CS101 Introduction to Computing

Lecture 8

Binary Numbers & Logic Operations



The focus of the last lecture was on the microprocessor

- During that lecture we learnt about the function of the central component of a computer, the microprocessor
- And its various sub-systems
 - Bus interface unit
 - Data & instruction cache memory
 - Instruction decoder
 - ALU
 - Floating-point unit
 - Control unit



Learning Goals for Today

- 1. To become familiar with number system used by the microprocessors binary numbers
- 2. To become able to perform decimal-to-binary conversions
- 3. To understand the NOT, AND, OR and XOR logic operations the fundamental operations that are available in all microprocessors



BNARY BASE 2) numbers



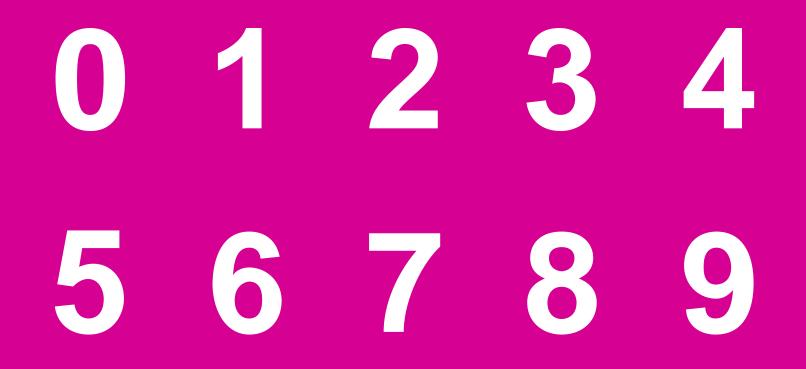
DECIMAL

(BASE 10)

numbers

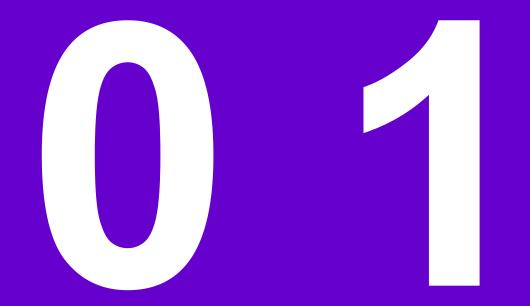


Decimal (base 10) number system consists of 10 symbols or digits





Binary (base 2) number system consists of just two





Other popular number systems

Octal

- -base = 8
- 8 symbols (0,1,2,3,4,5,6,7)
- Hexadecimal
 - -base = 16
 - 16 symbols (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)



The right-most is the least significant digit

$4202 = 2 \times 10^{0} + 0 \times 10^{1} + 2 \times 10^{2} + 4 \times 10^{3}$

The left-most is the most significant digit



$4202 = 2x10^{0} + 0x10^{1} + 2x10^{2} + 4x10^{3}$ 1's multiplier



$4202 = 2x10^{0} + 0x10^{1} + 2x10^{2} + 4x10^{3}$ 10's multiplier



$4202 = 2 \times 10^{0} + 0 \times 10^{1} + 2 \times 10^{2} + 4 \times 10^{3}$

100



$4202 = 2 \times 10^{0} + 0 \times 10^{1} + 2 \times 10^{2} + 4 \times 10^{3}$

1000's multiplier



1000

The right-most is the least significant digit

$10011 = 1 \times 2^{0} + 1 \times 2^{1} + 0 \times 2^{2} + 0 \times 2^{3} + 1 \times 2^{4}$

The left-most is the most significant digit



$10011 = 1x2^{0} + 1x2^{1} + 0x2^{2} + 0x2^{3} + 1x2^{4}$ 1's multiplier



2 $10011 = 1x2^{0} + 1x2^{1} + 0x2^{2} + 0x2^{3} + 1x2^{4}$ 2's multiplier



$10011 = 1 \times 2^{0} + 1 \times 2^{1} + 0 \times 2^{2} + 0 \times 2^{3} + 1 \times 2^{4}$



$10011 = 1 \times 2^{0} + 1 \times 2^{1} + 0 \times 2^{2} + 0 \times 2^{3} + 1 \times 2^{4}$



