Digital Computer Fundamentals An Introduction

Jonathan Goode

November 9, 2016

What Is A Digital Computer?

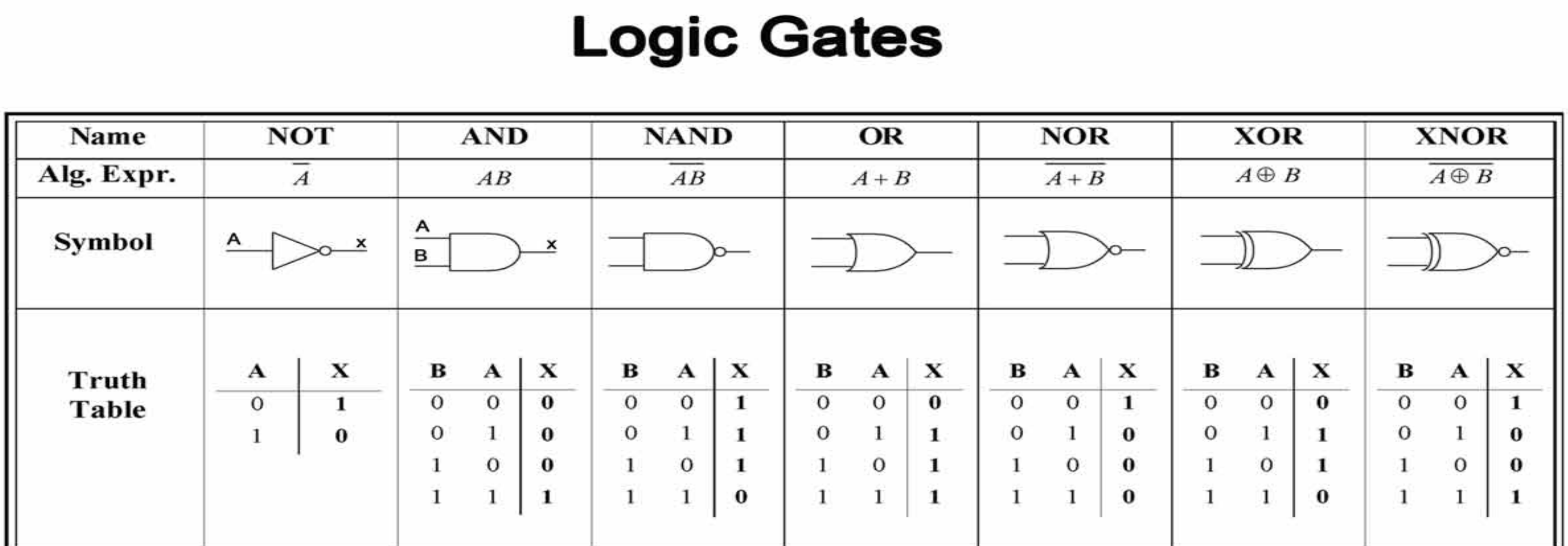
- It’s just an adding machine
  - Made out of Gates (Logic, not Robert or Bill)
  - Incredibly, Inconceivably, Wicked Fast
  - Programmable

What Are Gates?

- Logic Elements that perform functions
  - And
  - Or
  - Inversion or Negation
  - Often combined into one element, e.g. Nand

The OR Function

- If any input is True (High, or One), the output is True (High, or One)
- If all inputs are False (Low, or Zero), the output is False (Low, or Zero)



