Essentials of Computer Architecture 2nd edition Comer Solutions Manual

SKU: 9781138626591-SOLUTIONS

Module II

Fundamentals Of Digital Logic

Copyright © 2016 by Douglas C. Comer. All rights reserved.

Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Our Goals

- Understand the basics
  - Concepts
  - How computers work at the lowest level
- Avoid whenever possible
  - Device physics
  - Engineering design rules
  - Implementation details

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Electrical Terminology

- Voltage
  - Quantifiable property of electricity
  - Measure of potential force
  - Unit of measure: *volt*
- Current
  - Quantifiable property of electricity
  - Measure of electron flow along a path
  - Unit of measure: *ampere (amp)*

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Analogy

- Voltage is analogous to water pressure
- Current is analogous to flowing water
- Can have
  - High pressure with little flow
  - Large flow with little pressure

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Measuring Voltage

- Device used is called *voltmeter*
- Note: can only be measured as *difference* between two points
- We will
  - Assume one point represents zero volts (known as *ground*)
  - Express voltage of second point with respect to ground

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

In Practice

- In lab, chips will operate on five volts
- Two wires connect each chip to *power supply*
  - Ground (zero volts)
  - Power (five volts)
- Every chip needs power and ground connections
- Notes
  - Logic diagrams do not show power and ground
  - Raspberry Pi operates on 3.3 volts, so conversion is required to connect the Pi to other chips

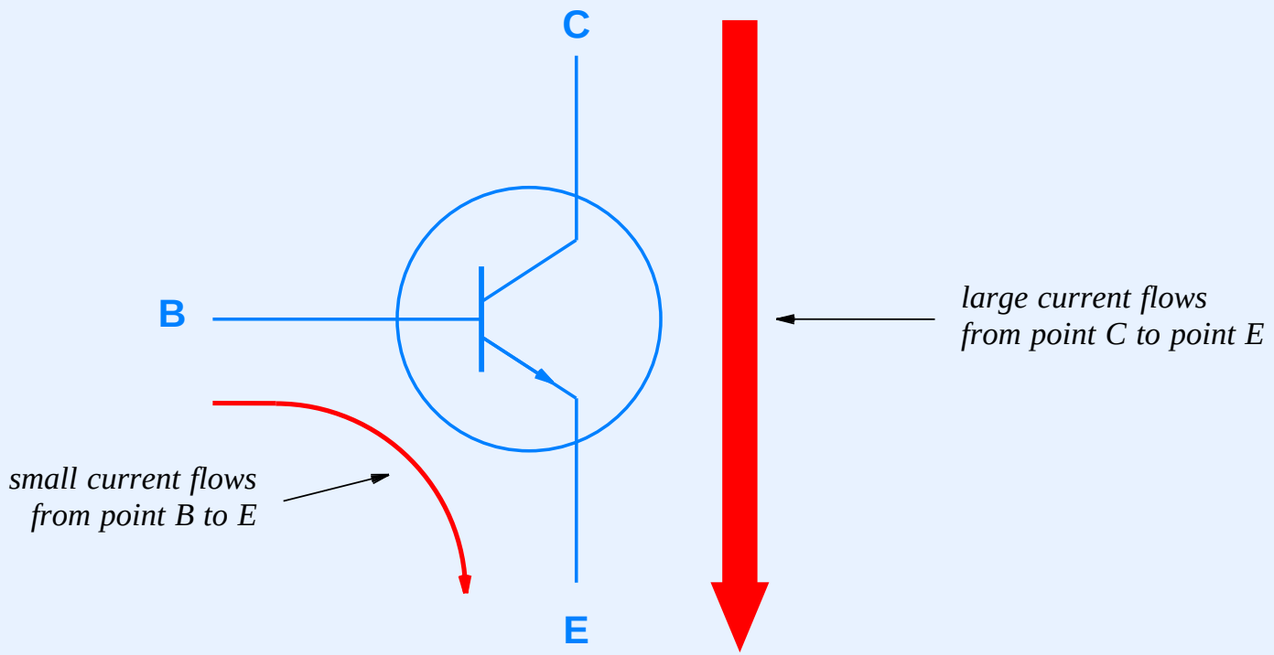
Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Transistor

- Basic building block of electronic circuits
- Operates on electrical current
- Traditional transistor
  - Has three external connections
    - Emitter
    - Base (control)
    - Collector
  - Acts like an *amplifier* — a small current between base and emitter controls large current between collector and emitter

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Illustration Of A Traditional Transistor



- *Amplification* means the large output current varies exactly like the small input current

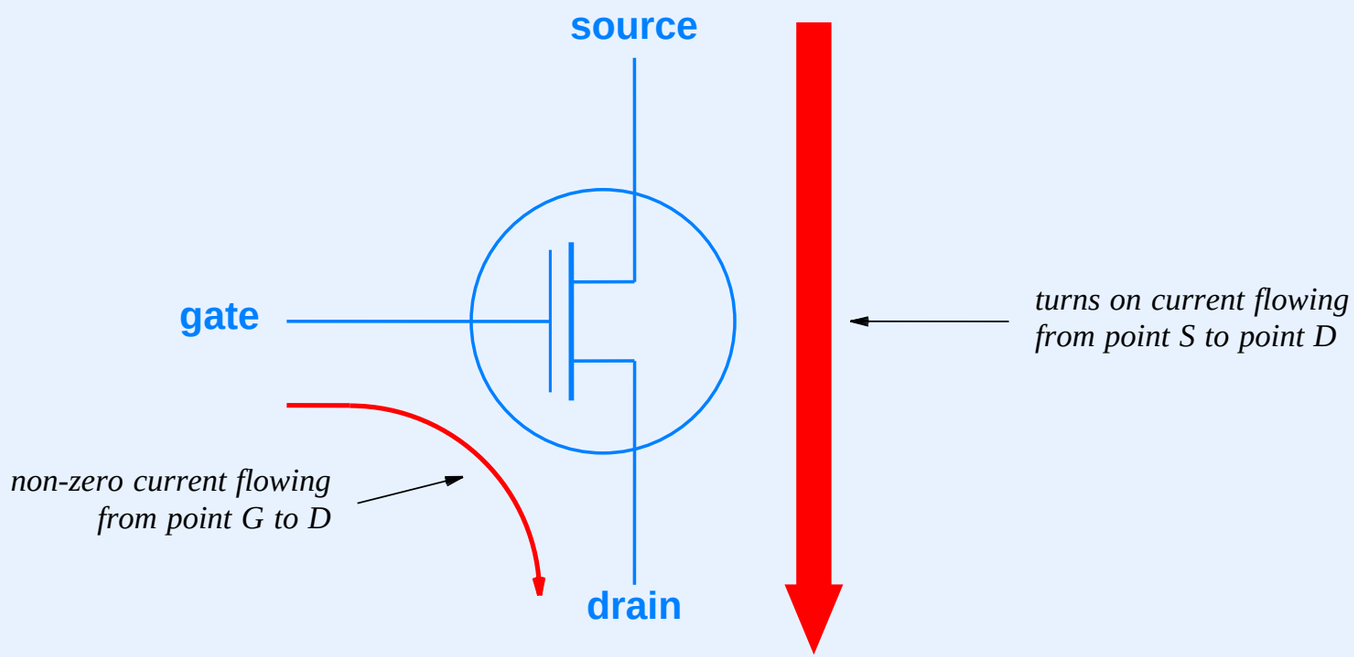
Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Field Effect Transistor

- Called a *Metal Oxide Semiconductor FET (MOSFET)* when used on a CMOS chip
- Three external connections
  - Source
  - Gate
  - Drain
- Designed to act as a switch (on or off)
  - When the input reaches a threshold (i.e., becomes logic 1), the transistor turns on and passes full current
  - When the input falls below a threshold (i.e., becomes logic 0), the transistor turns off and passes no current

