

How do computers compute?

Representing data

- Representing all data with 1s and 0s, on/off, true/false
- Numbers -> binary (positional like decimal)
- Letters/symbols -> ASCII, Unicode, UTF8

Positional Notation

- Binary numbers, like decimal numbers, use *place notation*

$$\mathbf{1101 = 1 \times 1000 + 1 \times 100 + 0 \times 10 + 1 \times 1}$$

$$= \mathbf{1 \times 10^3 + 1 \times 10^2 + 0 \times 10^1 + 1 \times 10^0}$$

except that the base is 2 not 10

$$\mathbf{1101 = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1}$$

$$= \mathbf{1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0}$$

Base or
radix

1101 in binary is 13 in decimal

Binary to Decimal

- What is the decimal equivalent of the binary number 1101?
- A. 10
- B. 11
- C. 12
- D. 13
- E. 14

Decimal to Binary

- What is the binary representation of the decimal value 21?
- A. 10010
- B. 10100
- C. 10101
- D. 10110
- E. 10111

Binary combinations, True/False possibilities

- One bit
 - 0
 - 1
- Two Bits
 - 00
 - 01
 - 10
 - 11
- Three bits
 - 000
 - 001
 - 010
 - 011
 - 100
 - 101
 - 110
 - 111
- Four bits
 - 0000
 - 0001
 - 0010
 - 0011
 - 0100
 - 0101
 - 0110
 - 0111
 - 1000
 - 1001
 - 1010
 - 1011
 - 1100
 - 1101
 - 1110
 - 1111

Binary Addition

0	0	1	1
<u>+0</u>	<u>+1</u>	<u>+0</u>	<u>+1</u>

Binary Addition

$$\begin{array}{r} 001101 \\ +010111 \\ \hline \end{array}$$

Binary Addition

$$\begin{array}{r} 011101 \\ +010011 \\ \hline \end{array}$$

- A. 101010
- B. 110101
- C. 110000
- D. 111000
- E. 101101

Bytes

- A byte is eight bits treated as a unit
 - Adopted by IBM in 1960s
 - A standard measure ever since
 - Bytes encode the Latin alphabet using ASCII -- the American Standard Code for Information Interchange



```
0101 0101  
0101 0111
```

ASCII

ASCII	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0000	N _U	S _H	S _X	E _X	E _T	E _Q	A _K	B _L	B _S	H _T	L _F	Y _T	F _F	C _R	S ₀	S _I
0001	D _L	D ₁	D ₂	D ₃	D ₄	N _K	S _Y	E _Σ	C _N	E _M	S _B	E _C	F _S	G _S	R _S	U _S
0010		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
0011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0101	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
0110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0111	p	q	r	s	t	u	v	w	x	y	z	{		}	~	D _T
1000	°	° ₁	° ₂	° ₃	I _N	N _L	S _S	E _S	H _S	H _J	Y _S	P _D	P _V	R _I	S ₂	S ₃
1001	D _C	P ₁	P _Z	S _E	C _C	M _M	S _P	E _P	O ₈	O _Q	O _A	C _S	S _T	O _S	P _M	A _P
1010	°	i	¢	£	♀	¥		§	..	©	♂	«	¬	-	®	—
1011	°	±	²	³	´	μ	¶	·	¸	¹	º	»	¼	½	¾	¿
1100	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
1101	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
1111	đ	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

0100 0011
0101 0011
0101 0000

0100 0111 | 0110 1111 | 0010 0000 | 0101 0011 | 0110 1100 | 0111 0101 | 0110 0111 | 0111 0011

ASCII

What is the first letter of the message at the bottom?

- A. G
- B. t

ASCII	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0000	N _U	S _H	S _X	E _X	E _T	E _Q	A _K	B _L	B _S	H _T	L _F	Y _T	F _F	C _R	S ₀	S _I
0001	D _L	D ₁	D ₂	D ₃	D ₄	N _K	S _Y	E _Σ	C _N	E _M	S _B	E _C	F _S	G _S	R _S	U _S
0010		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
0011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0101	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
0110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0111	p	q	r	s	t	u	v	w	x	y	z	{		}	~	D _T
1000	8 ₀	8 ₁	8 ₂	8 ₃	I _N	N _L	S _S	E _S	H _S	H _J	Y _S	P _D	P _V	R _I	S ₂	S ₃
1001	D _C	P ₁	P _Z	S _E	C _C	M _M	S _P	E _P	O ₈	O _Q	O _A	C _S	S _T	O _S	P _M	A _P
1010	A _o	i	ç	£	♀	¥		§	..	©	♂	«	¬	-	®	—
1011	°	±	²	³	´	μ	¶	·	¸	¹	º	»	¼	½	¾	¿
1100	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
1101	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
1111	đ	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

