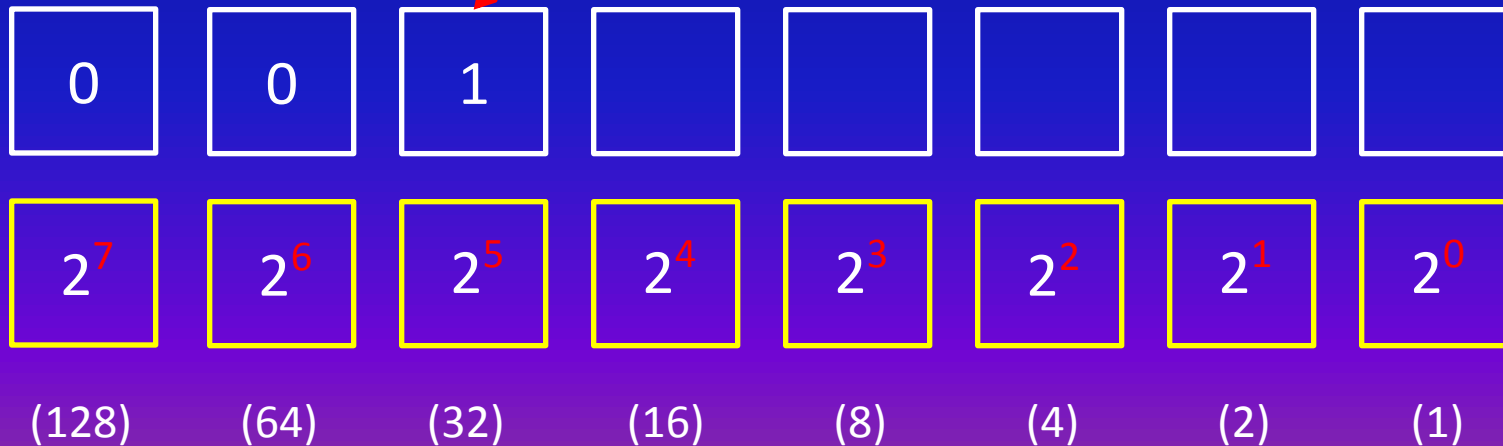
Backwards . . .

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

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

One!

Then, *TAKE IT OUT* . . .

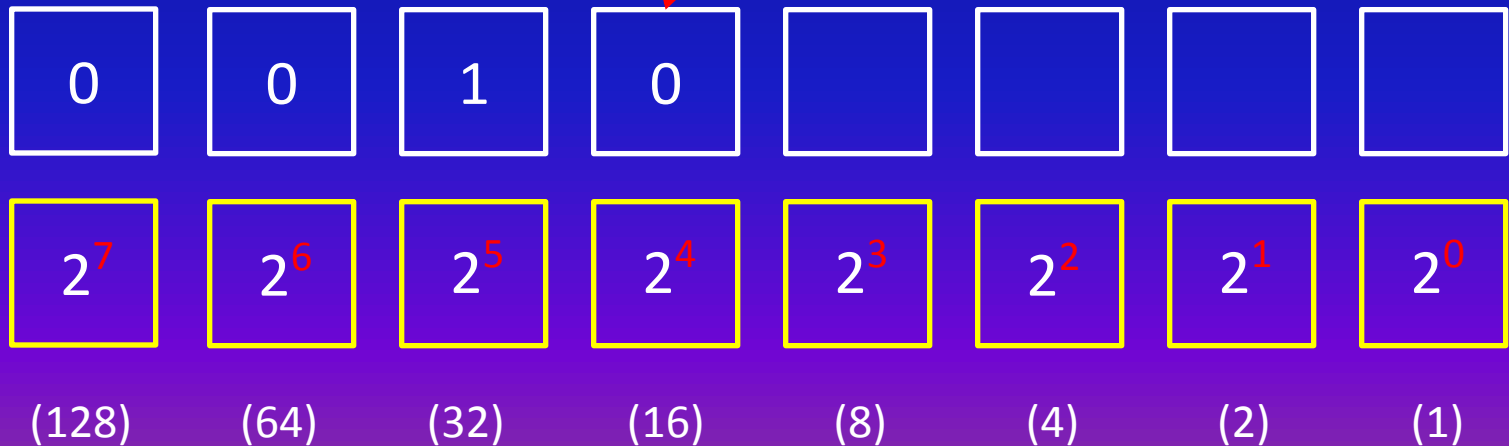


Backwards . . .

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

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

Zero! (None)



Backwards . . .

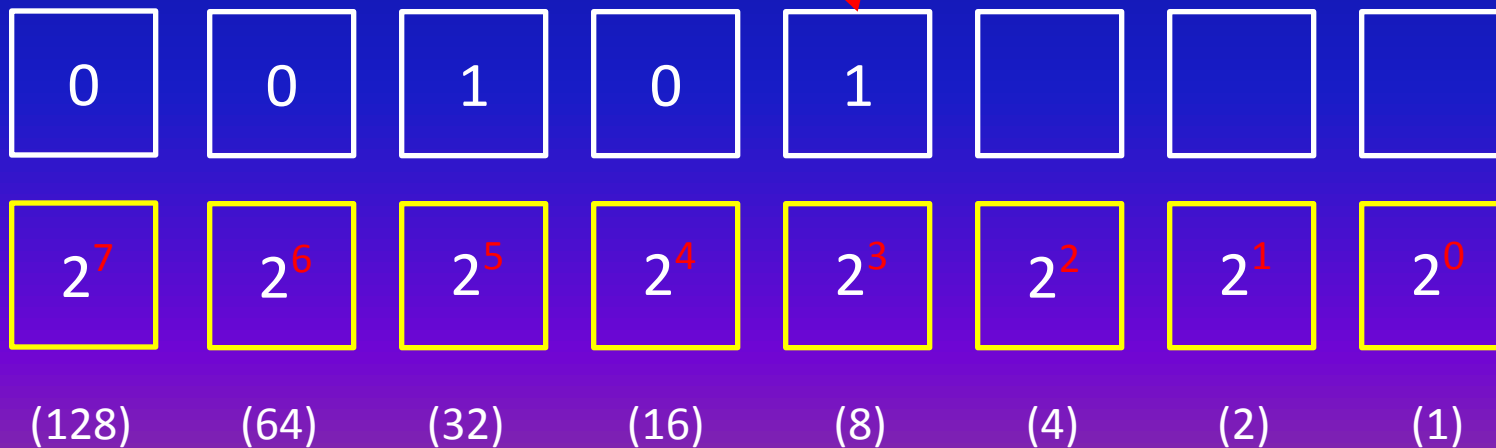
$$\begin{array}{r} 45 \\ -32 \\ \hline 13 \\ -8 \\ \hline 5 \end{array}$$

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

One!



Then, *TAKE IT OUT* . . .



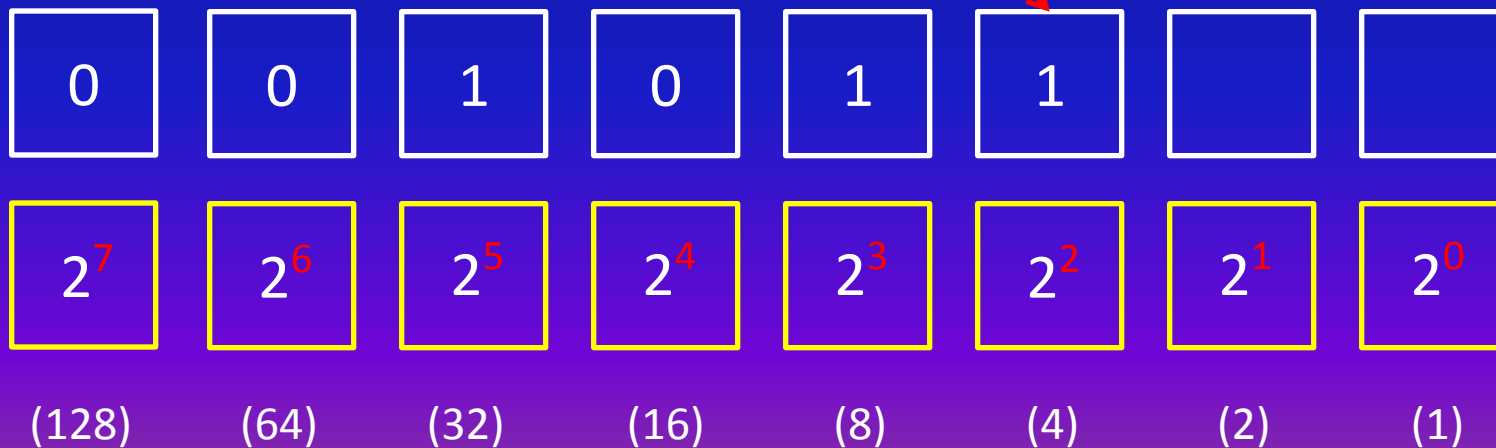
Backwards . . .

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

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

One!

Then, *TAKE IT OUT* . . .

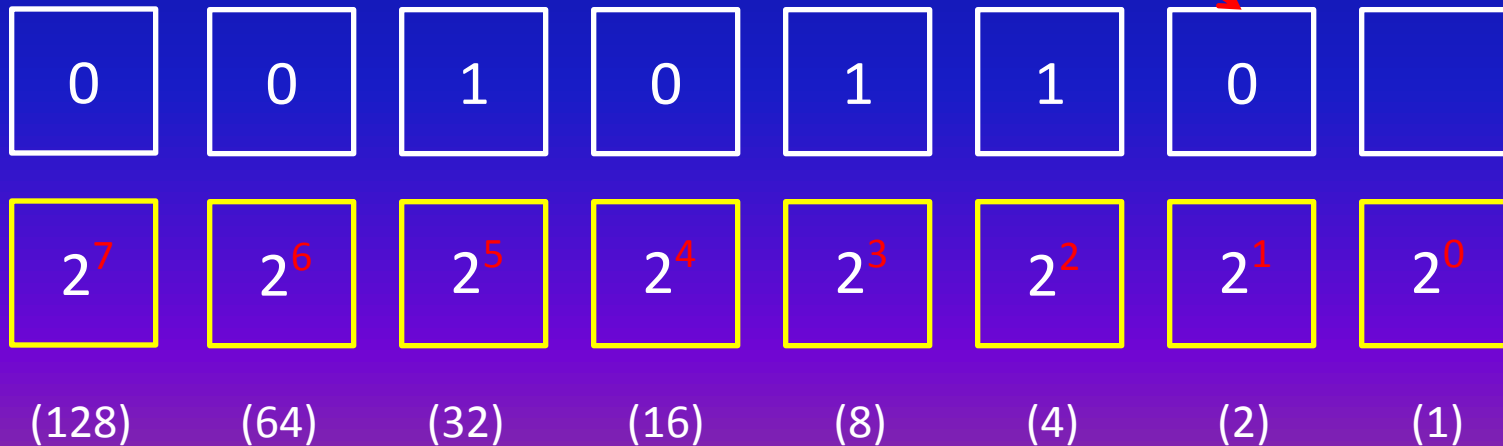
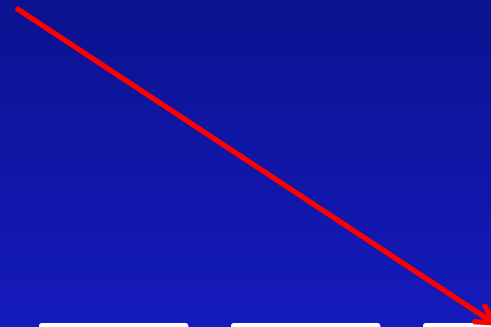


Backwards . . .

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

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

Zero! (None)



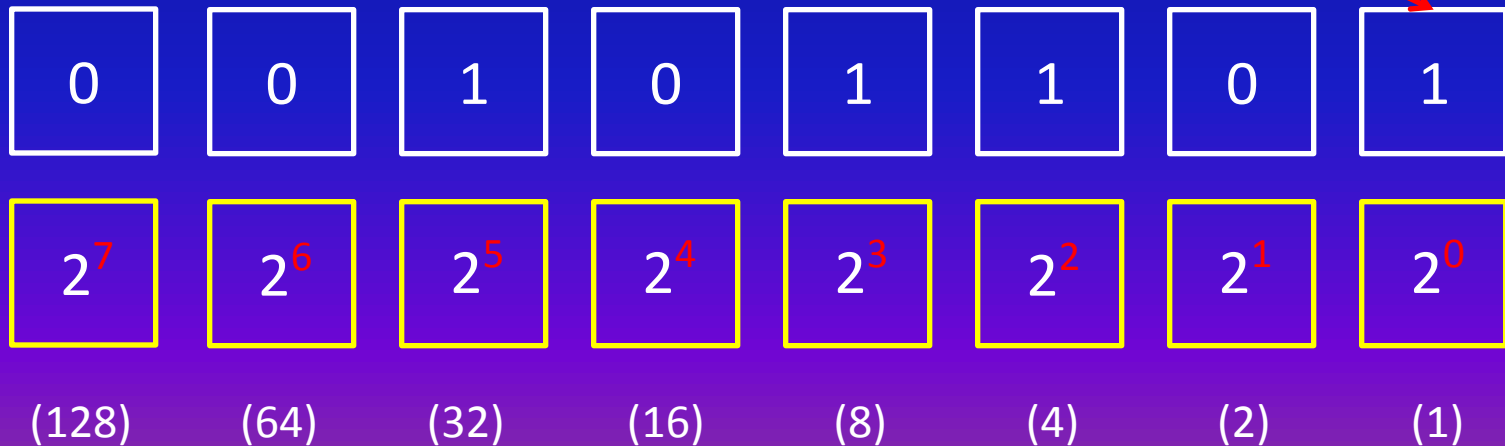
Backwards ...