Copyright © 2016 by Douglas Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

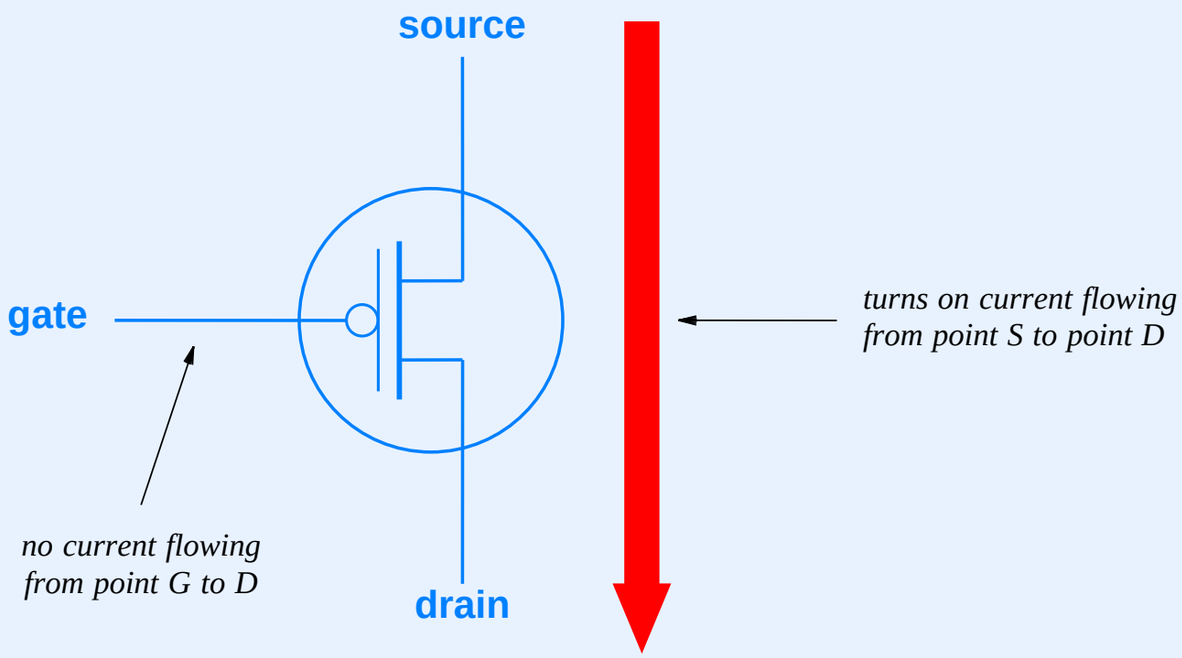
Illustration Of A Field Effect Transistor (Used For Switching)



- Input arrives at the gate
- Logic zero (zero volts) means the transistor is off; logic 1 (positive voltage) turns the transistor on

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Alternative Field Effect Transistor (Also Used For Switching)



- Circle on the gate indicates an inversion
- Logic 0 (zero volts) turns the transistor on, and logic 1 (positive voltage) turns the transistor off

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Boolean Logic

- Mathematical basis for digital circuits
- Three basic functions: *and*, *or*, and *not*

A	B	A and B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A or B
0	0	0
0	1	1
1	0	1
1	1	1

A	not A
0	1
1	0

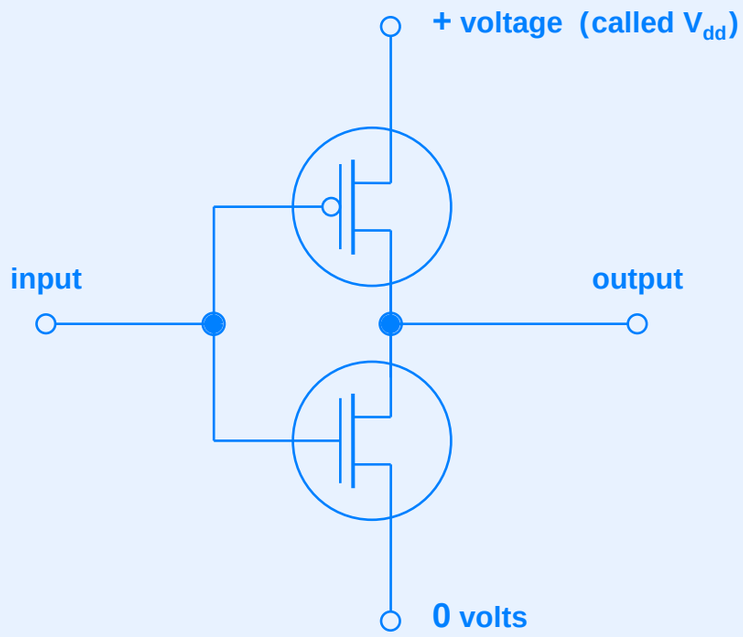
Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Digital Logic

- Can implement Boolean functions with transistors
- Five volts represents Boolean *1* (*true*)
- Zero volts represents Boolean *0* (*false*)

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Transistors Implementing Boolean Not



- When input is zero volts, output is connected to + voltage
- When input is five volts, output is connected to 0 volts
- Hardware engineers use V_{dd} to denote positive voltage

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Logic Gate

- Hardware component
- Consists of integrated circuit
- Implements an individual Boolean function
- To reduce complexity, hardware uses inverse of Boolean functions
  - Nand gate implements *not and*
  - Nor gate implements *not or*
  - Inverter implements *not*

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Truth Tables For Nand, Nor, and Xor Gates

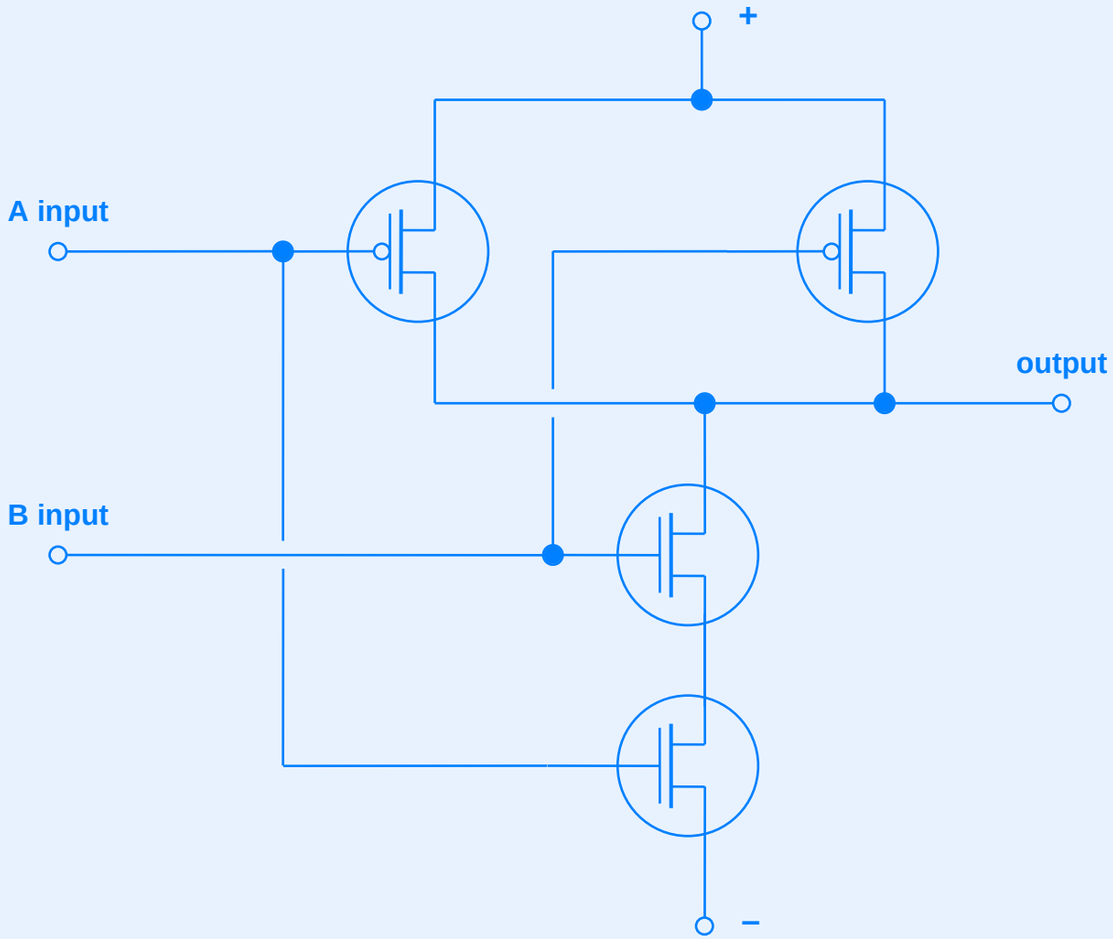
A	B	A nand B
0	0	1
0	1	1
1	0	1
1	1	0

A	B	A nor B
0	0	1
0	1	0
1	0	0
1	1	0

A	B	A xor B
0	0	0
0	1	1
1	0	1
1	1	0

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Example Of Internal Gate Structure (Nand Gate)