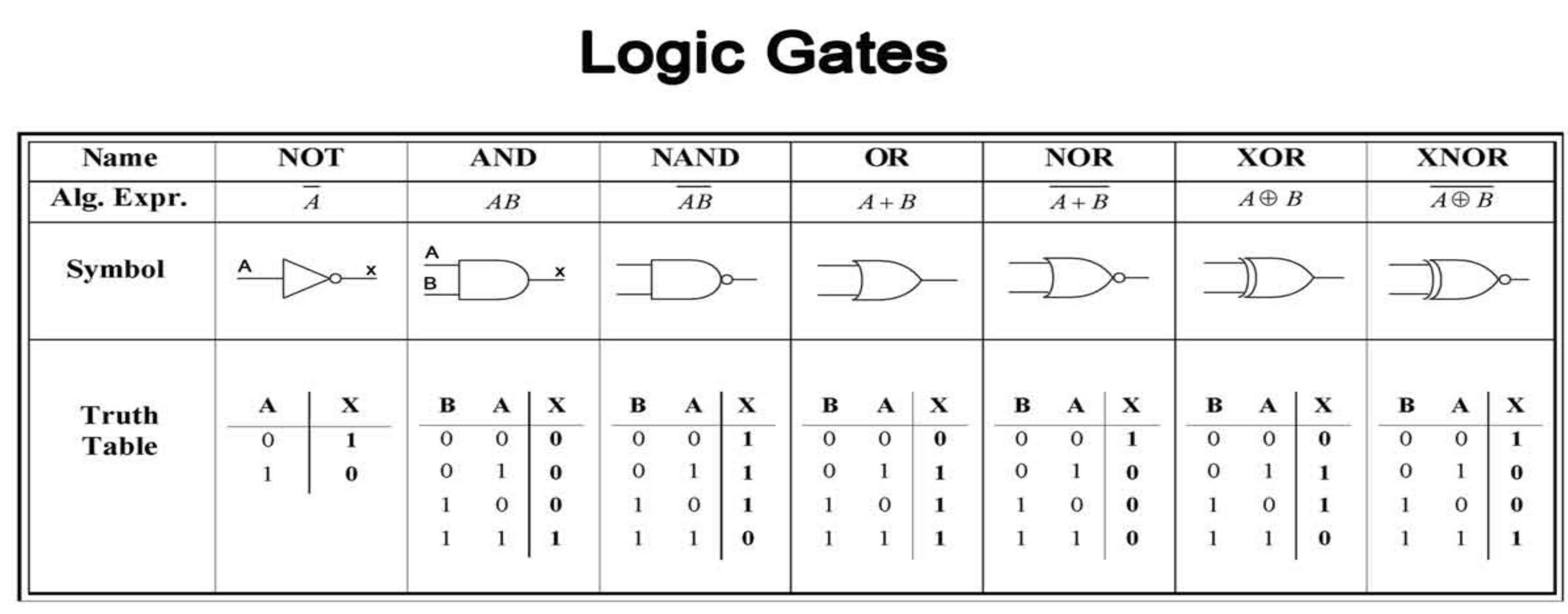
Logic Gates

Name	NOT	AND	NAND	OR	NOR	XOR	XNOR
Alg. Expr.	$\overline{A}$	AB	$\overline{AB}$	$A+B$	$\overline{A+B}$	$A \oplus B$	$\overline{A \oplus B}$
Symbol	A, x	A, B, x					
Truth Table	A X 0 1 1 0	B A X 0 0 0 0 1 0 1 0 0 1 1 1	B A X 0 0 1 0 1 1 1 0 1 1 1 0	B A X 0 0 0 0 1 1 1 0 1 1 1 1	B A X 0 0 1 0 1 0 1 0 0 1 1 0	B A X 0 0 0 0 1 1 1 0 1 1 1 0	B A X 0 0 1 0 1 0 1 0 0 1 1 1

Logic Gates Diagram From http://i.imgur.com/M59IOZQ.jpg

The AND Function

- If all inputs are True, the output is True
- If any input is False the output is False



Logic Gates

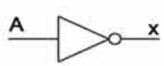
Name	NOT	AND	NAND	OR	NOR	XOR	XNOR
Alg. Expr.	$\overline{A}$	AB	$\overline{AB}$	$A+B$	$\overline{A+B}$	$A\oplus B$	$\overline{A\oplus B}$
Symbol	A x	A B x					
Truth Table	A X 0 1 1 0	B A X 0 0 0 0 1 0 1 0 0 1 1 1	B A X 0 0 1 0 1 1 1 0 1 1 1 0	B A X 0 0 0 0 1 1 1 0 1 1 1 1	B A X 0 0 1 0 1 0 1 0 0 1 1 0	B A X 0 0 0 0 1 1 1 0 1 1 1 0	B A X 0 0 1 0 1 0 1 0 0 1 1 1