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

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

One!

Then, *TAKE IT OUT* ...



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)



(Aug. 29, 2013 Stops Here)



(Sep. 03, 2013 Starts Here)

A Short Review



- Why do computers like binaries?
- What is positional notation?
- What is the value (count) of 111?
- How to convert binary to decimal?
- How to convert decimal to binary?

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?

$$\begin{array}{r} 2 \overline{)13} \quad 1 \\ \underline{2)6} \quad 0 \\ \underline{2)3} \quad 1 \\ \underline{2)1} \quad 1 \\ 0 \end{array}$$

The answer: 1101

Long Division Method

For decimals: the trivial case, for illustration purpose only

$$\begin{array}{r} 10 \overline{)1478} \quad 8 \\ 10 \overline{)147} \quad 7 \\ 10 \overline{)14} \quad 4 \\ 10 \overline{)1} \quad 1 \\ \quad 0 \end{array}$$

The answer: 1478

Long Division Method

$$\begin{array}{r} 2 \overline{)45} \quad 1 \\ 2 \overline{)22} \quad 0 \\ 2 \overline{)11} \quad 1 \\ 2 \overline{)5} \quad 1 \\ 2 \overline{)2} \quad 0 \\ 2 \overline{)1} \quad 1 \\ 0 \end{array}$$

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

One More Time!

(Base Two, or Binary ← Base Ten)

Problem: Convert 97_{10} to binary

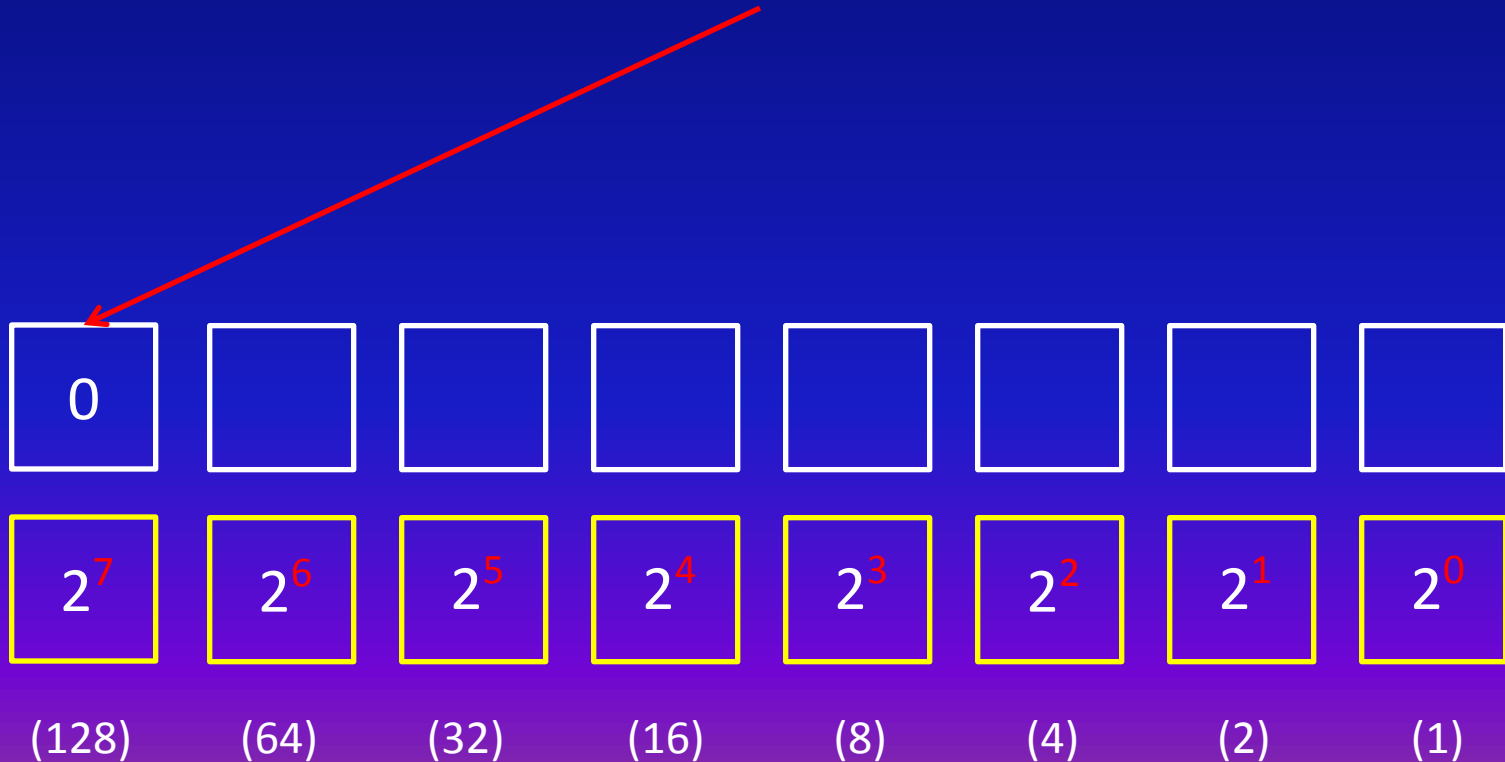
97

One More Time!

BEGIN FROM THE LEFT

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

Zero! (None)



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

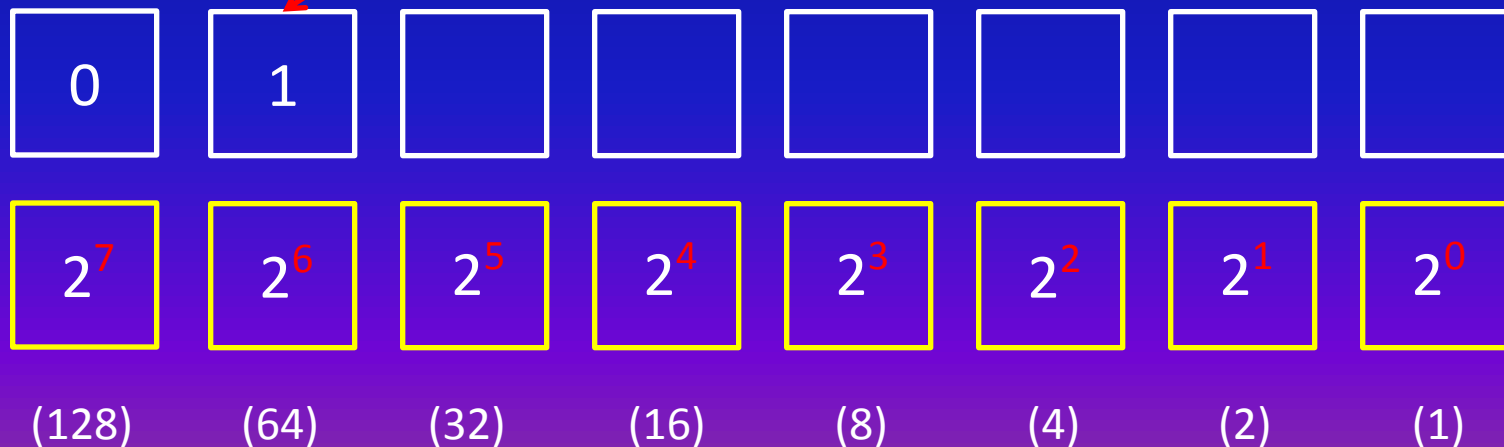
One More Time!

BEGIN FROM THE LEFT

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

One!

Then, *TAKE IT OUT* ...



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

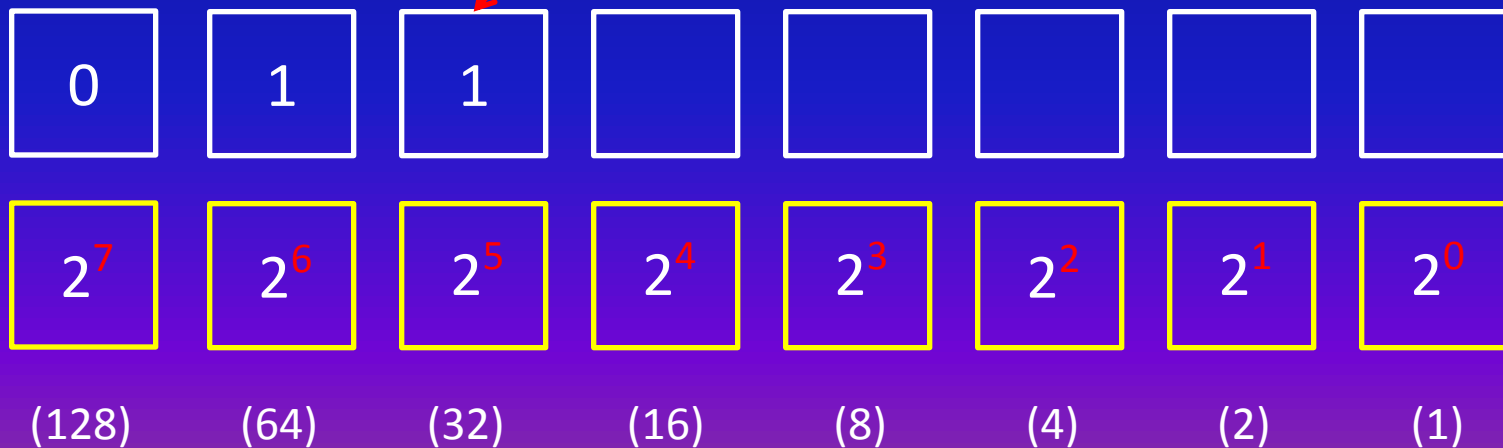
One More Time!

BEGIN FROM THE LEFT

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

One!

Then, *TAKE IT OUT* ...



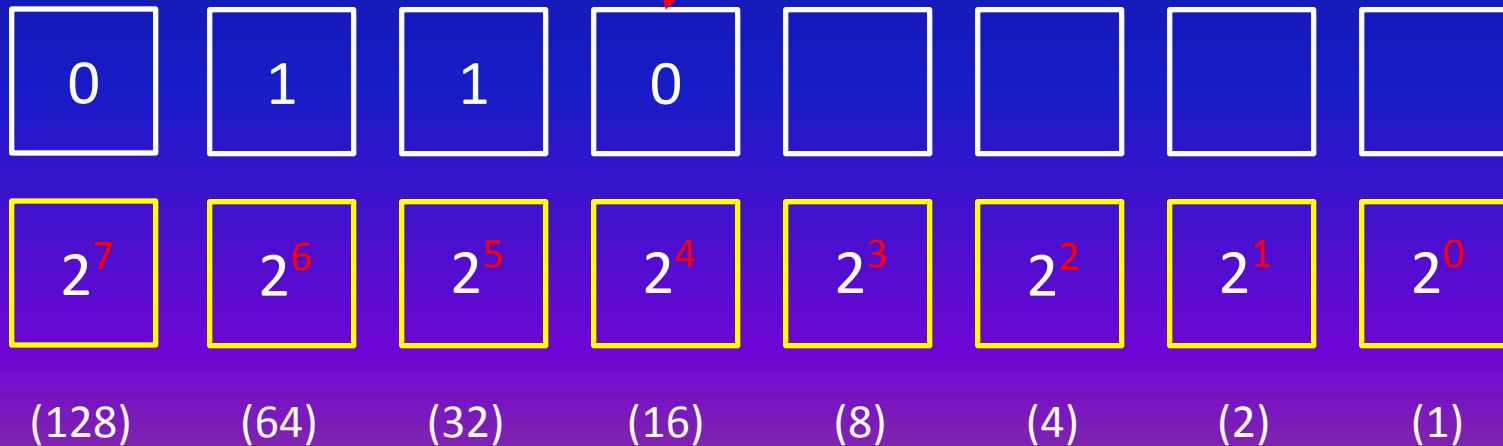
$$\begin{array}{r} 97 \\ - 64 \\ \hline 33 \\ - 32 \\ \hline 1 \end{array}$$

One More Time!