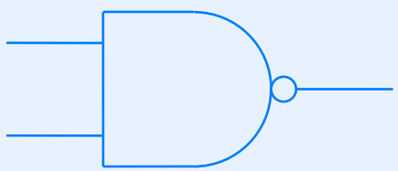


- Solid dot indicates electrical connection

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Symbols Used In Schematic Diagrams

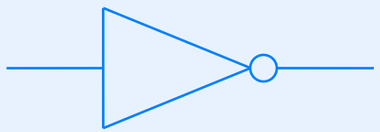
- Basic gates



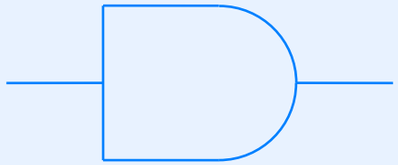
nand gate



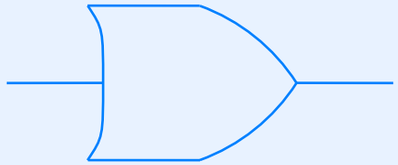
nor gate



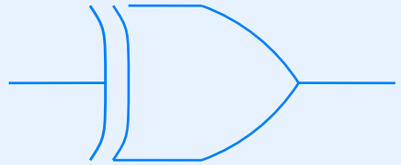
inverter



and gate



or gate



xor gate

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

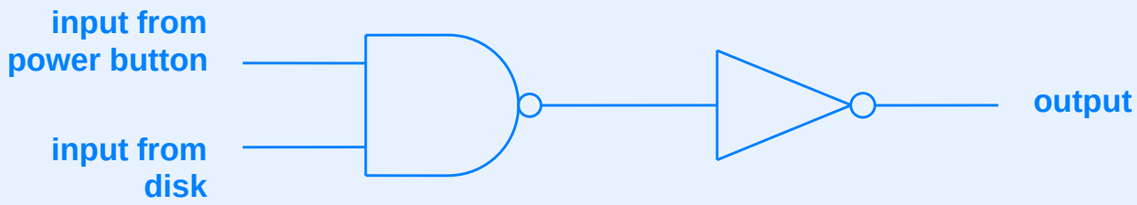
Technology For Logic Gates

- Most popular technology known as *Transistor-Transistor Logic* (*TTL*)
- Allows direct interconnection (a wire can connect output from one gate to input of another)
- Single output can connect to multiple inputs
  - Called *fanout*
  - Limited to a small number

Copyright © 2016 by Douglas Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

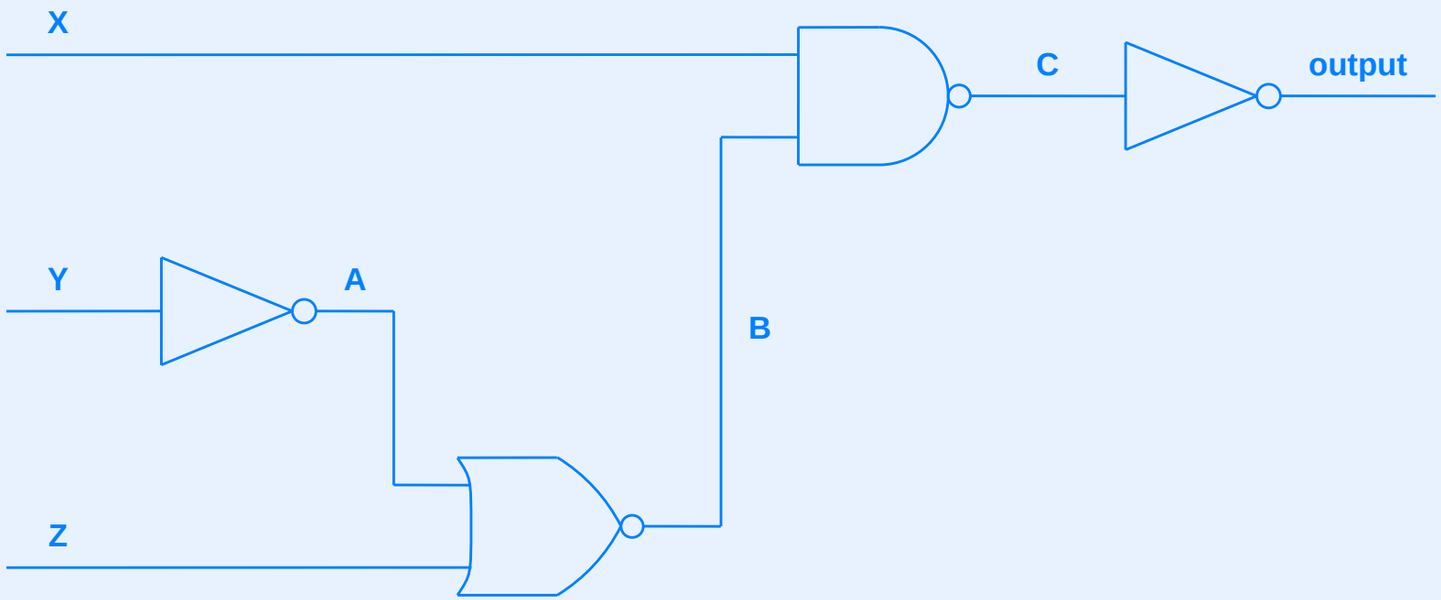
Example Interconnection Of TTL Gates

- Suppose we need a signal to indicate that the power button is depressed and the disk is ready
- Two logic gates are needed to form logical *and*
  - Output from nand gate connected to input of inverter



Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Consider The Following Circuit



- Question: what does the circuit implement?