$10011 = 1 \times 2^{0} + 1 \times 2^{1} + 0 \times 2^{2} + 0 \times 2^{3} + 1 \times 2^{4}$



C	οι			
in	De			
0	10	20	30	0
1	11	21	31	1
2	12	22	32	10
3	13	23	33	11
4	14	24	34	100
5	15	25	35	101
6	16	26	36	110
7	17	27		111
8	18	28		1000
9	19	29		1001

Counting in Binary

1010	10100	11110
1011	10101	11111
1100	10110	100000
1101	10111	100001
1110	11000	100010
1111	11001	100011
10000	11010	100100
10001	11011	
10010	11100	
10011	44404	

TU'I

1001

When the system is natural for digital computers

The fundamental building block of a digital computer – the switch – possesses two natural states, ON & OFF.

It is natural to represent those states in a number system that has only two symbols, 1 and 0, i.e. the binary number system

In some ways, the decimal number system is natural to us humans. Why?



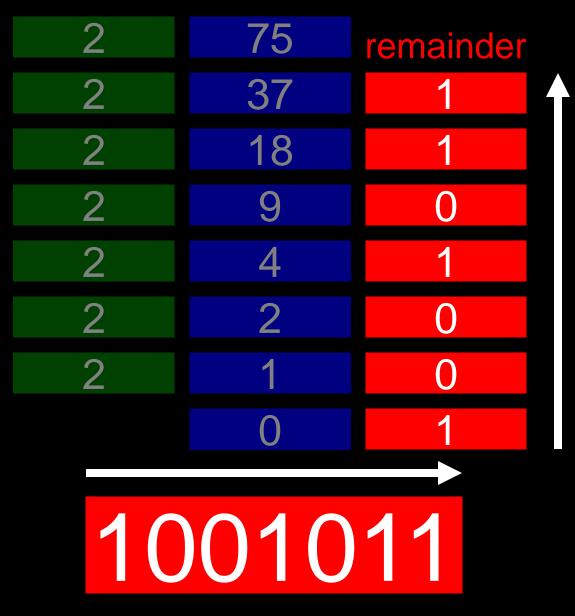
Byte = 8 bits



Decimal → Binary conversion



Convert 75 to Binary





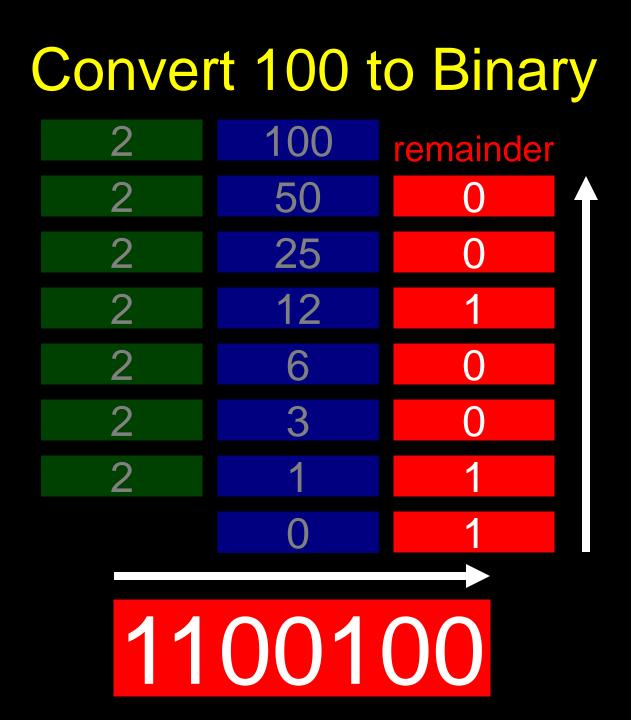
Check

$1001011 = 1x2^{0} + 1x2^{1} + 0x2^{2} + 1x2^{3} + 0x2^{4} + 0x2^{5} + 1x2^{6}$

= 1 + 2 + 0 + 8 + 0 + 0 + 64

=75







That finishes our first topic - introduction to binary numbers and their conversion to and from decimal numbers

Our next topic is ...