BEGIN FROM THE LEFT

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

Zero! (none)



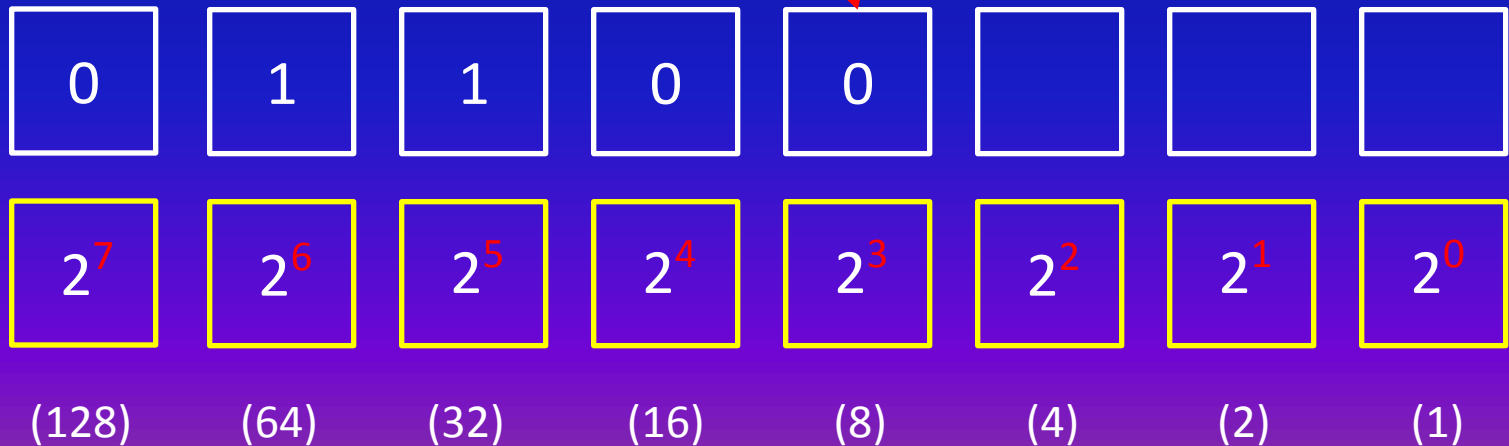
$$\begin{array}{r} 97 \\ - 64 \\ \hline 33 \\ - 32 \\ \hline 1 \end{array}$$

One More Time!

BEGIN FROM THE LEFT

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

Zero! (none)



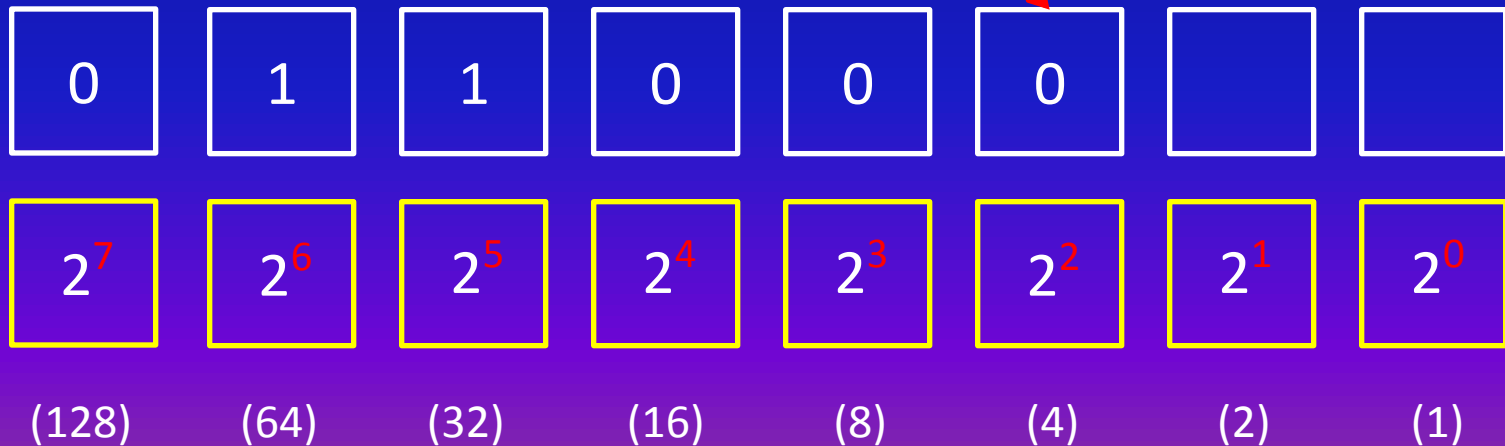
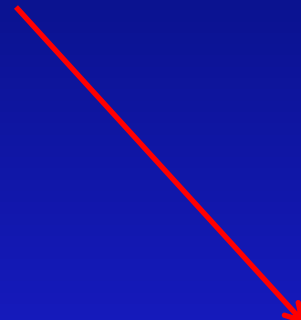
$$\begin{array}{r} 97 \\ - 64 \\ \hline 33 \\ - 32 \\ \hline 1 \end{array}$$

One More Time!

BEGIN FROM THE LEFT

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

Zero! (none)



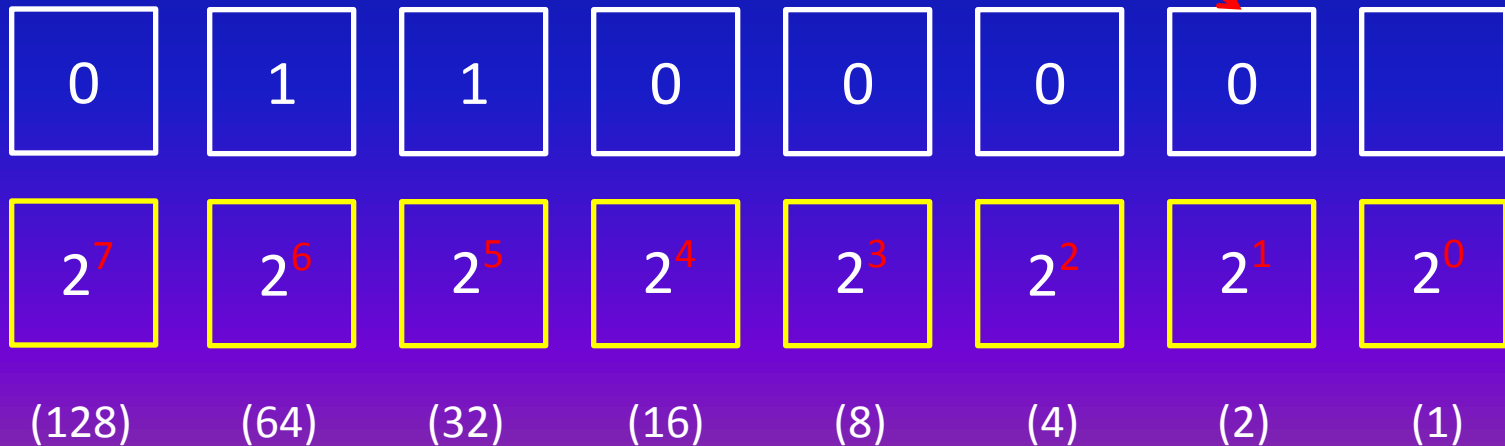
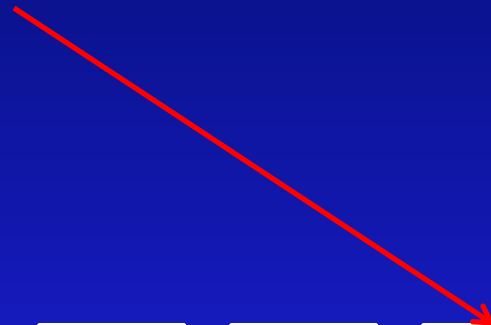
$$\begin{array}{r} 97 \\ - 64 \\ \hline 33 \\ - 32 \\ \hline 1 \end{array}$$

One More Time!

BEGIN FROM THE LEFT

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

Zero! (none)



One More Time!

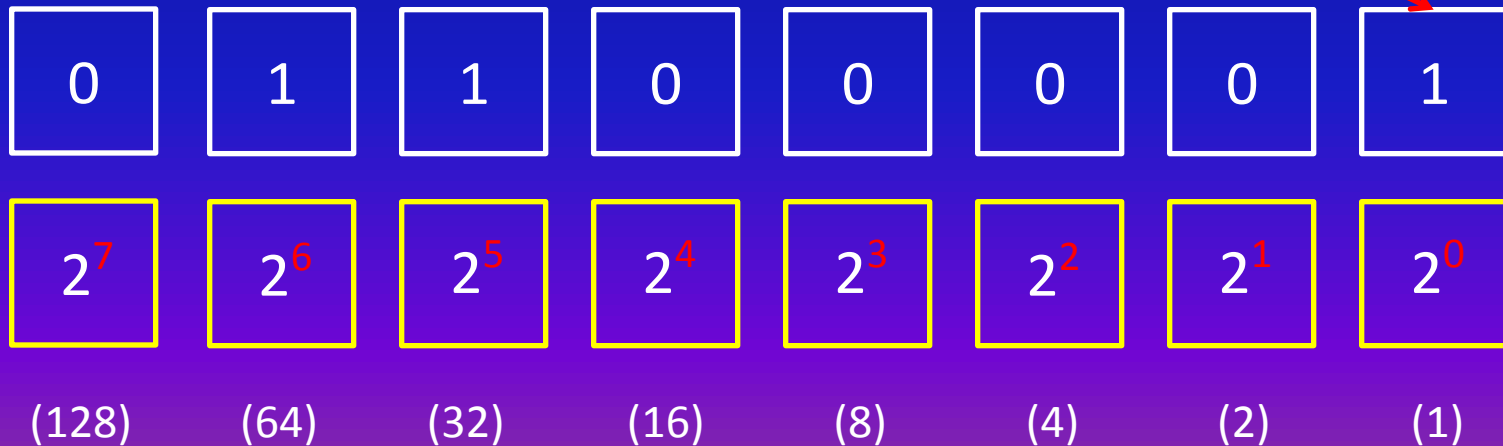
$$\begin{array}{r} 97 \\ - 64 \\ \hline 33 \\ - 32 \\ \hline 1 \\ - 1 \\ \hline 0 \end{array}$$

BEGIN FROM THE LEFT

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

One!

Then, *TAKE IT OUT* . . .

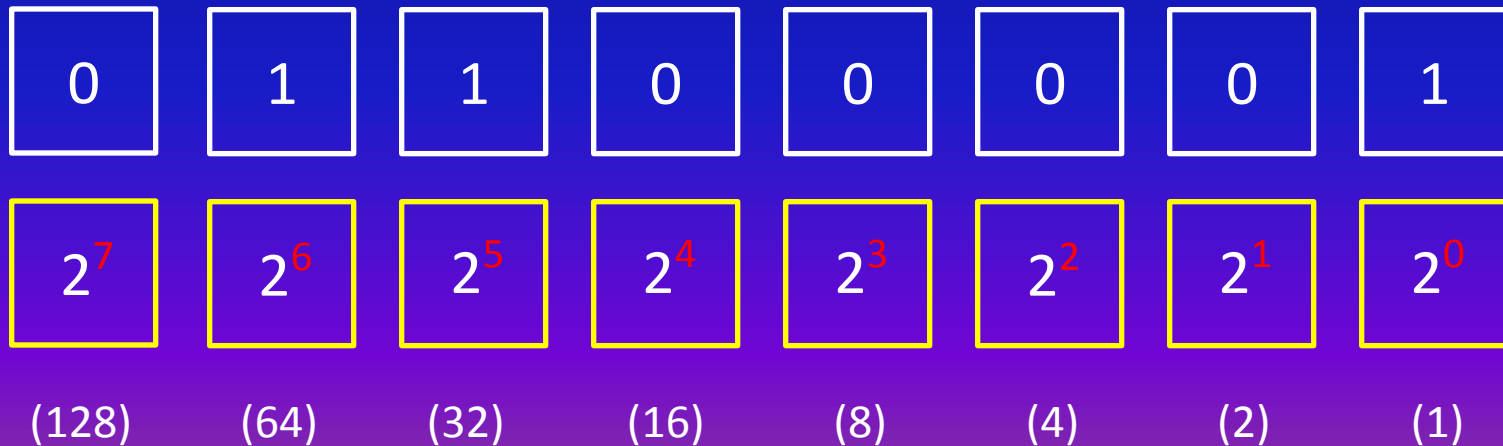


WALA!

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

$$97_{10} = \mathbf{01100001}_2$$

$$97_{10} = \mathbf{1100001}_2$$



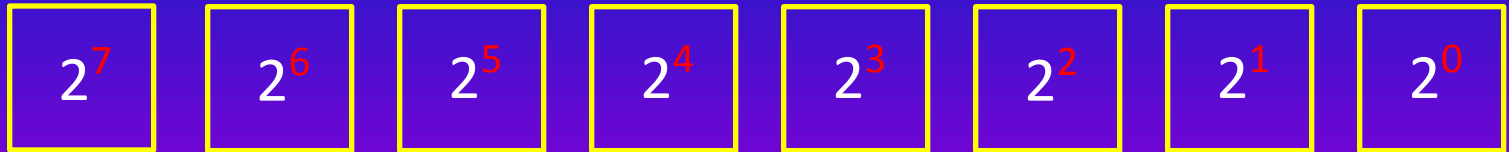
Long Division Method

2)97 1
2)48 0
2)24 0
2)12 0
 2)6 0
 2)3 1
 2)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



(128)

(64)

(32)

(16)

(8)

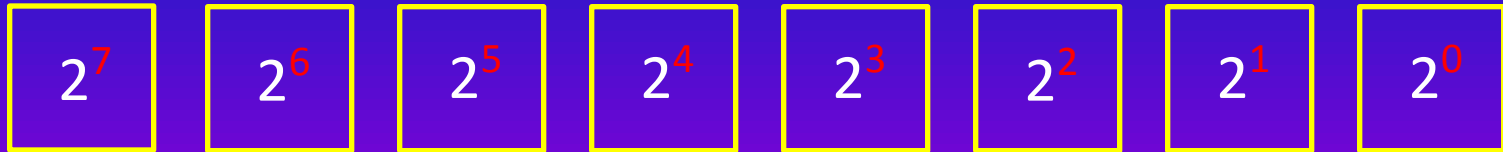
(4)

(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**



(128)

(64)

(32)

(16)

(8)

(4)

(2)

(1)



The End of Chapter 1



Section 1

Chapter 2

Binary Operation

(Sep. 03, 2013)

Questions?