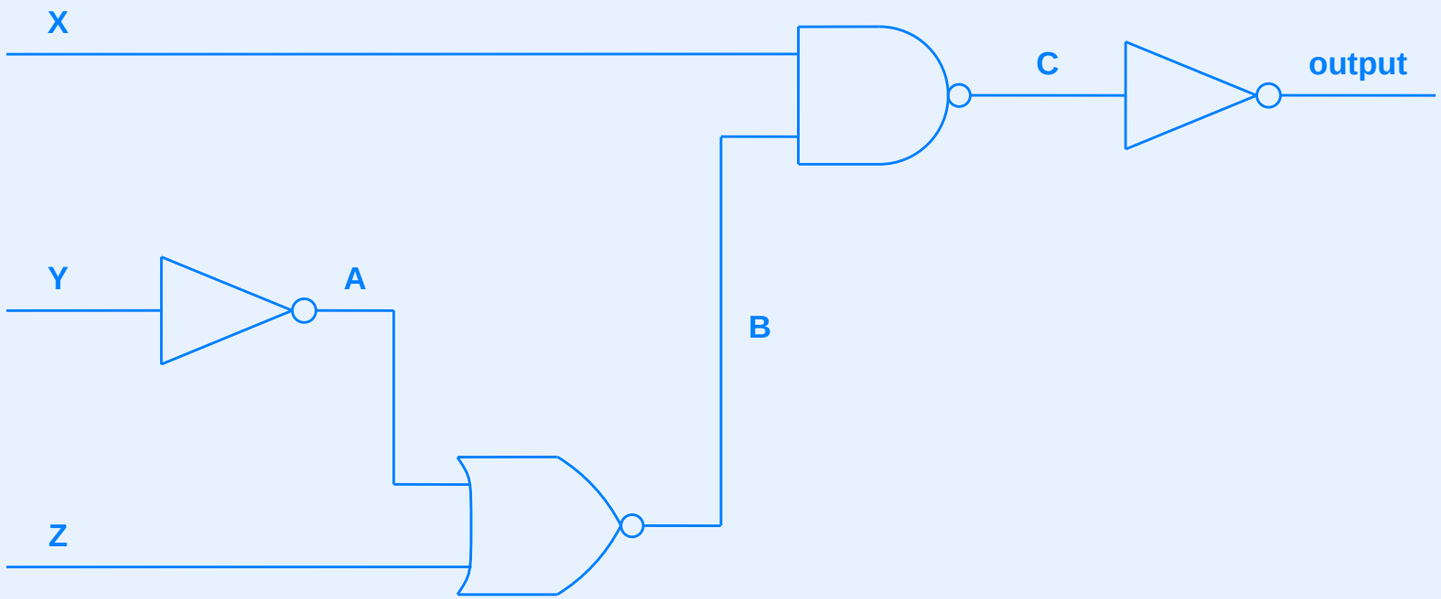
Copyright © 2016 by Douglas C. Comer. All rights reserved
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Two Ways To Describe A Circuit

- Boolean expression
  - Often used when designing circuit
  - Can be transformed to equivalent version that requires fewer gates
- Truth table
  - Enumerates inputs and outputs
  - Often used when debugging a circuit

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

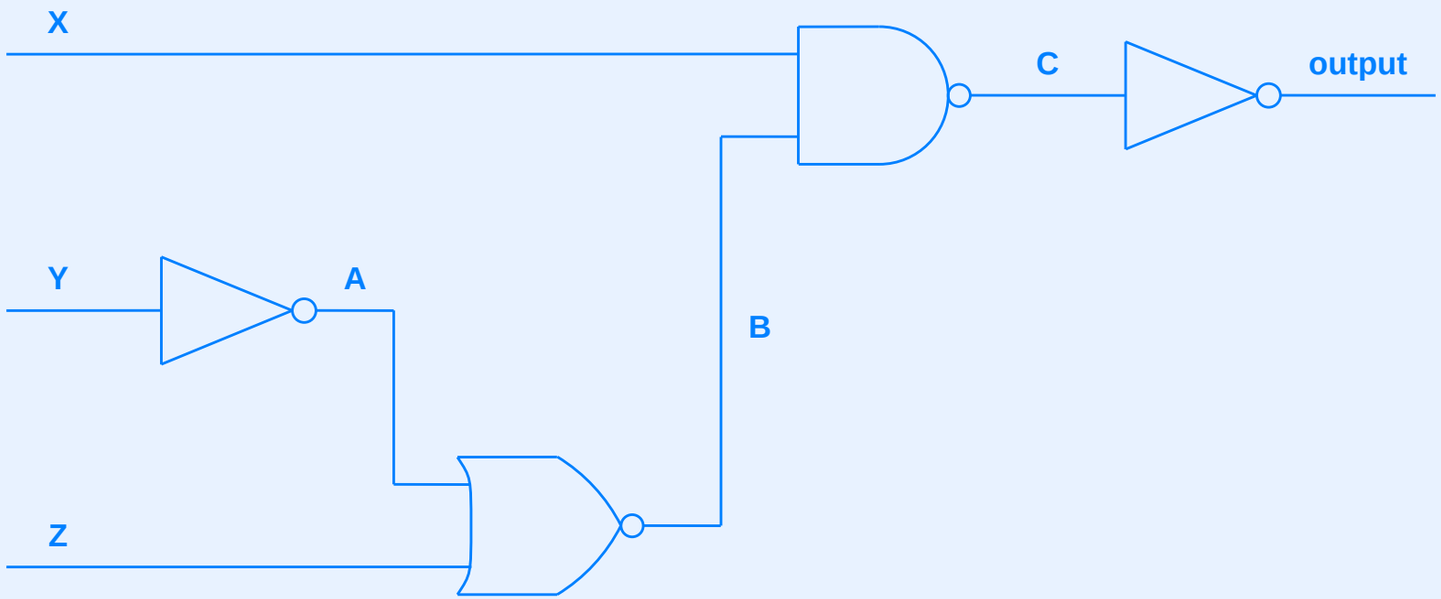
Describing A Circuit With Boolean Algebra



- Value at point *A* is: *not Y*
- Value at point *B* is: *Z nor (not Y)*

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Describing A Circuit With Boolean Algebra



- Value at point *C* is: *(X nand ((Z nor (not Y))*
- Value at output is: *X and (Z nor (not Y))*

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Simplifying Boolean Expressions

- Rules are similar to conventional algebra
  - Associative
  - Reflexive
  - Distributive
- See Appendix 2 in the text for details

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Describing A Circuit With A Truth Table

X	Y	Z	A	B	C	output
0	0	0	1	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	1	0
0	1	1	0	0	1	0
1	0	0	1	0	1	0
1	0	1	1	0	1	0
1	1	0	0	1	0	1
1	1	1	0	0	1	0

- Table lists all possible inputs and output for each
- Can also state values for intermediate points

Copyright © 2016 by Douglas C. Comer. All rights reserved
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Nand / Nor Vs. And / Or

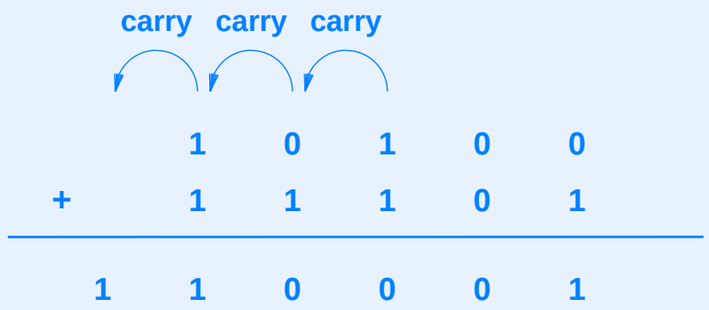
- Mathematically, *nand / nor / not* is equivalent to *and / or / not*
- Practically
  - It is possible to construct *and* and *or* gates
  - Sometimes, humans find *and* and *or* operations easier to understand
- Example circuit or truth table output can be described by Boolean expression:

X and Y and (not Z))

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Binary Addition

- How does a computer perform addition?
- Analogous to the method used in elementary school
- Each digit is a single bit