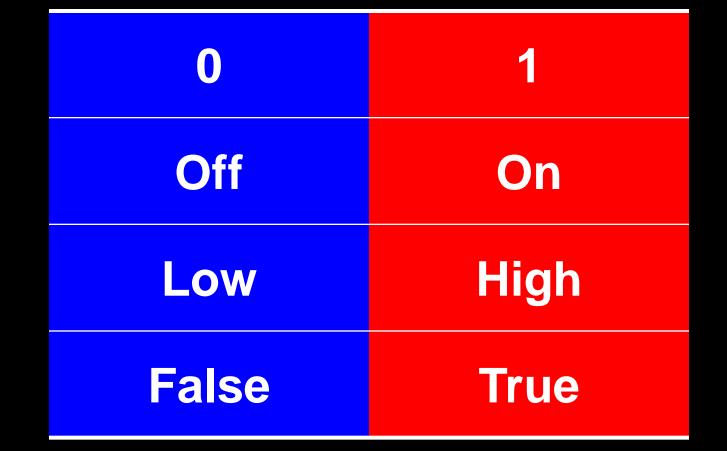
Boo ean Operations



Let X, Y, Z, be Boolean variables. Boolean variables can only have binary values i.e., they can have values which are either 0 or 1

For example, if we represent the state of a light switch with a Boolean variable x, we will assign a value of 0 to x when the switch is OFF, and 1 when it is ON

A few other names for the states of these Boolean variables





We define the following logic operations or functions among the Boolean variables

Name	Example	Symbolically
NOT	y = NOT(x)	X
AND	z = x AND y	x • y
OR	z = x OR y	<i>x</i> + <i>y</i>
XOR	$z = x \operatorname{XOR} y$	$x \oplus y$

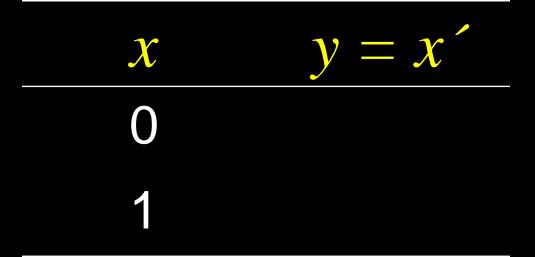
UniTubeCore

We'll define these operations with the help of truth tables

what is the truth table of a logic function A truth table defines the output of a logic function for all possible inputs

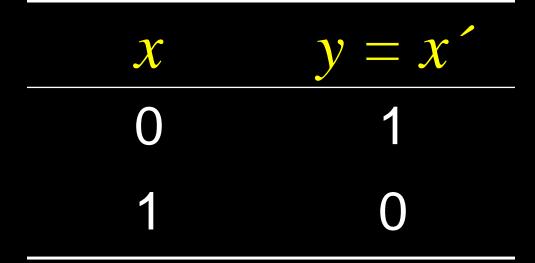


Truth Table for the NOT Operation (y true whenever x is false)





Truth Table for the NOT Operation





Truth Table for the AND Operation (z true when both x & y true)





Truth Table for the AND Operation



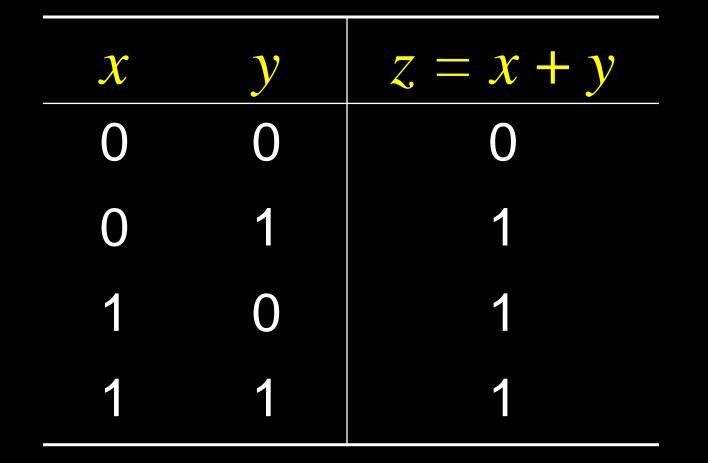


Truth Table for the OR Operation (z true when x or y or both true)