- What is $111 + 101$?
- What is $111 - 101$?
- What is the value of $(111)_2$, $(111)_8$, $(111)_{10}$, $(111)_{16}$?
- What is $(ABC)_{16}$?

CH2: Objectives

- Binary Addition
- Binary Subtraction
- 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:

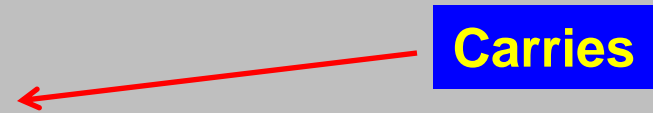
Find the answer of $10110 + 00111$?

Binary Addition

Answer: $10110 + 00111 = 11101$

$$\begin{array}{r} 10110 \\ + 00111 \\ \hline 11101 \end{array}$$

Carries



Check the answer:

$$10110_2 = 22$$

$$00111_2 = 7$$

$$11101_2 = 29$$

Binary Addition

Question:

Find the answer of $1101 + 1001$?

Binary Addition

Answer: $1101 + 1001 = 10110$

$$\begin{array}{r} 1 \quad 1 \\ 1101 \\ + 1001 \\ \hline 10110 \end{array}$$

Carries



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: $1110 - 1011 = 0011$

$$\begin{array}{r} 022 \leftarrow \text{borrowed} \\ 1\cancel{1}\cancel{1}0 \\ - 1011 \\ \hline 0011 \end{array}$$

Check the answer:

$$1110_2 = 14$$

$$1011_2 = 11$$

$$0011_2 = 3$$

Binary Subtraction

Question:

Find the answer of $10110 - 00111$?

Binary Subtraction

Answer: $10110_2 - 00111_2 = 01111_2$

$$\begin{array}{r} 01222 \leftarrow \text{borrowed} \\ \cancel{10110} \\ - 00111 \\ \hline 01111 \end{array}$$

Check the answer:

$$10110_2 = 22$$

$$00111_2 = 7$$

$$01111_2 = 15$$

Binary Subtraction

Question (your turn):

Find the answer of $1101 - 1001$?



(Sep. 03, 2013 Stops Here)



(Sep. 05, 2013 Starts Here)

A Short Review



- Binary addition
 - What is $(10011)_2 + (00111)_2$?
- Binary subtraction
 - What is $(10011)_2 - (00111)_2$?

CH2. Binary Operation

Topic: Other numeral systems

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 base-8 instead of base-2

Question: What is the decimal value of 456_8 ?

$$\begin{aligned} & 4 \times 8^2 + 5 \times 8^1 + 6 \times 8^0 \\ = & 256 + 40 + 6 = 302 \end{aligned}$$



Octal Numerals

Question (your turn):

What is the decimal value of 123_8 ?

Octal Numerals

Answer

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

Hexadecimal Template

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

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

$$\begin{aligned} & 4 \times 16^2 + 5 \times 16^1 + 6 \times 16^0 \\ = & 4 \times 256 + 5 \times 16 + 6 \times 1 = 1110 \end{aligned}$$



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

$$\text{FF}_{16} = 15 \times 16 + 15 = (255)_{10}$$

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

Thus,

$$(\text{FF})_{16} = (255)_{10} = (1111\ 1111)_2$$

**Why do programmers always mix up
Halloween and Christmas?**

Because Oct 31 equals Dec 25.



The End of Chapter 2



Section 1

Chapter 3

Binary Encoding/Decoding

(Sep. 05, 2013)

Questions?

- How does a computer store “Hello” in the disk?
- What does “01001010 01011010” mean in characters?

Encoding / Decoding

The interface between machines and human

- Encoding, in general, is a code that pairs each character with something else
- **Encoding text** involves transforming it into an "encoded" number (e.g., using ASCII code), then into a binary sequence.
- **Decoding text** is a reverse process, i.e., converting from binary sequence into a string of characters
- Once data are transformed into encoded binary sequence, it can then be stored, transmitted and processed by digital devices

Encoding / Decoding

- Music (e.g, au), videos (e.g., mpg), pictures (e.g, gif, png) any many other things can be encoded.
- **Morse code** was introduced in the 1840s and is used to encode each letter of the Latin alphabet, each Arabic numeral, and some other characters via a series **of long and short presses of a telegraph key**

ASCII Code

ASCII: American Standard Code for Information Interchange

--<http://www.asciitable.com/>

ASCII was introduced in 1963 and is used to encode letters, numerals, symbols, and device control codes as fixed-length codes using **integers (e.g., binary, decimal, hexadecimal)**.

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		DEL

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	β	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	â	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	⊙	249	.
138	è	154	Ü	170	⌋	186	⌨	202	⌚	218	⌣	234	Ω	250	.
139	ï	155	•	171	¼	187	〈	203	⌛	219	■	235	δ	251	√
140	î	156	£	172	½	188	〉	204	⌜	220	■	236	∞	252	∞
141	ì	157	¥	173	¡	189	⌫	205	=	221	■	237	φ	253	²
142	Ä	158	£	174	«	190	⌬	206	⌟	222	■	238	ε	254	■
143	Å	159	f	175	»	191	⌭	207	⌚	223	■	239	∩	255	

Source: www.LookUpTables.com



(Sep. 05, 2013 Stops Here)



(Sep. 10, 2013 Starts Here)

A Short Review



- Octal numerals. What is $(31)_8$?
- Hexadecimal numerals. What is $(31)_{16}$?
- ASCII table: pair characters and symbols with integers, which can be expressed by decimals and binaries.

Example of Encoding

Question:

Encoding “Hello!” to binary sequences

String broken to characters

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

Steps:

- (1) Break into six characters, including the exclamation mark.
- (2) Look up the ASCII table, find the **decimal number** of the character
- (3) 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.

Steps:

- (1) Convert the binary to decimal
- (2) Look up the ASCII table, find corresponding character
- (3) Assemble the character into a string

Exercise

Answer:

ASCII	J	Z
Decimal	74	90
Binary	0100 1010	0101 1010

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

Logic Circuit & Logic Table

(Sep. 10, 2013)

Questions?

- How does a computer do computing?
- How does a computer calculate $(011)_2 + (101)_2$?
- What are logic gates?
- What is logic operation? What is Boolean algebra?

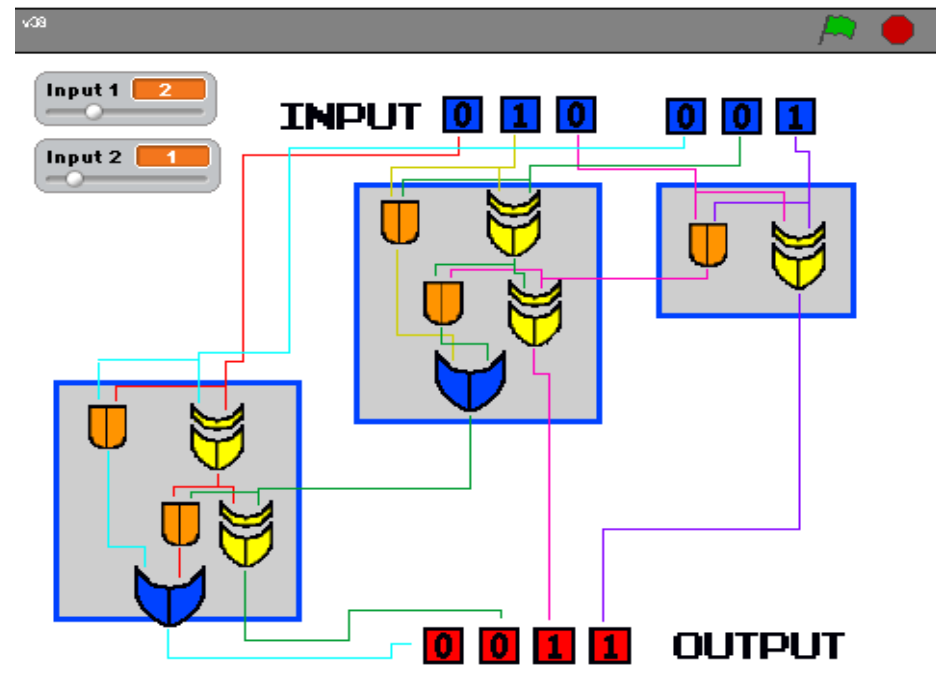
CH4: Objectives

- Understanding computing at the basic level
- Understanding logic gates, logic table, logic operation, Boolean algebra and logic circuits
- Understanding how an adding machine is implemented using basic logic gates

Logic Circuit

- **Logic Circuit**: a computer circuit that consists of a number of logic gates and performs **logical operations** on data.
- Logic circuits carry out computing processes, such as adding, subtraction and word processing

**A three-bit
binary adding
machine**





(Sep. 10, 2013 Stops Here)



(Sep. 12, 2013 Starts Here)

A Short Review



- How does a computer store “Hello” in the disk?
- What does “01001010 01011010” mean in characters?
- What is a logic circuit?

Logic Operation

- A calculation in which the quantity being operated on and results of the calculation **can have only two values: true (“1”), or false (“0”)**

True: 1

False: 0

- Logic operation is also called **Boolean algebra**

Boolean Algebra

There are three basic Boolean operations:
(1) AND, (2) OR and (3) NOT

e.g. True AND True = True
True AND False = False

NOT True = False

NOT False = True

Boolean Algebra: AND

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

$$A \text{ AND } B$$
$$A \cdot B$$

Boolean Algebra:
multiplication

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

AND operation
logic table

- 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

Boolean Algebra: OR

$$A + B$$

Boolean Algebra:

Addition

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


OR logic table

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

Boolean Algebra: NOT



**Negated,
Reversed**

INPUT	OUTPUT
A	NOT A
	1
1	0

Logic table

- The operation of reversing the input state

Logic Gate

A logic gate is a physical device (the hardware) that implements a Boolean operation

There are six basic logic gates, which are

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

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).

Type - OR gate



$$A + B$$

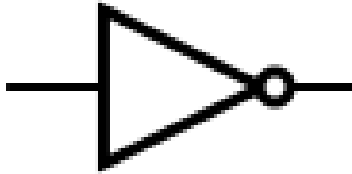
Symbol

Boolean Algebra:
addition

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

Logic
table

Type - NOT gate



Symbol

$$\bar{A}$$

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.

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

- 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

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

Type - NAND gate

Prove:

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

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

A	B	\bar{A}	\bar{B}	$\bar{A} + \bar{B}$
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

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

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

•XOR stands for "Exclusive OR". The thing to remember about EOR gates is this: An XOR 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

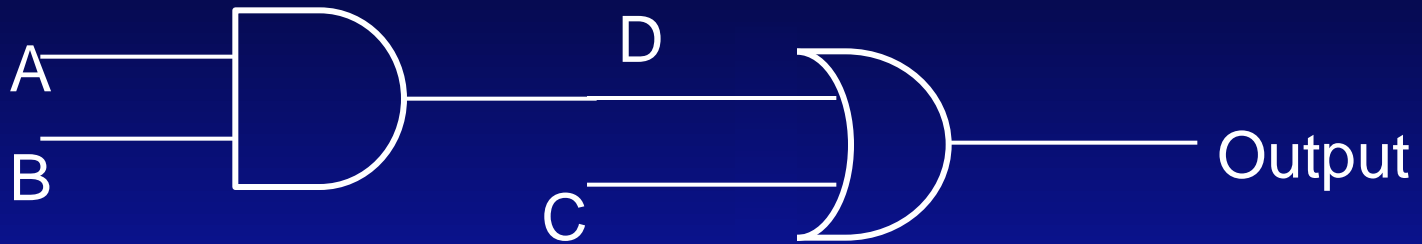
NOR: If ANY input to the NOR gate is 1, then it's output is 0

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



(Sep. 17, 2013 Stops Here)



(Sep. 19, 2013 Starts Here)

Quiz - 2

A Short Review



- Logic Operation

- (1) AND

- (2) OR

- (3) NAND

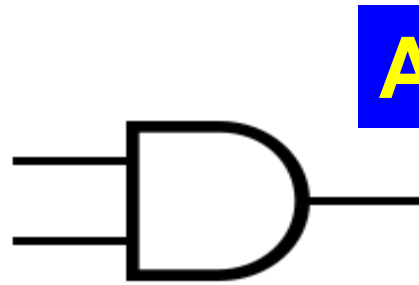
- (4) NOR

- (5) NOT

- (6) XOR

- Logic Table

(cont.)



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

(cont.)



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

$$\overline{A + B}$$

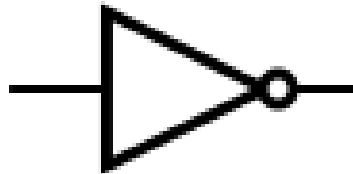
Boolean Algebra

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



(cont.)

