Datapath: registers, memories, Arithmetic logic units

Control: registers read/write, which ALU operation needs to be performed

Fetch: gets instruction, and translate that into the control unit

Professor cycle is fetch, then decode, then execute

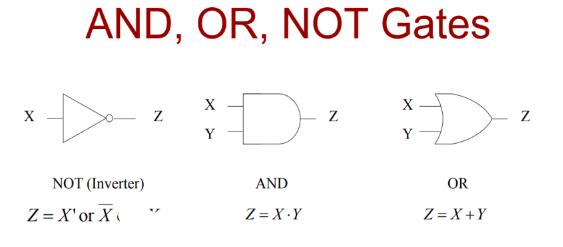
Digital Logic is based on binary values with 3 operations

AND - if all inputs are true, then output is true

OR - any inputs are true, then output is true

NOT- output is opposite of input

AND, OR, and NOT Gates

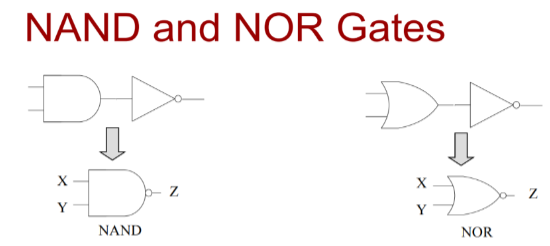


Transistors in logic

* The connection of transistors depends on the logical operation
  + In series circuits, AND; all transistors must be on to turn on the light
  + In parallel circuits, OR; any transistor can be on to turn on the light
* Can have more than just 2 inputs like 3 or 4 for example
* OR is anything that is one or more
* AND is if all are connected

NAND and NOR gates are basically the opposite of AND and OR Gates

* NAND is essentially a NOT and an AND in series; opposite of AND; Output is true if not all inputs are true
* NOR is essentially a NOT an an OR in series; opposite of OR; output is true if none of the inputs are true



| XNOR and XOR Gates: own unique types of gates   * XOR - True if an odd number of inputs are true/ true if the inputs are different * XNOR - True if an even number of inputs are true/ true if the inputs are the same |  |
| --- | --- |

Logic Functions

* A logic function maps input combinations to an output value (1 or 0)
* 3 possible representations of function
  + Equation
  + Schematic
  + Truth tables
* Easily convert between these representations
* Truth table is the only representation that is unique

Truth Table

* Is used to describe the behavior of a digital circuit
* Shows all possible combinations of inputs as a separate row in the table
* Shows what the output is for each input combination

Truth tables for equations/circuits

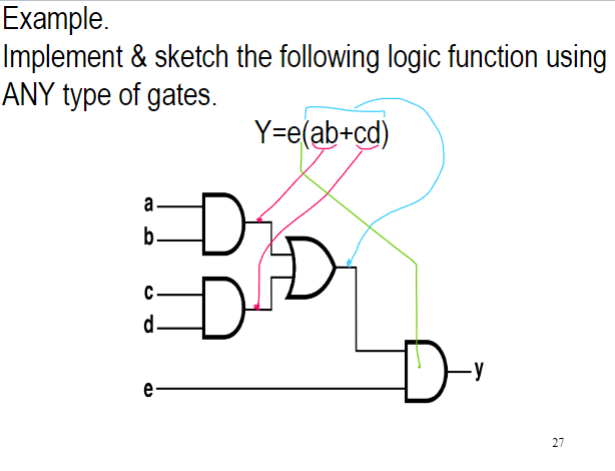
* Given a circuit with n-inputs, 2^n possible combinations exist aka 2^n rows in a Truth Table

Boolean Algebra

* A set of theorems that help us manipulate logical expressions/equations
  + Axioms: Basis / assumptions used (facts)
  + Theorems: manipulations that we can use

Boolean algebra Terminology

* A literal is an instance of a single variable or its complement
* Product term is a single variable or a literal logic product (ANDing) of two or more literals
* A sum term is a single literal or a logical sum ORing of two or more literals
* Sum of Products Form: An SOP expression is a logical sum (OR) of product terms
* Product of sums: A POS expression is a logical product (AND) of sum terms)



Circuit delay and critical path

* Wires have delay: time it takes for data to travel on them
* Path delay - time for input to affect output
* Critical path - path with the longest path delay
* Circuit delay - delay of critical path

**Minterms and max terms**

* A minterm can be generated for every combination of inputs
* A minterm is the anding of variables that will evaluate to 1 for only that combination
* A minterm checks a specific input combination and outputs one when found

Canonical sums are the sum of all of the minterms or the OR of all of the minterms where F = 1

* A maxterm can be generated for every combination of inputs
* A maxterm is the oring of variables that will evaluate to 10 for only that combination
* A minterm checks a specific input combination and outputs 0 when found

We want to optimize, speed, area, and power when designing a circuit, usually only 2 out of the 3 are possible

Venn diagrams are great for representing the logic of 1, 2 or 3 variable functions

DeMorgan’s theorem for logic simplification circuit analysis