Truth Table for the OR Operation



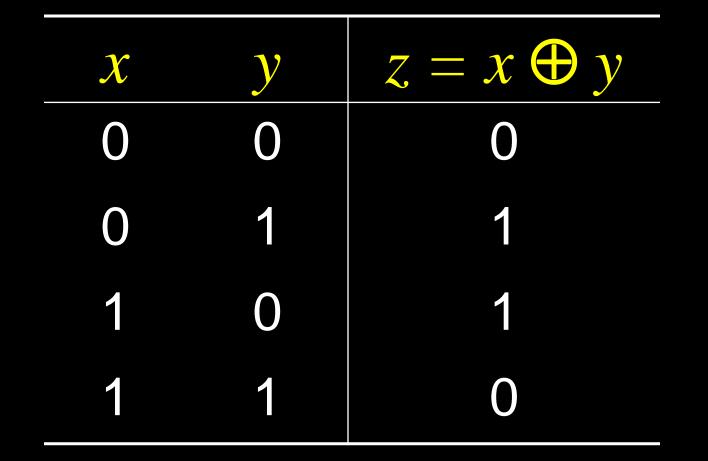


Truth Table for the XOR Operation (z true when x or y true, but not both)

${\mathcal X}$	У	$z = x \oplus y$
0	0	
0	1	
1	0	
1	1	



Truth Table for the XOR Operation





Those 4 were the fundamental logic operations. Here are examples of a few more complex situations

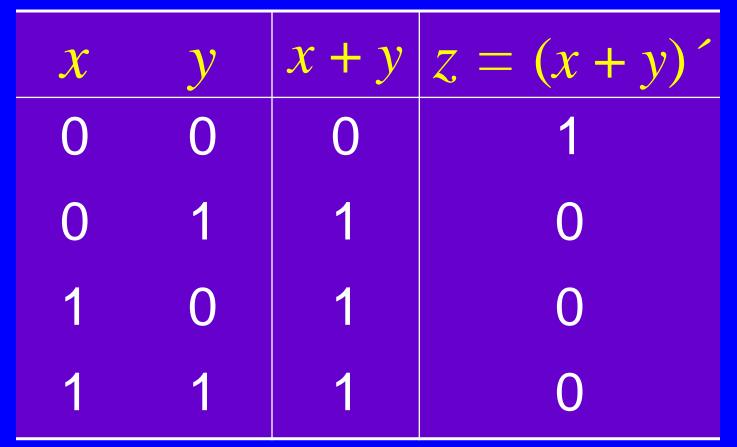
z = (x + y)'

 $z = y \cdot (x + y)$

$z = (y \cdot x) \bigoplus$ *w* **STRATEGY: Divide & Conquer**



$z = (x + y)^{\prime}$





$z = y \cdot (x + y)$

X	у	<i>x</i> + <i>y</i>	$z = y \cdot (x + y)$
0	0	0	0
0	1	1	1
1	0	1	0
1	1	1	1



$z = (y \cdot x) \oplus w$

X	У	W	$y \cdot x$	$z = (y \cdot x) \oplus$
				${\mathcal W}$
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

re

Number of rows in a truth table?



n = number of input variables



Assignment # 3

- A. Convert the following into binary numbers:i. The last three digits of your roll number
 - ii. 256
- B. x, y & z are Boolean variables. Determine the truth tables for the following combinations:

$$(x \cdot y) + y$$

ii. $(x \oplus y)' + w$

Consult the CS101 syllabus for the submission instructions & deadline



What have we learnt today?

- 1. About the binary number system, and how it differs from the decimal system
- 2. Positional notation for representing binary and decimal numbers
- 3. A process (or algorithm) which can be used to convert decimal numbers to binary numbers
- 4. Basic logic operations for Boolean variables, i.e. NOT, OR, AND, XOR, NOR, NAND, XNOR
- 5. Construction of truth tables (How many rows?)



Focus of the Next Lecture

Next lecture will be the 3rd on Web dev

The focus of the one after that, the 10th lecture, however, will be on software. During that lecture we will try:

- To understand the role of software in computing
- To become able to differentiate between system and application software