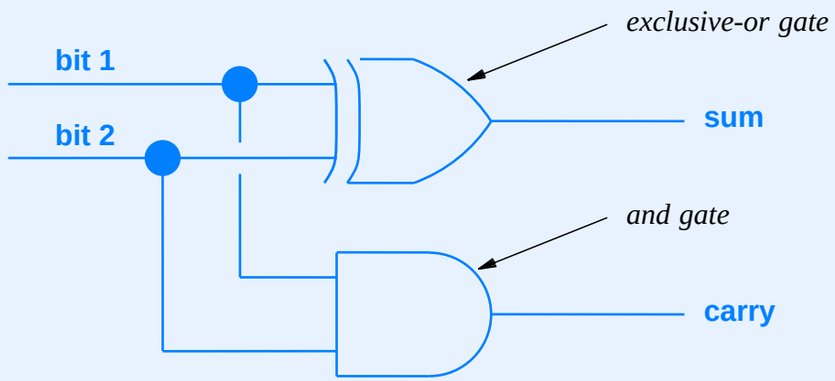


- Note: first bit never has a carry input

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Half-Adder Circuit

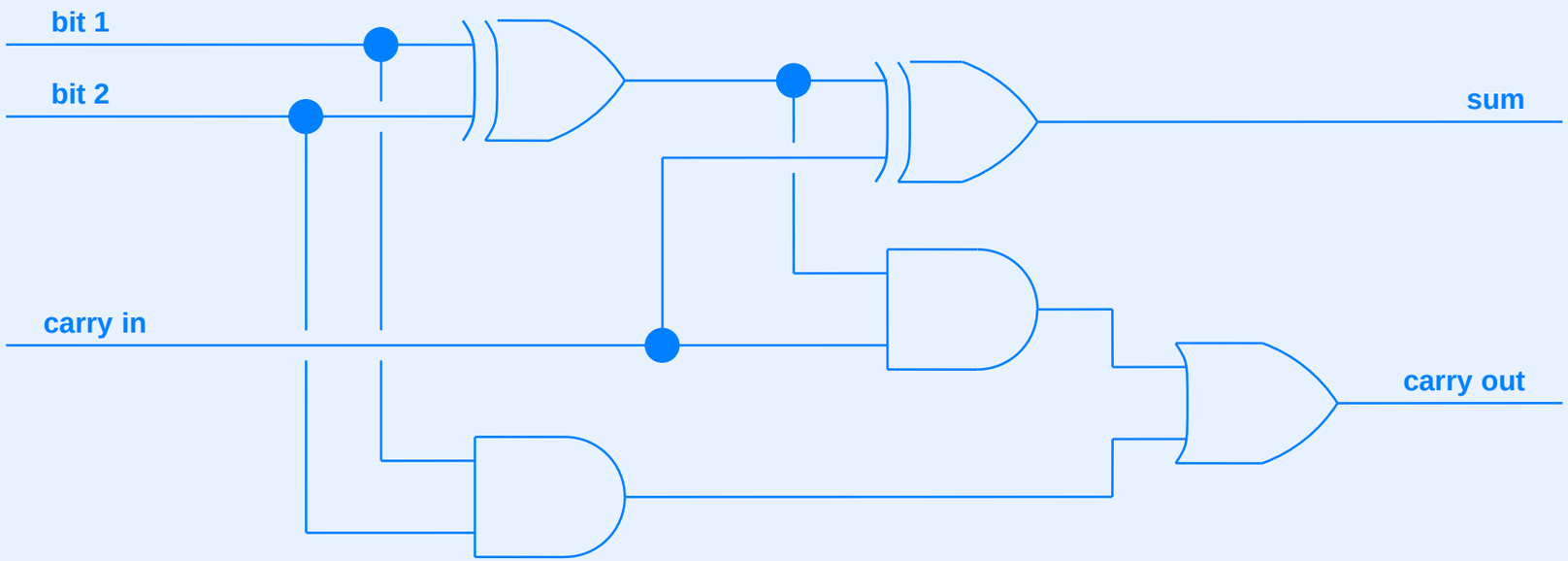
- Adds two input bits
- Produces two output bits
  - Sum
  - Carry
- We will use *exclusive or* gate plus *and* gate



Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Full-Adder Circuit

- Input is two bits plus a carry
- Produces two output bits
  - Sum
  - Carry



Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

In Practice

- A single gate only has a few connections
- A chip has many pins for external connections
- Result: package multiple gates on each chip
- We will see examples shortly

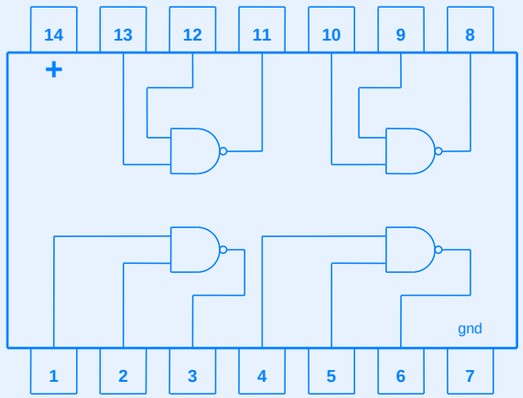
Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

An Example Logic Gate Technology

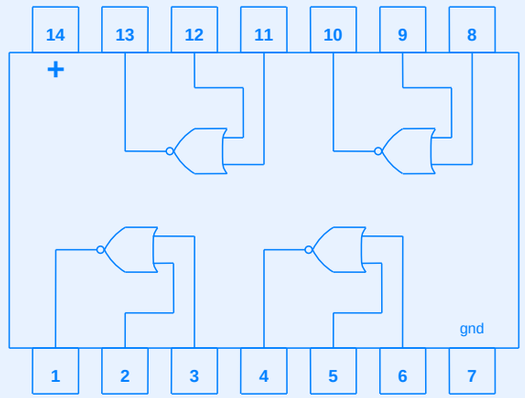
- 7400 family of chips
- Package is about one-half inch long
- Implement TTL logic
- Powered by five volts
- Each chip contains multiple gates

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

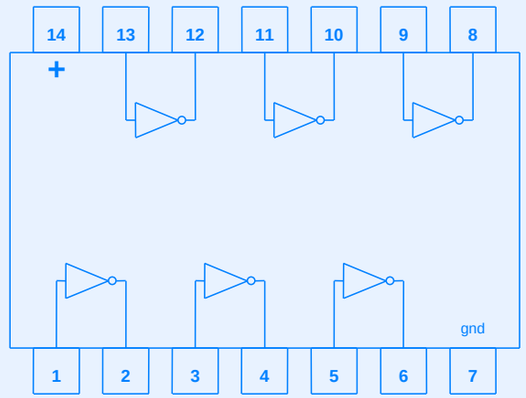
Example Gates On 7400-Series Chips



7400
(Quad 2-input NAND)



7402
(Quad 2-input NOR)



7404
(Hex Inverter)

- Pins 7 and 14 connect to ground and power
- Power and ground *must* be connected for the chip to operate

Copyright © 2016 by Douglas Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Logic Gates And Computers

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Logic Gates And Computers

- Question: how can computers be constructed from simple logic gates?

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Logic Gates And Computers

- Question: how can computers be constructed from simple logic gates?
- Answer: they cannot

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Logic Gates And Computers

- Question: how can computers be constructed from simple logic gates?
- Answer: they cannot
- Logic gates only provide a Boolean combination of inputs (known as *combinatorial circuits*)

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Logic Gates And Computers

- Question: how can computers be constructed from simple logic gates?
- Answer: they cannot
- Logic gates only provide a Boolean combination of inputs (known as *combinatorial circuits*)
- Additional functionality is needed
  - Circuits that maintain state
  - Circuits that operate on a clock

Copyright © 2016 by Douglas Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

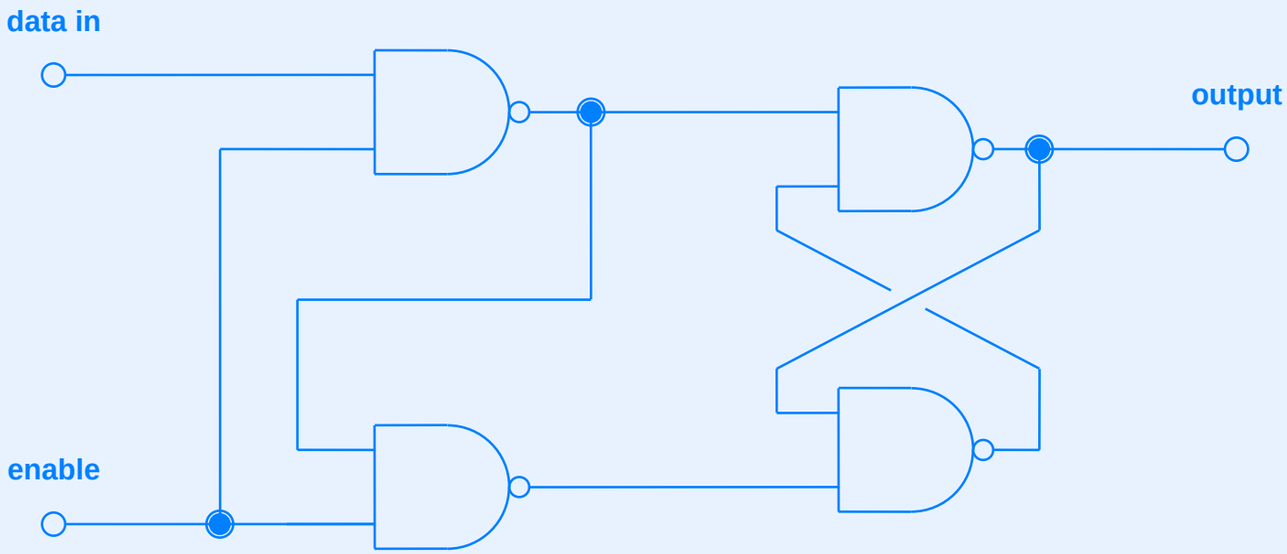
Circuits That Maintain State

- More sophisticated than combinatorial circuits
- Output depends on history of previous input as well as values on input lines