0100 0111|0110 1111|0010 0000|0101 0011|0110 1100|0111 0101|0110 0111|0111 0011

UTF-8

Uniform

Transformation

Format for bytes

(UTF-8) is

universal ... all

characters have a

place: 1-4 Bytes

(Unicode has space for over
1 million symbols)

لماذا لا يتكلمون اللّغة العربية فحسب؟

Защо те просто не могат да говорят **български**?

Per què no poden simplement parlar en **català**? 🗣️

他們爲什麼不說中文（台灣）？ 🗣️ 🗣️

Proč prostě nemluví **česky**?

Hvorfor kan de ikke bare tale **dansk**?

Warum sprechen sie nicht einfach **Deutsch**? 🗣️

Μα γιατί δεν μπορούν να μιλήσουν **Ελληνικά**; 🗣️

Why can't they just speak English?

¿Por qué no pueden simplemente hablar en **castellano**? 🗣️

Miksi he eivät yksinkertaisesti puhu **suomea**?

Pourquoi, tout simplement, ne parlent-ils pas **français** ? 🗣️

למה הם פשוט לא מדברים **עברית**?

Miért nem beszélnek egyszerűen **magyarul**?

Af hverju geta þeir ekki bara talað **íslensku**?

Perché non possono semplicemente parlare **italiano**? 🗣️

なぜ、みんな日本語を話してくれないのか？ 🗣️

세계의 모든 사람들이 한국어를 이해한다면 얼마나 좋을까? 🗣️

Waarom spreken ze niet gewoon **Nederlands**? 🗣️

Hvorfor kan de ikke bare snakke **norsk**?

Dlaczego oni po prostu nie mówią po **polsku**? 🗣️

Porque é que eles não falam em **Português (do Brasil)**?

Oare ăștia de ce nu vorbesc **românește**?

Почему же они не говорят **по-русски**?

Zašto jednostavno ne govore **hrvatski**?

Pse nuk duan të flasin vetëm **shqip**?

Varför pratar dom inte bara **svenska**? 🗣️

ทำไมเขาถึงไม่พูดภาษาไทย

Neden **Türkçe** konuşuyorlar?

UTF-8

Bits of code point	First code point	Last code point	Bytes in sequence	Byte 1	Byte 2	Byte 3	Byte 4
7	U+0000	U+007F	1	0xxxxxxx			
11	U+0080	U+07FF	2	110xxxxx	10xxxxxx		
16	U+0800	U+FFFF	3	1110xxxx	10xxxxxx	10xxxxxx	
21	U+10000	U+1FFFFFF	4	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

UTF-8

Bits of code point	First code point	Last code point	Bytes in sequence	Byte 1	Byte 2	Byte 3	Byte 4
7	U+0000	U+007F	1	0xxxxxxx			
11	U+0080	U+07FF	2	110xxxxx	10xxxxxx		
16	U+0800	U+FFFF	3	1110xxxx	10xxxxxx	10xxxxxx	
21	U+10000	U+1FFFFFF	4	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

What is the first Unicode value represented by this sequence?
11101010 1000011 10000111 0011111 11000011 10000000

- A. 000000001101010
- B. 0000000011101010
- C. 0000001010000111
- D. 1010000011000111

UTF-8

Bits of code point	First code point	Last code point	Bytes in sequence	Byte 1	Byte 2	Byte 3	Byte 4
7	U+0000	U+007F	1	0xxxxxxx			
11	U+0080	U+07FF	2	110xxxxx	10xxxxxx		
16	U+0800	U+FFFF	3	1110xxxx	10xxxxxx	10xxxxxx	
21	U+10000	U+1FFFFFF	4	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

How many Unicode characters are represented by this sequence?