Logic Gates Diagram From http://i.imgur.com/M59IOZQ.jpg

The NOT Function (Inverter)

- If the input is True, the output is False
- If the input is False, the output is True

Logic Gates

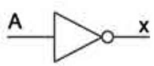
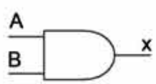
Name	NOT	AND	NAND	OR	NOR	XOR	XNOR
Alg. Expr.	$\overline{A}$	AB	$\overline{AB}$	$A+B$	$\overline{A+B}$	$A \oplus B$	$\overline{A \oplus B}$
Symbol							
Truth Table	A | X 0 | 1 1 | 0	B A | X 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1	B A | X 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0	B A | X 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1	B A | X 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0	B A | X 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0	B A | X 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1

Logic Gates Diagram From http://i.imgur.com/M59IOZQ.jpg

The NAND Function

- If all inputs are True, the output is False
- If any input is False, the output is True

Logic Gates

Name	NOT	AND	NAND	OR	NOR	XOR	XNOR
Alg. Expr.	$\overline{A}$	AB	$\overline{AB}$	$A+B$	$\overline{A+B}$	$A \oplus B$	$\overline{A \oplus B}$
Symbol							
Truth Table	A X 0 1 1 0	B A X 0 0 0 0 1 0 1 0 0 1 1 1	B A X 0 0 1 0 1 1 1 0 1 1 1 0	B A X 0 0 0 0 1 1 1 0 1 1 1 1	B A X 0 0 1 0 1 0 1 0 0 1 1 0	B A X 0 0 0 0 1 1 1 0 1 1 1 0	B A X 0 0 1 0 1 0 1 0 0 1 1 1

Logic Gates Diagram From http://i.imgur.com/M59IOZQ.jpg

Where's the Adder?
Stay Tuned

- Normal, Decimal Arithmetic vs Binary
- Each column has
  - Two operands, A and B
  - Possible Carry In, Cin
  - Possible Carry Out, Cout
- Formula for Binary Column

Decimal	
2	5
+3	+6
=5	=11
Binary	
0010	0101
+0011	+0110
=0101	=1011

A	0	1	0	1	0	1	0	1
B	0	0	1	1	0	0	1	1
Cin	0	0	0	0	1	1	1	1
Sum	0	1	1	0	1	0	0	1
Cout	0	0	0	1	0	1	1	1

A Little Minimization
Karnaugh Maps

- Main

A	0	1	0	1	0	1	0	1
B	0	0	1	1	0	0	1	1
Cin	0	0	0	0	1	1	1	1
Sum	0	1	1	0	1	0	0	1
Cout	0	0	0	1	0	1	1	1

- Sum

Cin|AB	00	01	11	10
0	0	1	0	1
1	1	0	1	0

- Cout

Cin|AB	00	01	11	10
0	0	0	1	0
1	0	1	1	1

Formula?