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Basic Circuit That Maintains State

- Known as *latch*
- Has two inputs: *data* and *enable*
- When enable is 1, output is same as data
- When enable goes to 0, output stays locked at current value



Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Propagation Delay

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

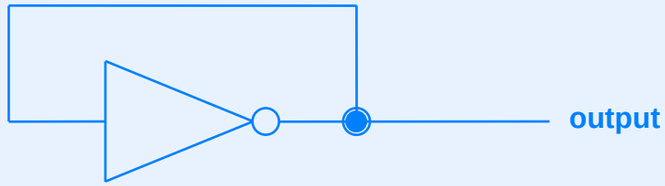
Propagation Delay

- Key in understanding a latch

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Propagation Delay

- Key in understanding a latch
- Consider the circuit

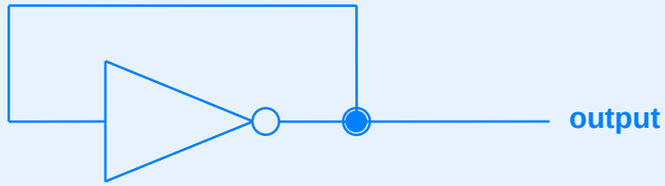


- What does it do?

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Propagation Delay

- Key in understanding a latch
- Consider the circuit

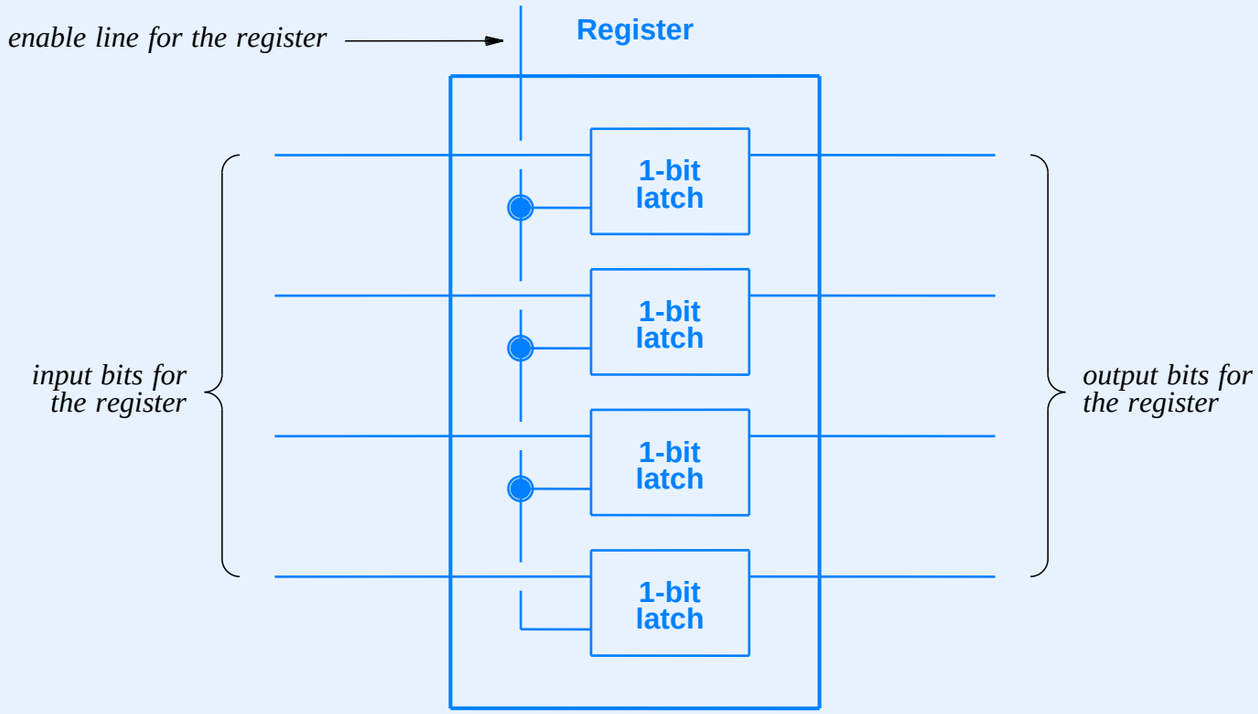


- What does it do?
- Mathematically, the circuit is meaningless because an inverter produces the complement of its input, but in this case the output is fed back into the input
- Practically, a *propagation delay* means the output stays the same for a short time, and then changes
- Result: output varies over time, 0 for time t, 1 for time t, 0 for time t, and so on, where t is the propagation delay

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Register

- Basic building block for a computer
- Acts like a miniature N-bit memory
- Can be built out of latches



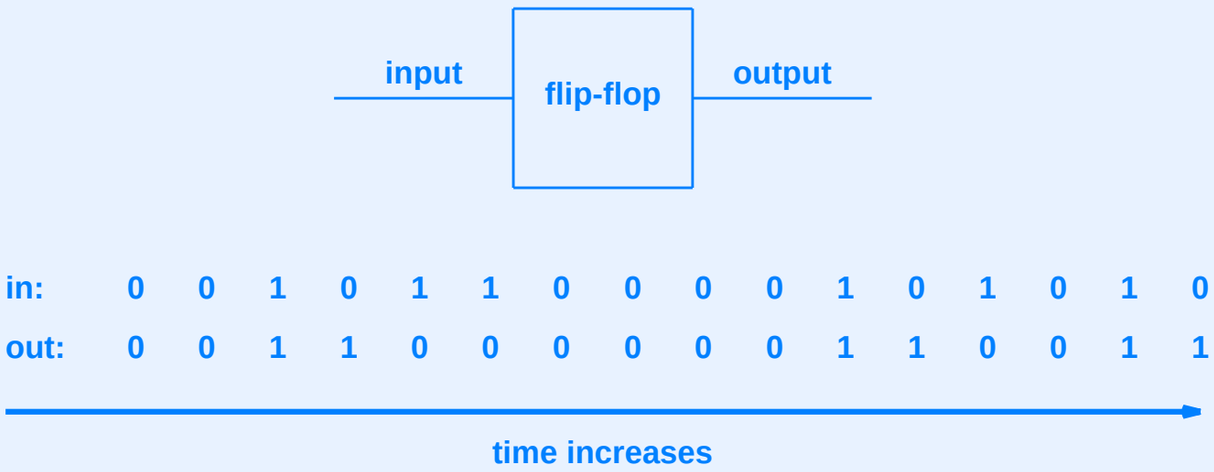
Copyright © 2016 by Douglas Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

A More Complex Circuit That Maintains State

- Basic *flip-flop*
- Can be constructed from a pair of latches
- Analogous to push-button power switch (i.e., push-on push-off)
- Each new 1 received as input causes output to reverse
  - First input pulse causes flip-flop to turn on
  - Second input pulse causes flip-flop to turn off