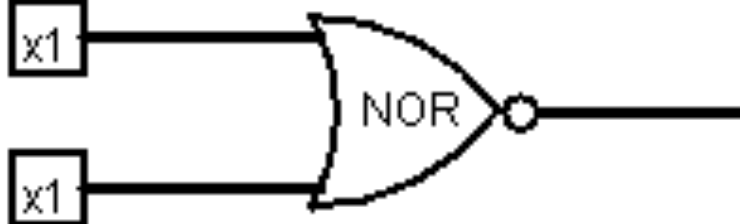
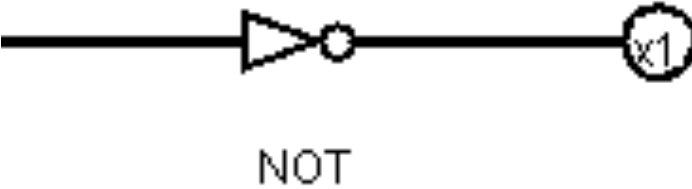
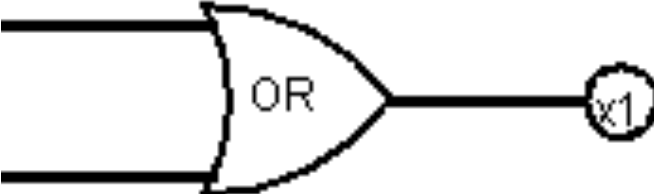
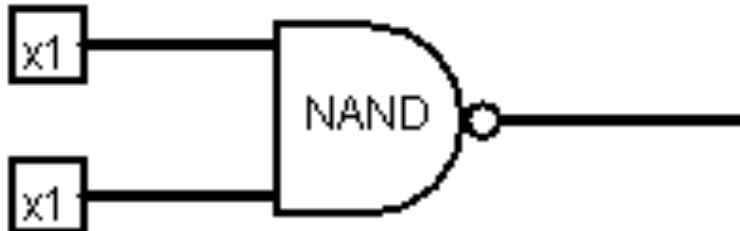
11101010 10000011 10000111 00111111 11000011 10000000

- A. 1
- B. 2
- C. 3
- D. 4
- E. 5




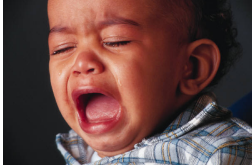



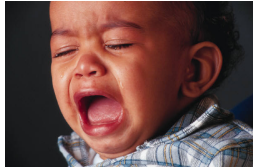
But how do they compute?

- Basic operations are arithmetic, compare, read/write memory, test and jump (to a different memory location for the next instruction).

Fundamental units of computers: Logic Gates



Truth Table for *And* (using *True* and *False*)

P		Q		P and Q
True		True		True
True		False		False
False		True		False
False		False		False

Truth Table for *And* (using 0 and 1)

P	Q	P and Q
1	1	
1	0	
0	1	
0	0	

Truth Table for *Or* (using 0 and 1)

P	Q	P or Q
1	1	
1	0	
0	1	
0	0	

Truth Table for *Not And* (using 0 and 1)

P	Q	P AND Q	NOT (P AND Q)
1	1	1	0
1	0	0	1
0	1	0	1
0	0	0	1

$(\text{NOT } P) \text{ OR } (\text{NOT } Q)$ vs. $\text{NOT } (P \text{ AND } Q)$

P	Q	NOT P	NOT Q	P AND Q	NOT (P AND Q)	(NOT P) OR (NOT Q)	
1	1	0	0	1	0	0	
1	0	0	1	0	1	1	
0	1	1	0	0	1	1	
0	0	1	1	0	1	1	

(NOT P) OR (NOT Q) vs. NOT (P AND Q)

P	Q	NOT P	NOT Q	P AND Q	NOT (P AND Q)	(NOT P) OR (NOT Q)
1	1	0	0	1	0	0
1	0	0	1	0	1	1
0	1	1	0	0	1	1
0	0	1	1	0	1	1

- $\text{NOT } (P \wedge Q) = \text{NOT } P \vee \text{NOT } Q$
- This is DeMorgan's Law of Boolean Algebra

P and Q or R

- What is P and Q or R if P is true, Q is false, and R is true?
- A. True
- B. False

Exclusive-OR == XOR

- Consider two propositions, either of which may be true or false
- Exclusive-or is the relationship between them when **JUST ONE OF THEM** is true.
- It **EXCLUDES** the case when both are true, so exclusive-or of the two is...
- False when both are true or both are false, and true in the other two cases.