NOT



Symbol

$$\bar{A}$$

Boolean Algebra:
Negated,
Reversed

INPUT		OUTPUT
A		NOT A
		1
1		0

XOR



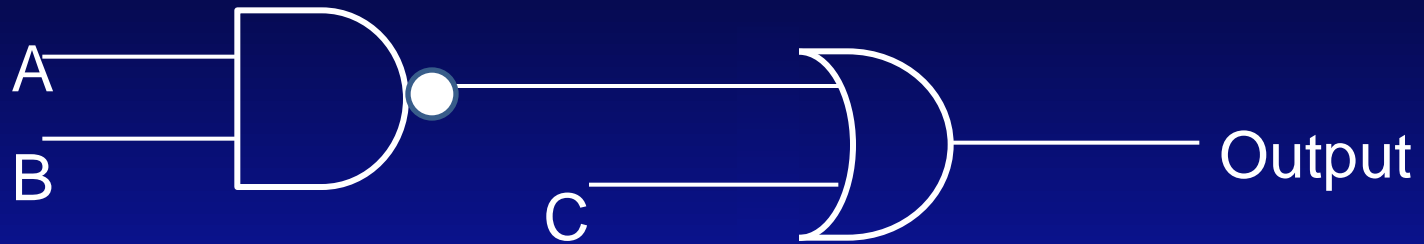
Symbol

$$A \oplus B$$

Boolean Algebra:
Exclusive OR

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

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



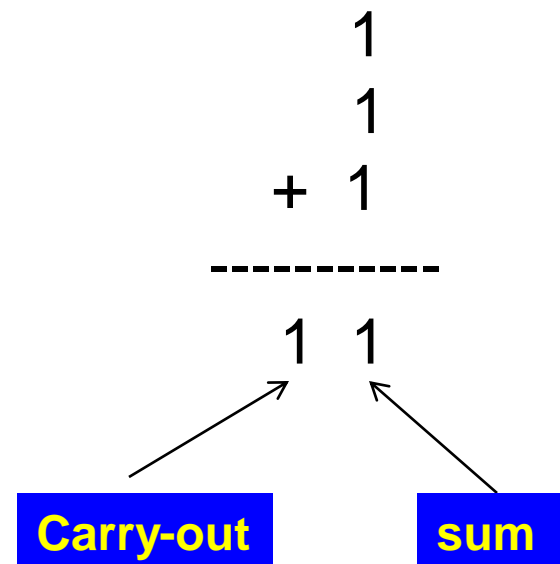
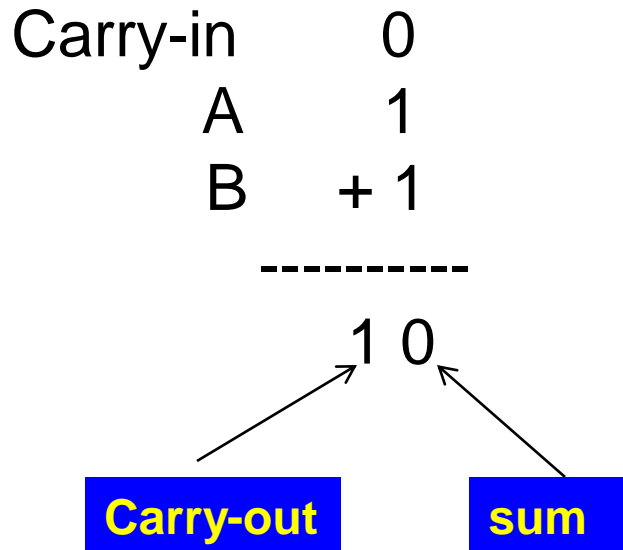
A	B	C	D	Output
0	0	0	1	1
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	0	0
1	1	1	0	1

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

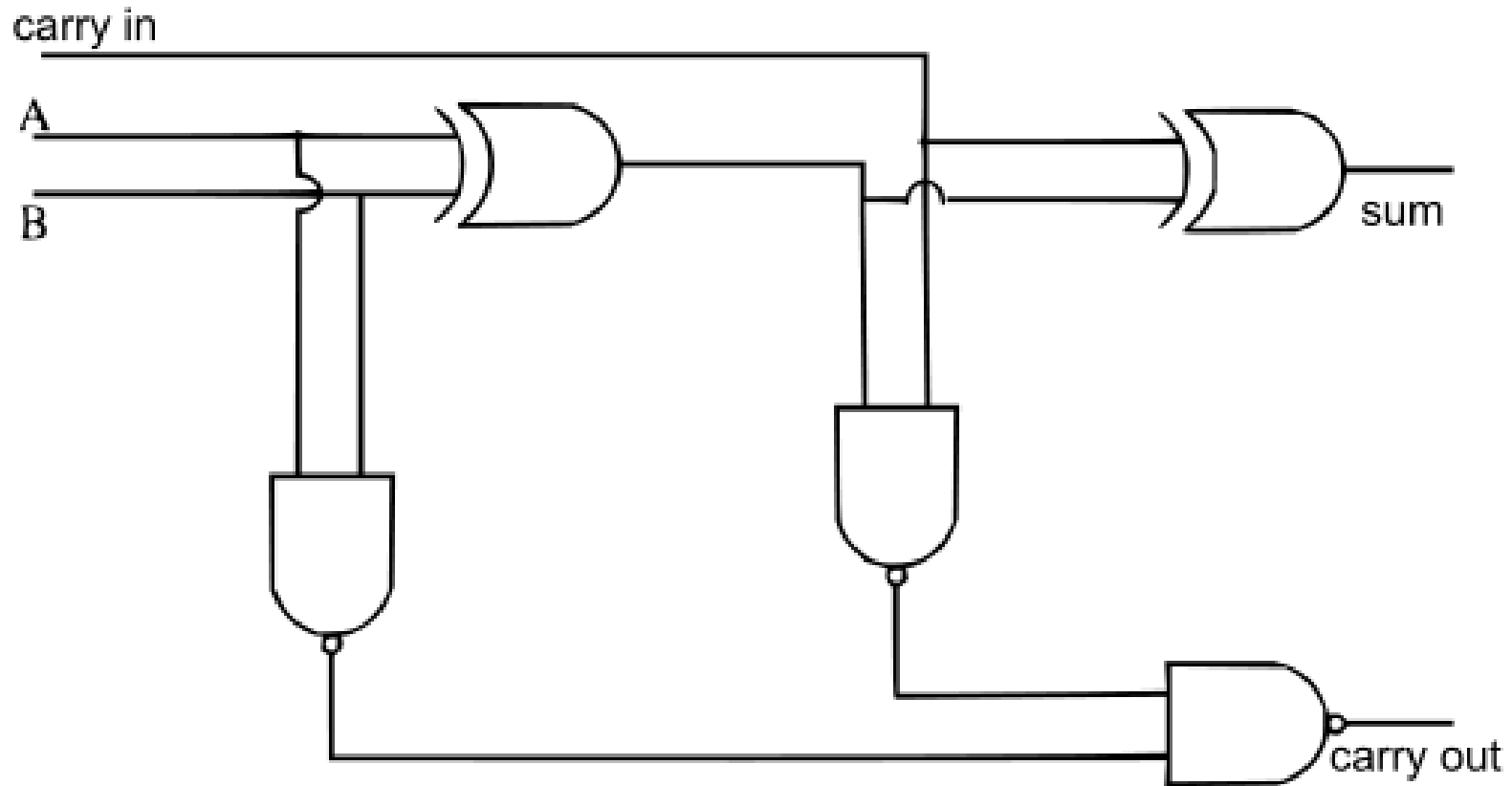


One-bit Adding Machine

The Logic Table: three inputs, two outputs

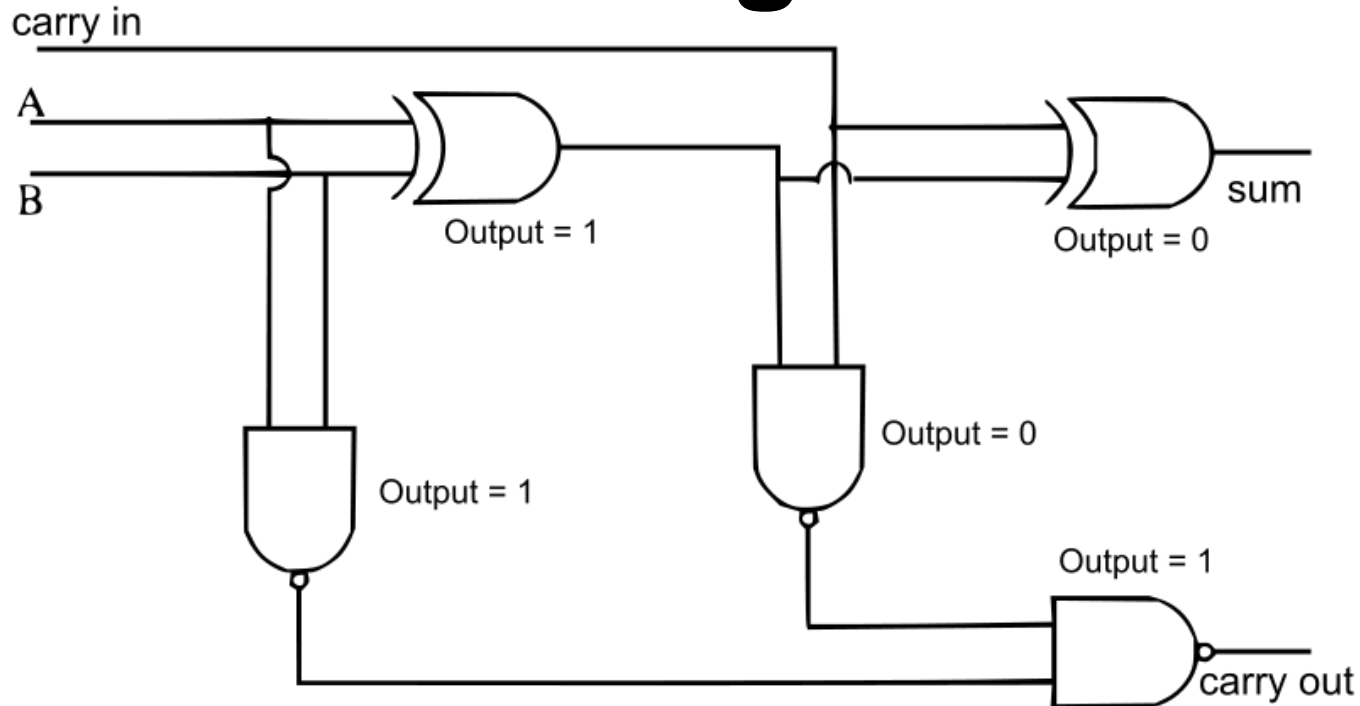
Carry-in	A	B	SUM	Carry-out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

1-bit Adding Machine



Two XOR gates and three NAND gates

1-bit Adding Machine



If $A = 1$, $B = 0$, and $\text{carry in} = 1$:

The upper left EOR gate output is 1

The lower left NAND gate output is 1

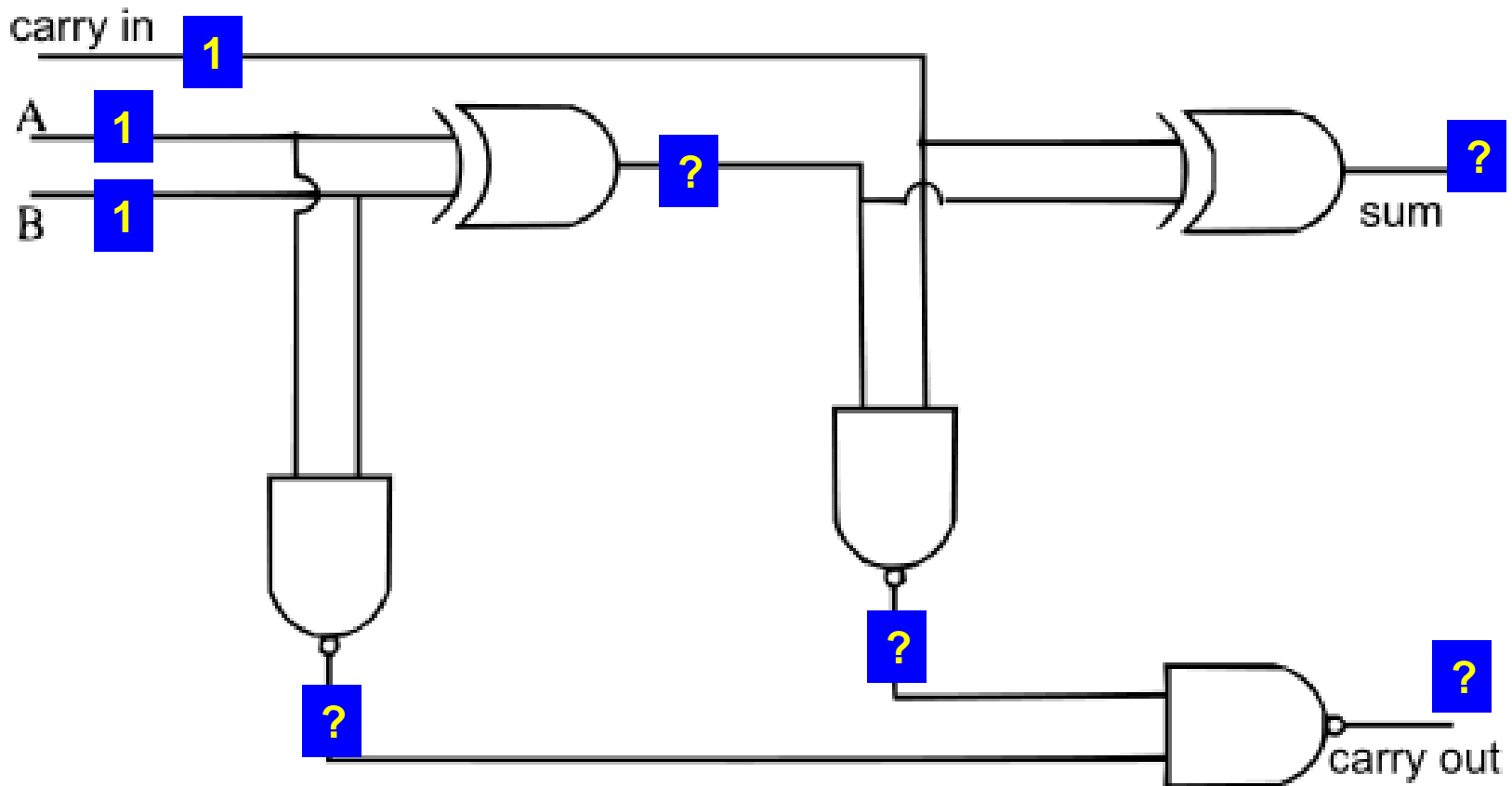
The center NAND gate output is 0

The upper right EOR gate (sum) output is 0

The lower right NAND gate (carry out) output is 1

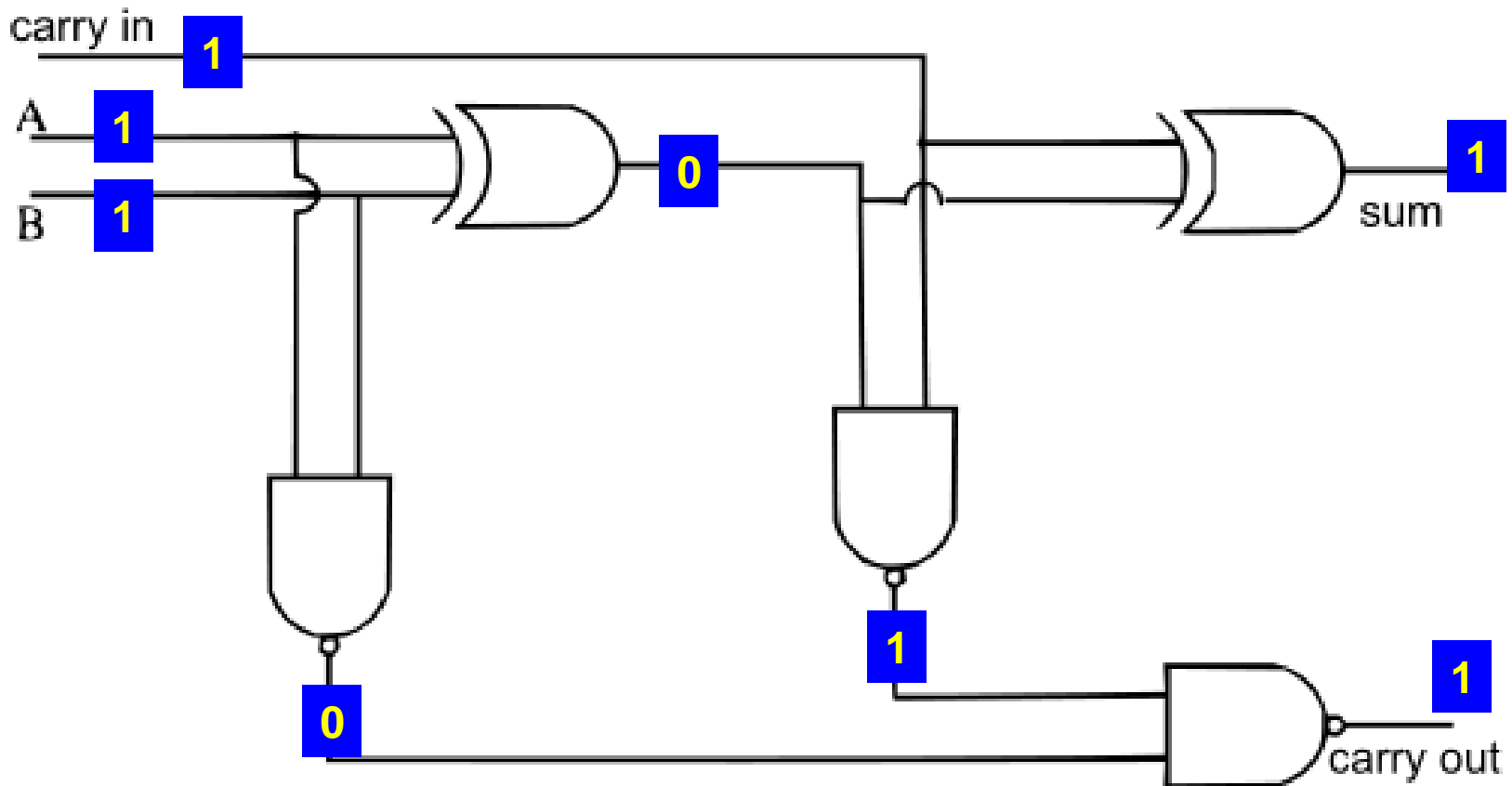
1-bit Adding Machine

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



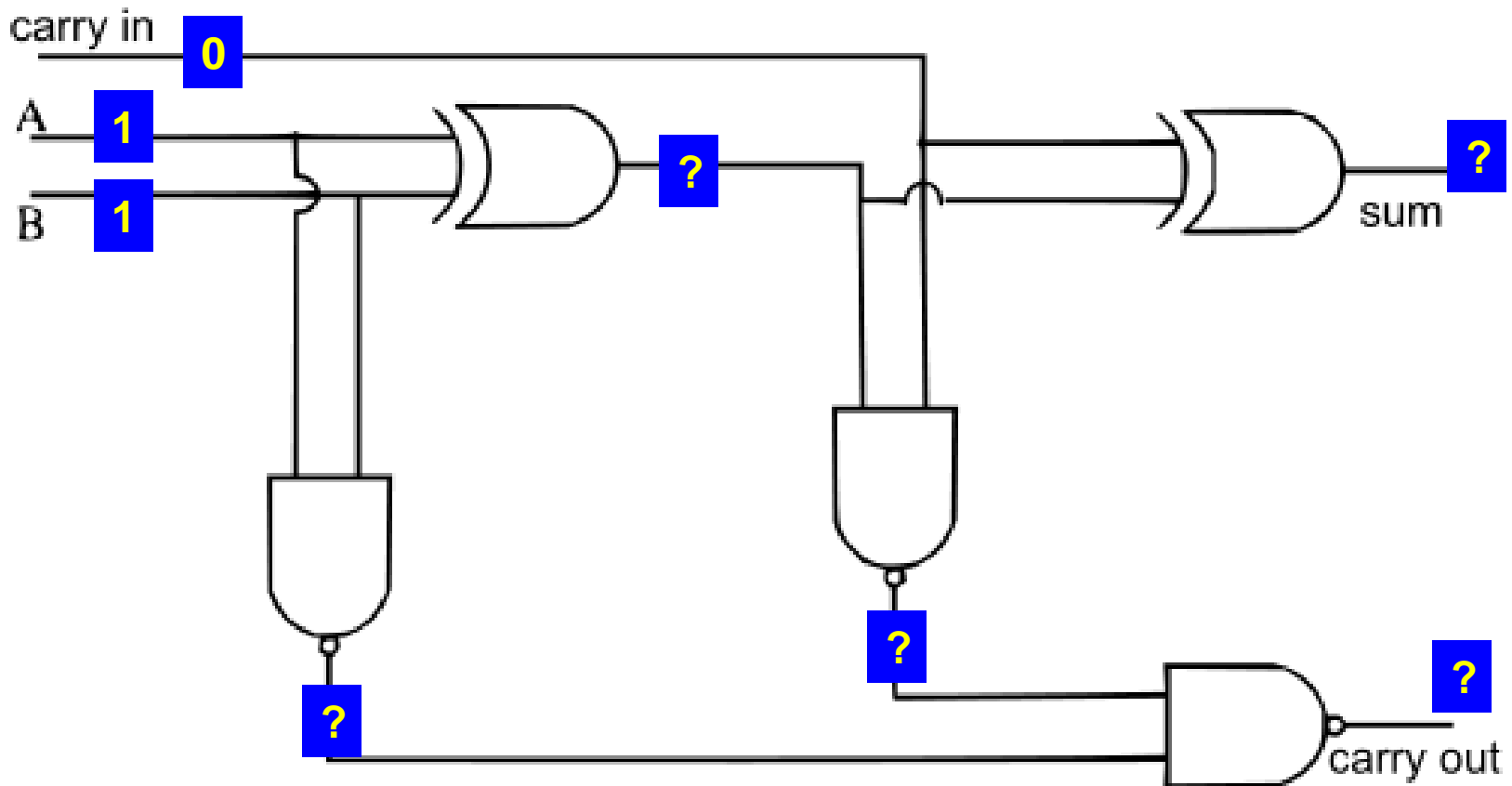
1-bit Adding Machine

Answer:



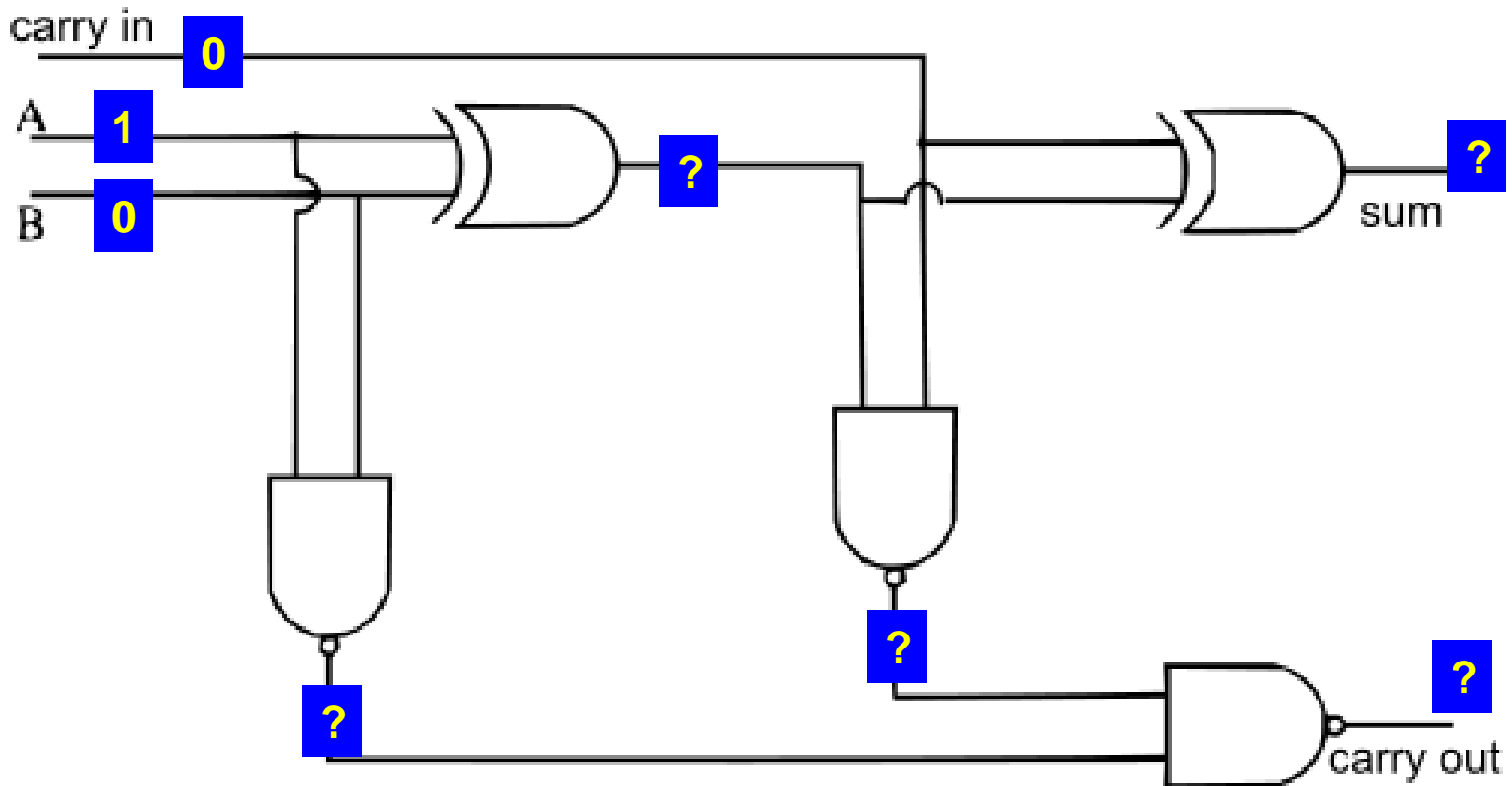
1-bit Adding Machine

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

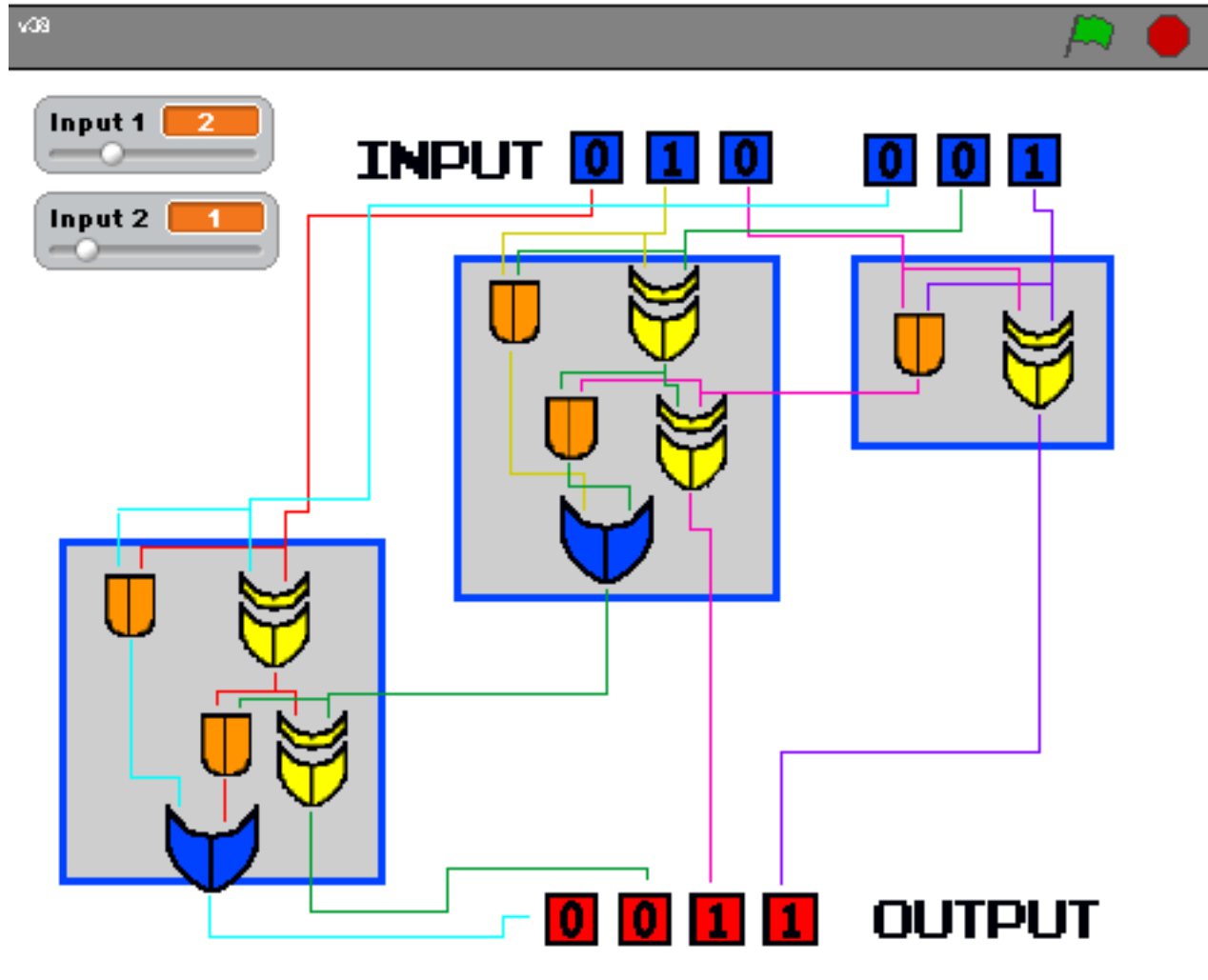


1-bit Adding Machine

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>



The End of Chapter 4



Section 1

Chapter 5

Data Storage

(Sep. 17, 2013)

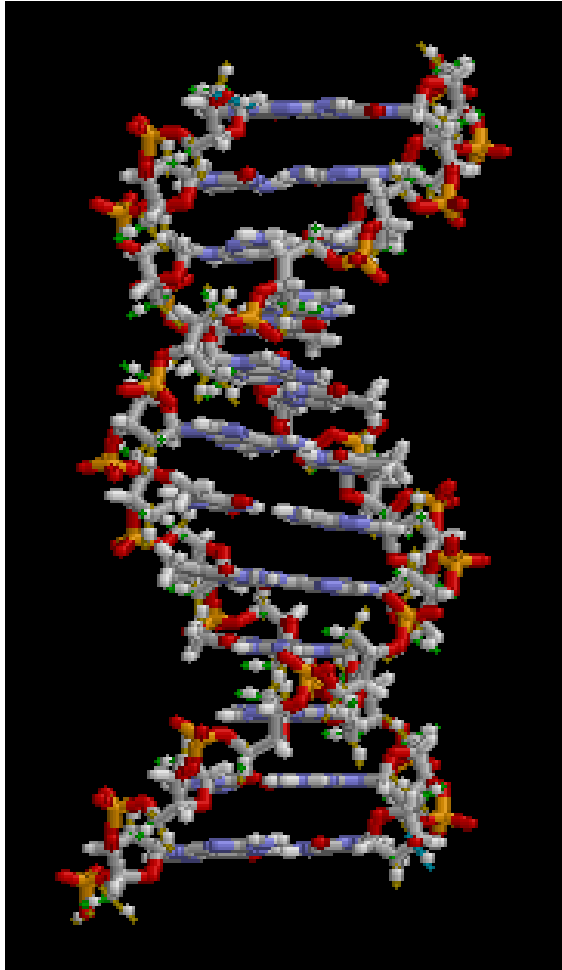
Questions?

- How many books does a 16-GB memory stick hold?
- How many songs does a 32-GB cell phone hold?
- What is the difference between one bit and one byte?
- What does a 24-bit monitor mean?
- What does “long integer” mean?

CH5: Objectives

- Understand different types of electronic data storages
- How to calculate data density?
- Understand bit patterns
- Understand how transistors work

Data Storage



DNA: deoxyribonucleic acid

Data Storage

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)



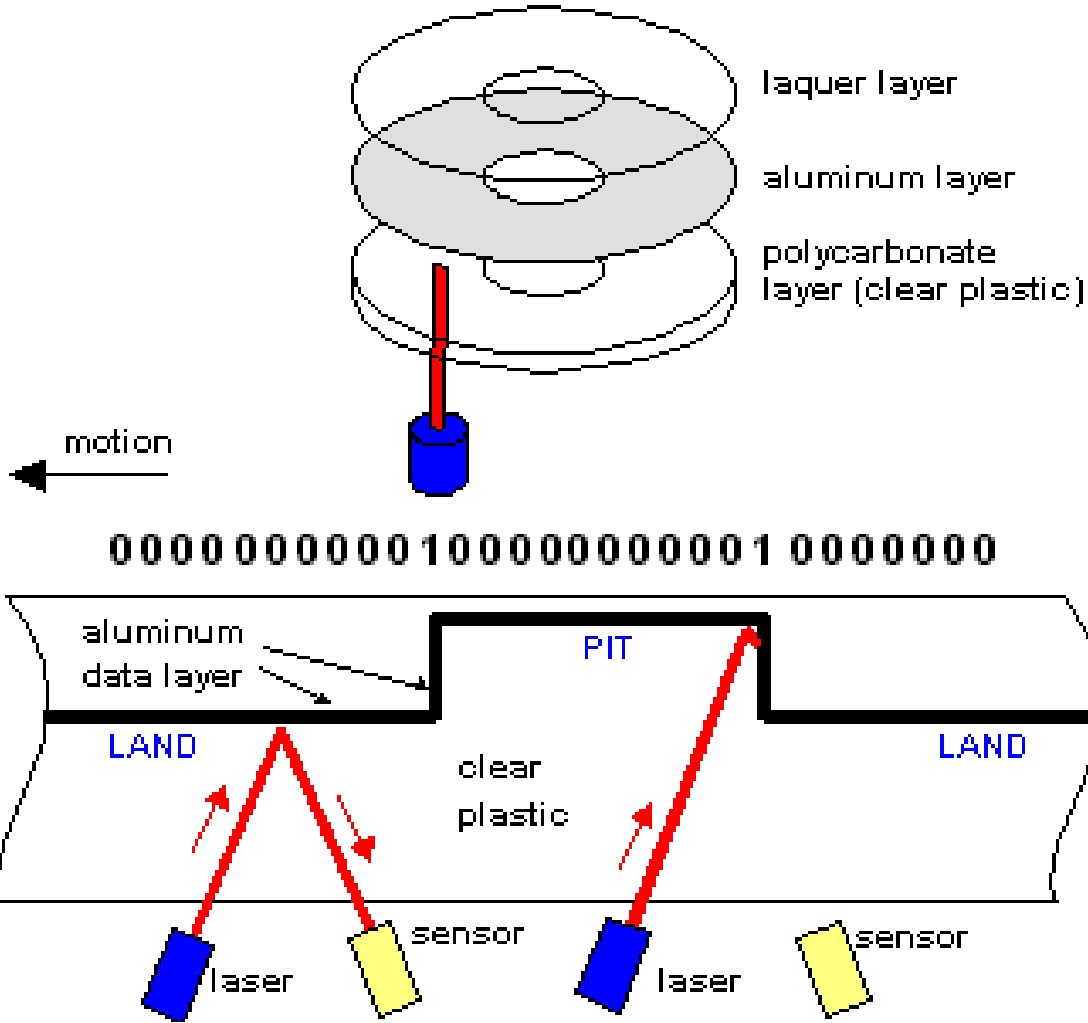
HDD



CCD

How CD-ROM work?

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

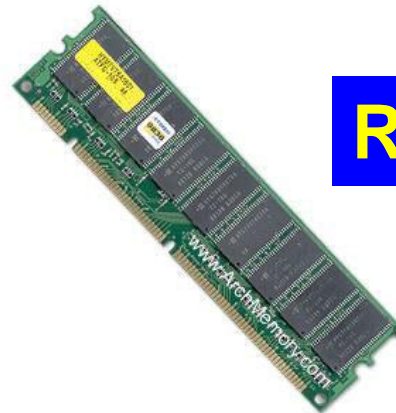


Data Storage in Computers

- Using silicon-based semi-conductor material
 - Building block: **transistors**
 - 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

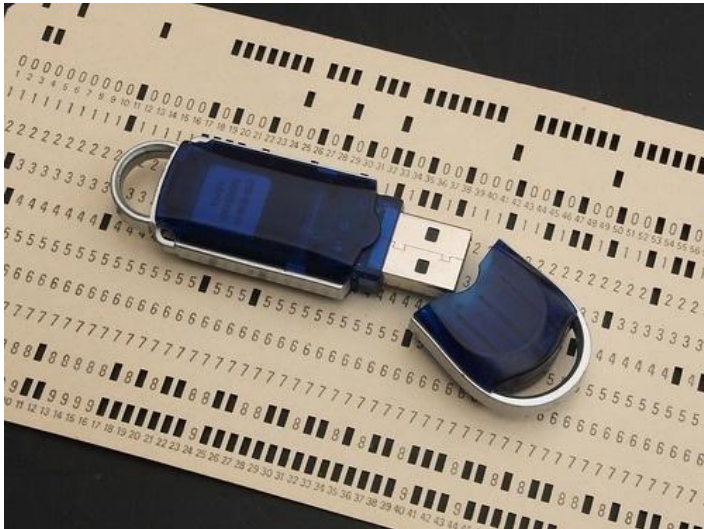
Data Density (character/inch²)

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)





(Sep. 17, 2013 Stops Here)