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

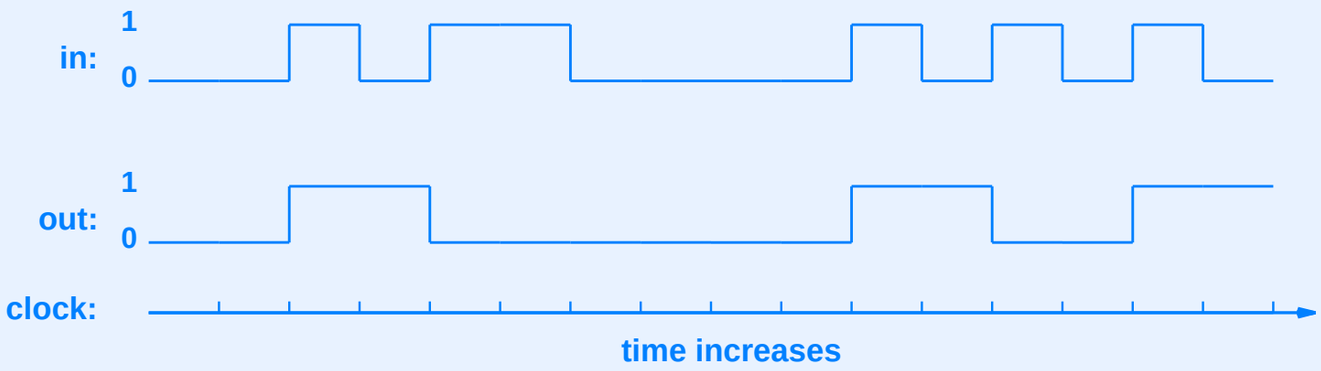
Output Of A Flip-Flop



- Note: output only changes when input makes a transition from zero to one (i.e., *rises*)

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Flip-Flop Action Plotted As Transition Diagram



- All changes synchronized with clock (described later)
- Output changes on *rising edge* of input
- Also called *leading edge*

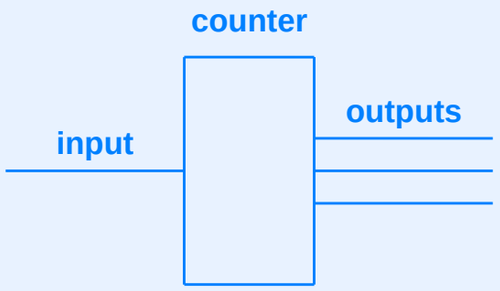
Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Binary Counter

- Counts input pulses
- Output is binary value
- Includes *reset line* to restart count at zero
- Example: 4-bit counter available as single integrated circuit

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Illustration Of Counter



(a)

time increases

input	outputs	decimal
0	0 0 0	0
0	0 0 0	0
1	0 0 1	1
0	0 0 1	1
1	0 1 0	2
1	0 1 0	2
0	0 1 0	2
1	0 1 1	3
0	0 1 1	3
1	1 0 0	4
0	1 0 0	4
1	1 0 1	5
	. . .	

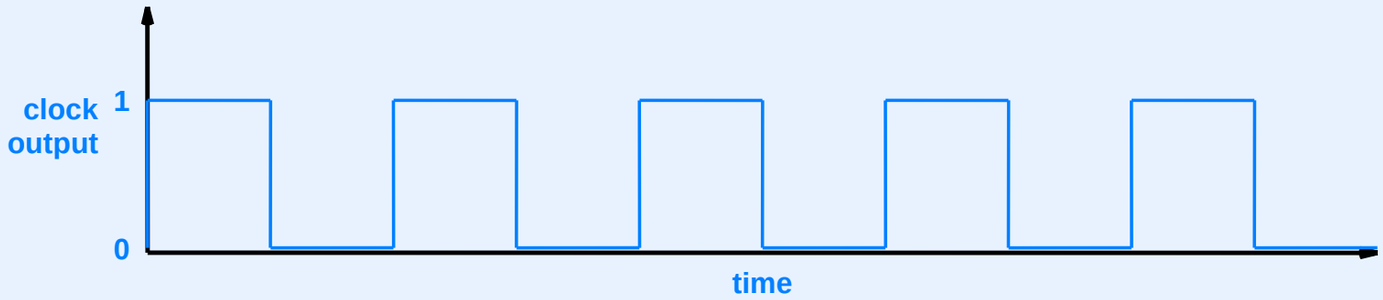
(b)

- Part (a) shows the schematic of a counter chip
- Part (b) shows the output as the input changes

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Clock

- Permits active circuits
- Electronic circuit that pulses regularly
- Measured in cycles per second (Hz)
- Output of clock is *square wave* (sequence of 1 0 1 0 1 ...)



Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

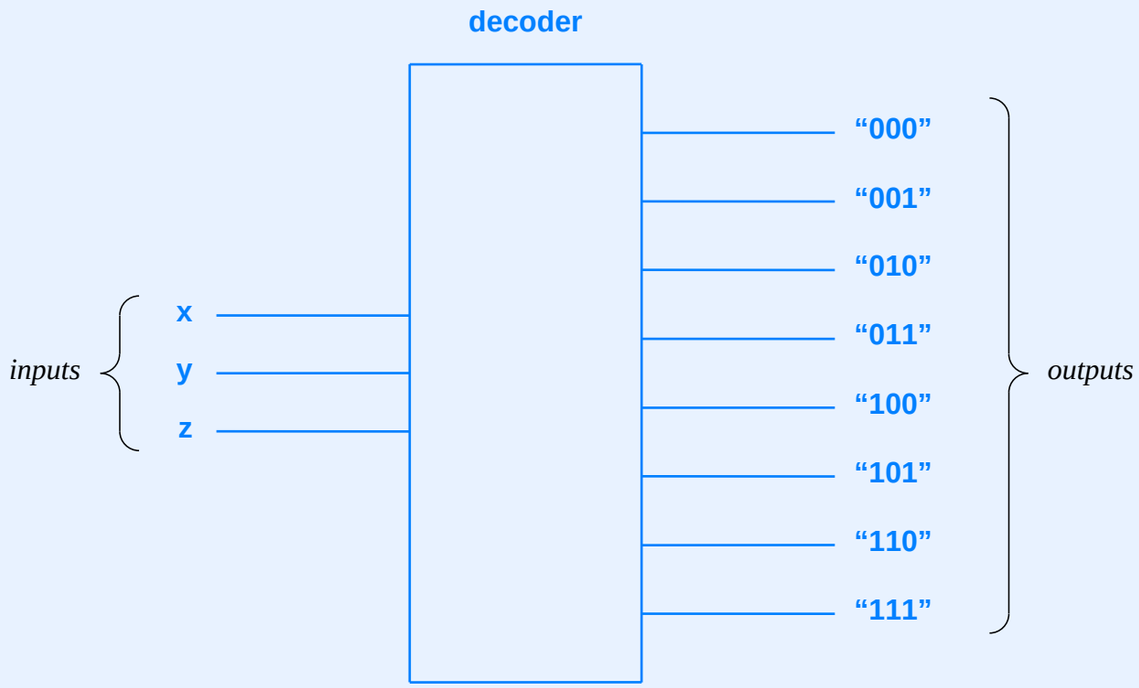
Decoder/Demultiplexor

- Takes binary number as input
- Uses input to select one output
- Technical distinction
  - *Decoder* simply selects one of its outputs
  - *Demultiplexor* feeds a special input to the selected output
- In practice: engineers often use the term “demux” for either, and blur the distinction

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Illustration Of Decoder

- Binary value on input lines determines which output is active



- Technical detail: on some decoder chips, an active output is logic 0 and all others are logic 1