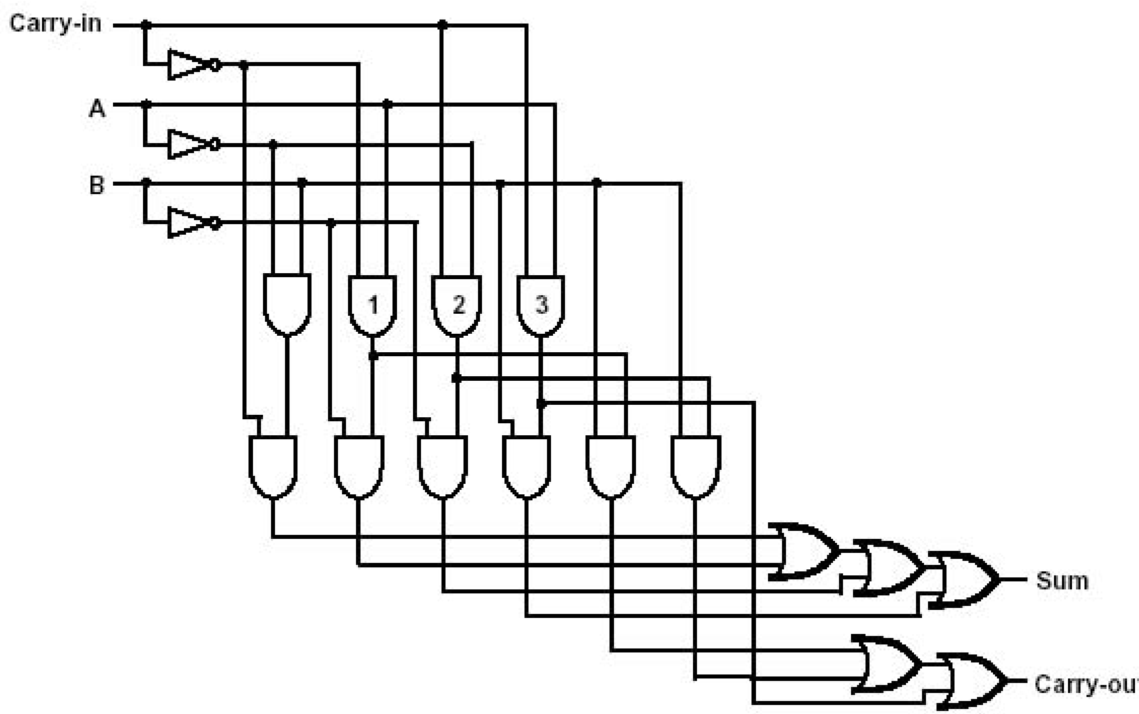
		Sum		
Cin|AB	00	01	11	10
0	0	1	0	1
1	1	0	1	0

		Cout		
Cin|AB	00	01	11	10
0	0	0	1	0
1	0	1	1	1

- Sum = (Not A and Not B and Cin) OR (Not A and B and Not Cin) OR (A and B and Cin) OR (A and not B and Cin)
- Cout = A and B OR ((A OR B) and Cin)
- Also, Cout = A and B OR not (not A and not B and Cin)
- Diagram shows Cout = (A and Cin) OR (A and B and not Cin) OR (not A and B and Cin), also correct and utilizes values already present for Sum

Adder from AND, OR and NOT gates

- Here is a circuit diagram of the adder (from http://cpuville.com/adder.htm)



Real Adders

- Real Adders consist of many adders connected in series
  - Sum size, e.g. 32 bits, determines how many stages
  - Inputs A and B are input in parallel, Carries ripple through in series (limiting the speed)

How Fast Is It?

- In 1964, the IBM 7094 had a 2 microsecond cycle time
  - Up to 500,000 operations per second! That's 30 million per minute.
  - This is still hard to fathom, 52 years later
  - Ordinary jet engines run at 10,000 rpm
  - The fastest jet microturbines run at 500,000 rpm, 1/60 of the speed

How Fast Are They Today?

- 3.3 Ghz is typical, 1650 times faster than before
- Multiple cores, pipelining and other techniques increase the speed even more
- Could present details on speedup techniques

Programming

- Computers step through a list of instructions
  - Fetch the instruction from memory, from an address specified in the Instruction Counter
  - Decode and perform specified operation, e.g.
    - CLA CLear and Add (Load) value from memory to Accumulator
    - ADD Add a value from memory to Accumulator
    - STO STOre the Accumulator into a memory location
    - Test and repeat if not finished

What Could Be Next

- Logic Design
  - Construction using only NAND Gates
  - Negative Numbers
  - Operations: Subtraction, Multiplication, Division
  - Floating Point
- Speed: Modern Computer Architecture
- Programming
  - Machine instruction details
  - Programming Languages
  - Software Development Issues
- What happens when you type on the keyboard – its operation may look like a mechanical typewriter but it's nothing like one