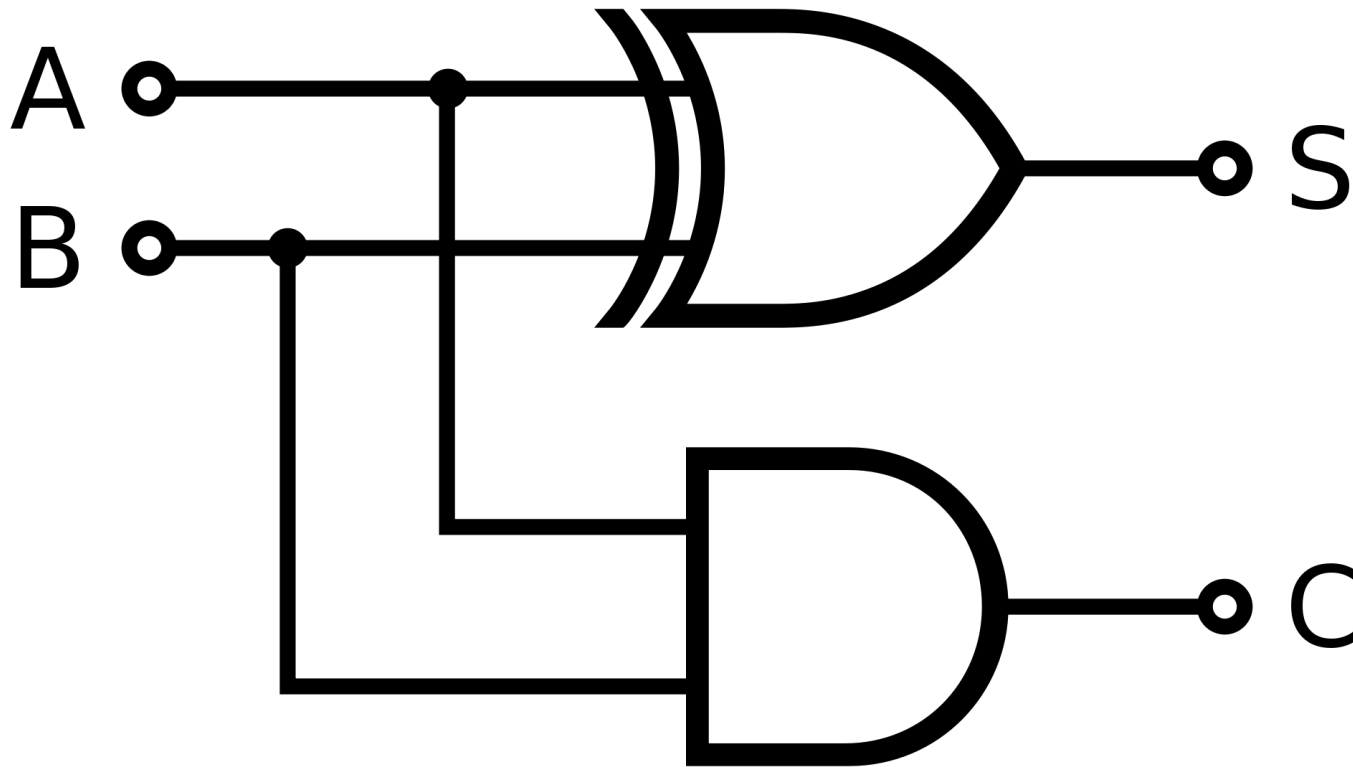
Truth Table for XOR

(using 0 and 1)

P	Q	P xor Q
1	1	
1	0	
0	1	
0	0	

What would you ever want XOR for anyway?

- [http://en.wikipedia.org/wiki/Adder_\(electronics\)](http://en.wikipedia.org/wiki/Adder_(electronics))
- Binary Addition



Summary: It All Works Because of Digital

- Key principle: information is represented as simply the presence or absence of a phenomenon at a given place and time!
- Phenomenon in computers: Electrical output on a line
 - Hole in punch card, early example.
- Logic Gates
 - Charge on line
 - No charge