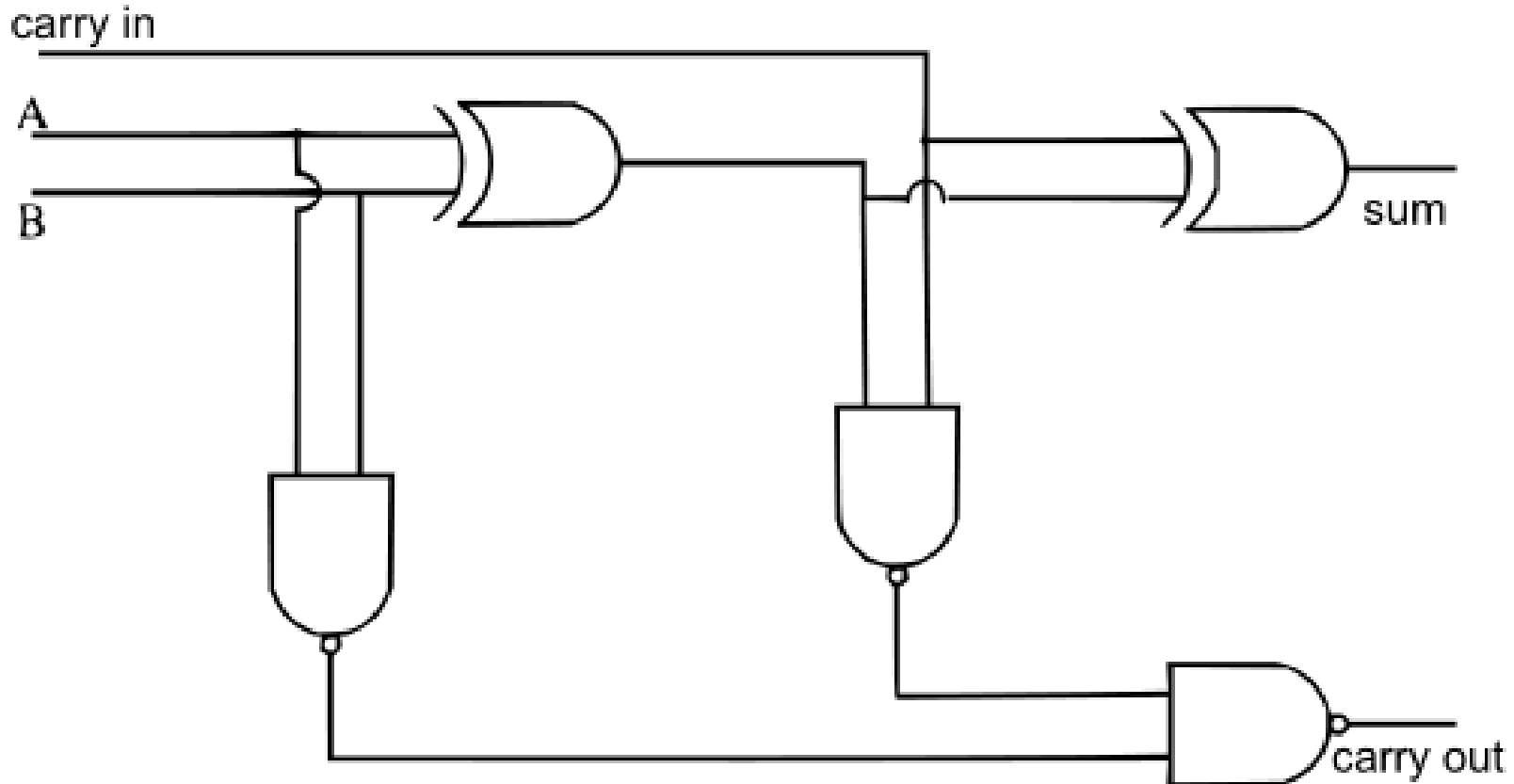


**(Sep. 19, 2013 Class
Cancelled Due to Campus
Power Outage)**



(Sep. 24, 2013 Starts Here)

A Short Review



1-bit Adding Machine

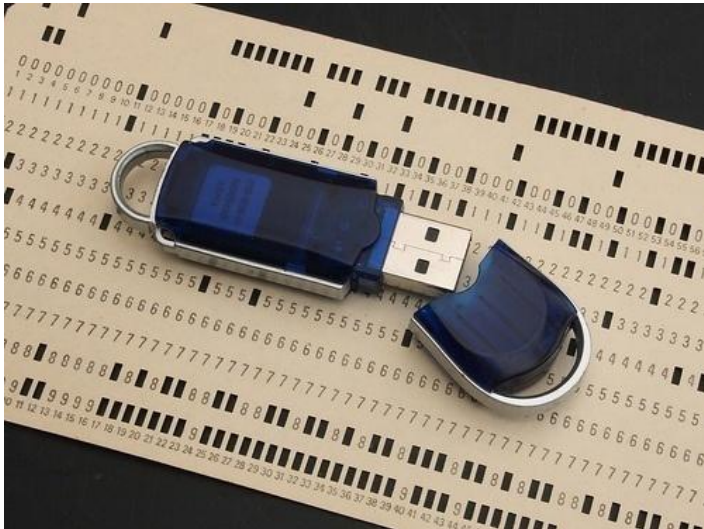
Data Density (character/inch²)

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

1 KB = 1024 Byte = 10^3 Byte

1 MB = 1 k x 1 k = 10^6 Byte

1 GB = 1 k x 1 k x 1 k = 10^9 Byte

1 TB = 1 k x 1 k x 1 k x 1k = 10^{12} Byte

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

Bit

- A bit is the individual zero and one that are stored by computer hardware
- A bit can have two different states (0 or 1), and is a single digit long.

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
 - **One ASCII character**

Other Types

Short Integer: 8 bits (max: $255 = 256 - 1$)

Integer: 16 bits (max: $65535 = 256 \times 256 - 1$)

Long Integer: 32 bits (max = $65536 \times 65536 - 1$)

Floating Point: 32 bits (max: $\sim 10^{38.53}$)

Double Floating Point: 64 bits (max: $\sim 10^{308.3}$)

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

$$2^8 = 256$$

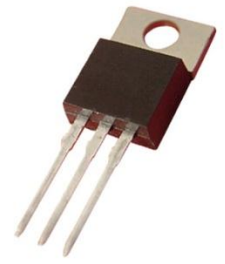
**24-bit monitor: support 1.7 million different colors,
the true color**

$$2^{24} = 256 \times 256 \times 256 = 1677216$$

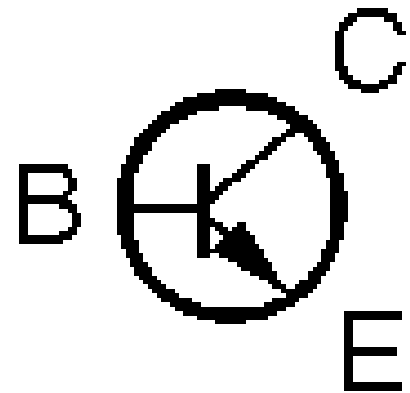
Transistors:

**the building block of modern
electronic devices**

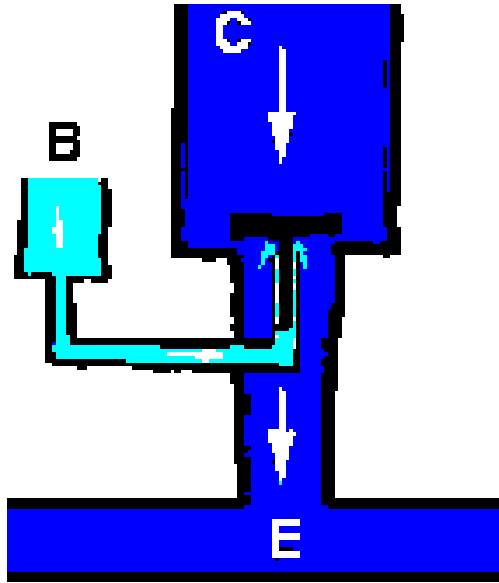
Transistors



- A transistor is a device used to amplify and switch electronic signals
- A transistor has three pins
 - Collector (C): acting as a source (power source)
 - Base (B): acting as a gate
 - Emitter (E): acting as a drain (the output)

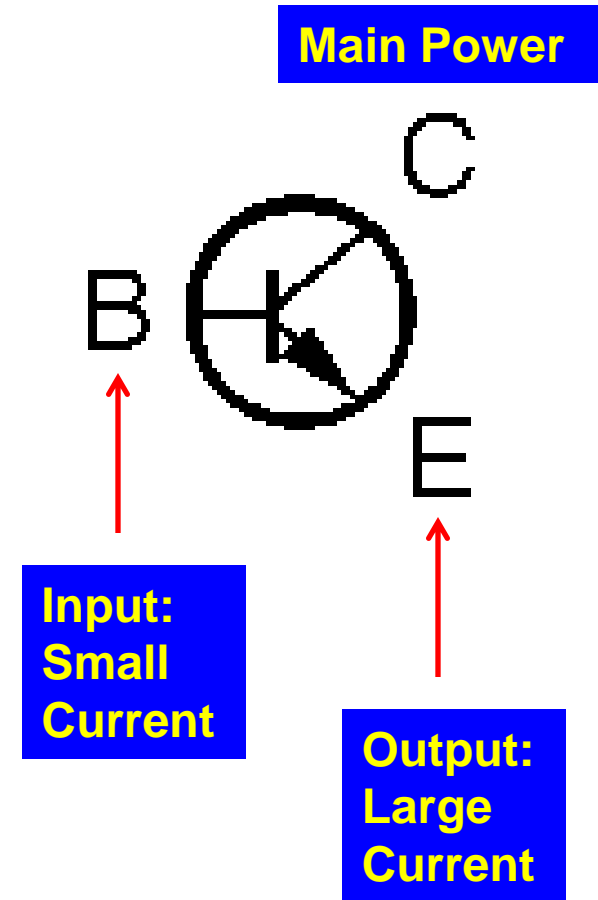


The Hydraulic Analogy



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

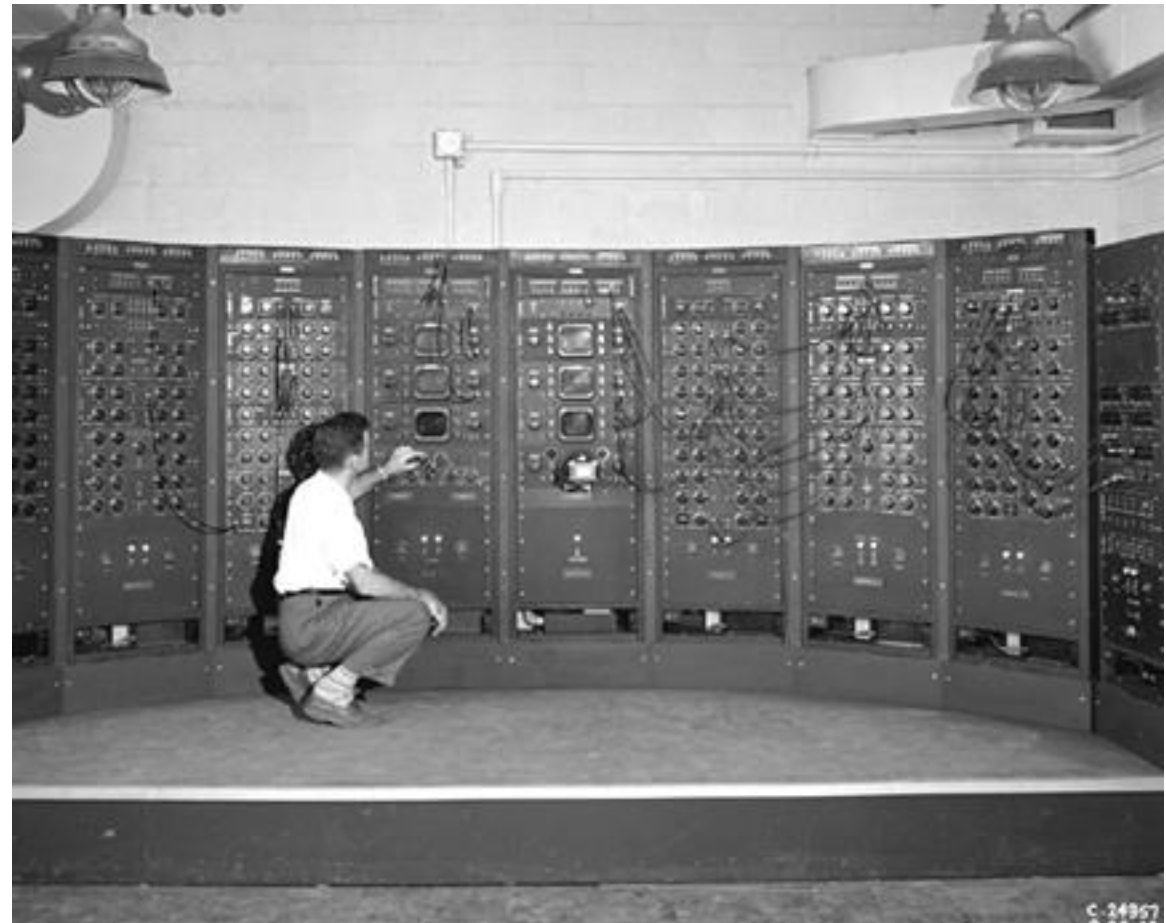
- It is often easier to think of electronic current flow in terms of water flow.
- A small water flow from B pushes the black plunger upward, allowing a large flow from C to E. This is the amplification.



Transistors

- When a small current is applied between Base and Emitter, 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 states through other transistors, which is essential for complex circuit and computing**

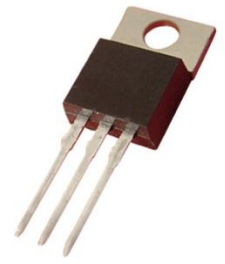
Transistor: Prior Digital Era



**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



- Modern transistors are made of **silicon-based semiconductor**
- Semiconductor: a material has dual properties of a pure conductor and a pure insulator. It is the foundation of modern electronic device.

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?

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.

Computer

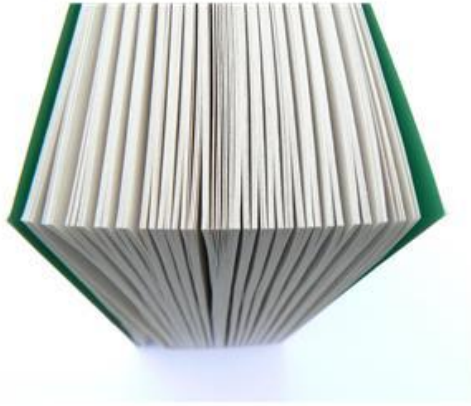
A computer is a machine that manipulates data according to a set of instructions called a computer program

--<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; this is how to turn “fast” into “smart”



The End of Chapter 5



The End of Session 1

Computer Fundamentals (Sep. 24, 2013)