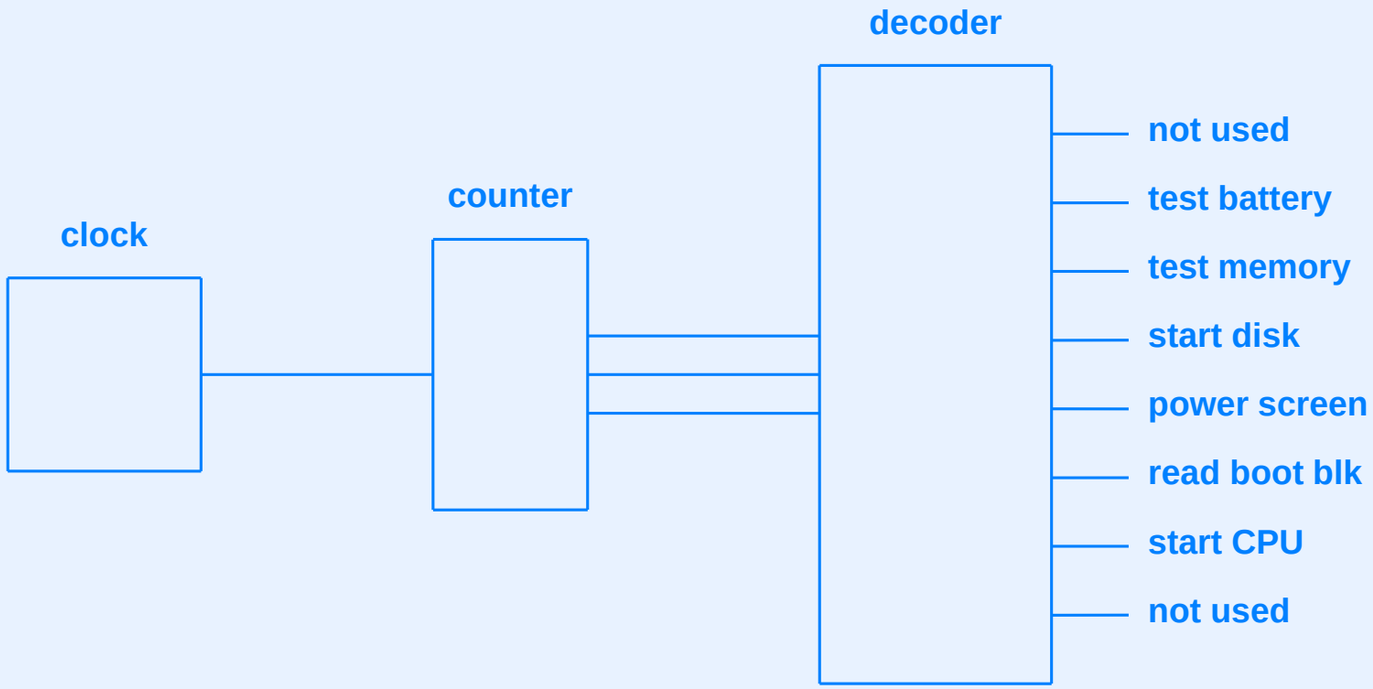
Copyright © 2016 by Douglas Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Example: Execute A Sequence Of Steps

- Imagine the power-on sequence for an embedded system
  - Test the battery
  - Power on and test the memory
  - Start the disk
  - Power up the display
  - Read boot sector from disk into memory
  - Start the CPU
- Separate hardware module performs each task
- Need to activate the modules in sequence

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Circuit To Execute A Sequence



- Technique: count clock pulses and use decoder to select an output for each possible counter output
- Note: counter will wrap around to zero, so this is an infinite loop

Copyright © 2016 by Douglas E. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

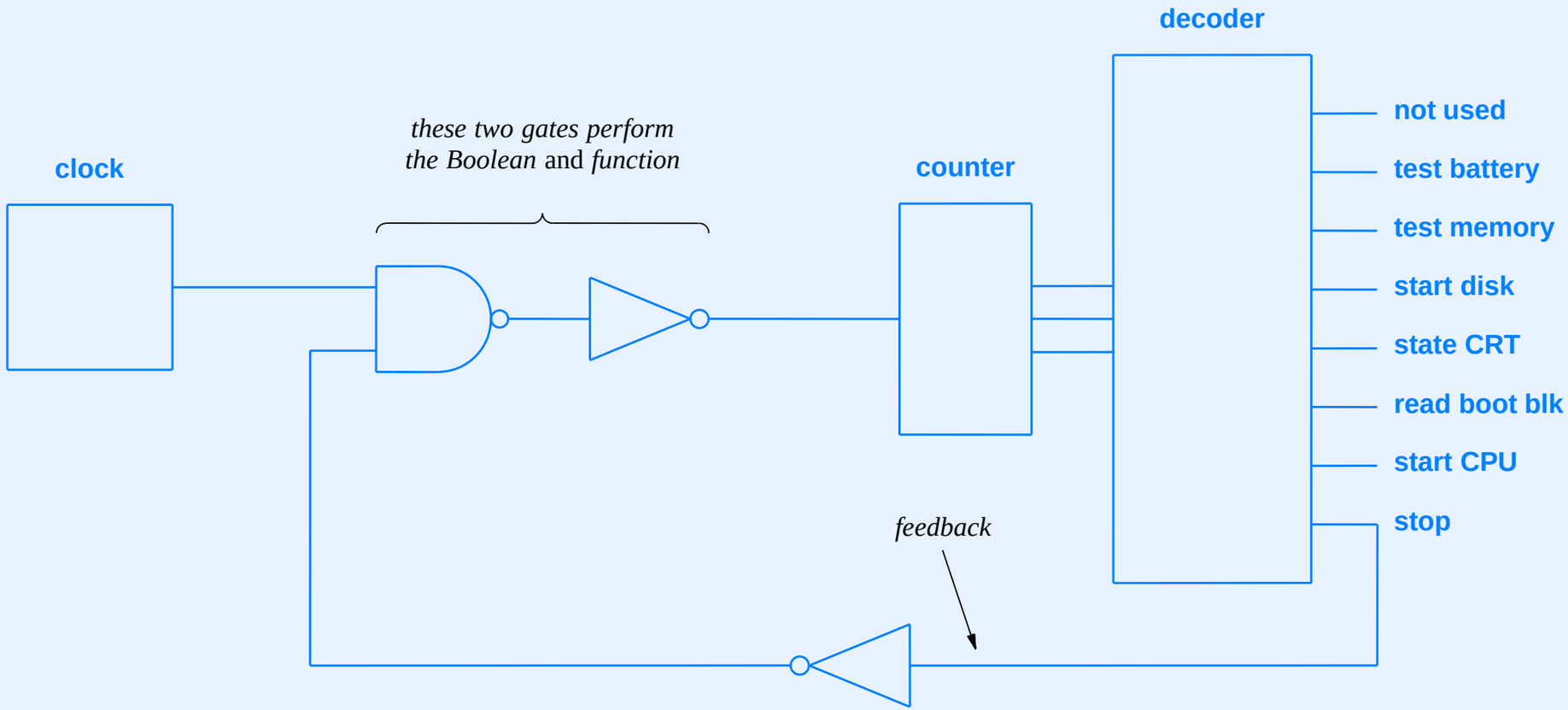
Feedback

- Output of circuit used as an input
- Called *feedback*
- Allows more control
- Example: stop sequence when output *F* becomes active
- Boolean algebra

CLOCK *and* (*not* F)

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Illustration Of Feedback For Termination



- Note additional input needed to restart sequence

Copyright © 2016 by Douglas Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

A Fundamental Difference

- Software
  - Uses iteration
  - Software engineers are taught to avoid replicating code
  - Iteration increases elegance
- Hardware
  - Uses replicated (parallel) hardware units
  - Hardware engineers are taught to avoid iterative circuits
  - Replication increases performance and reliability

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Using Spare Gates

- Note: because chip contains multiple gates, some gates may be unused
- May be possible to reduce total chips needed by employing unused gates
- Example: use a spare nand gate as an inverter by connecting one input to five volts:

$$1 \; nand \; x \quad = \quad not \; x$$

- Previous circuit can be implemented with a single chip (a quad 2-input *nand* gate)

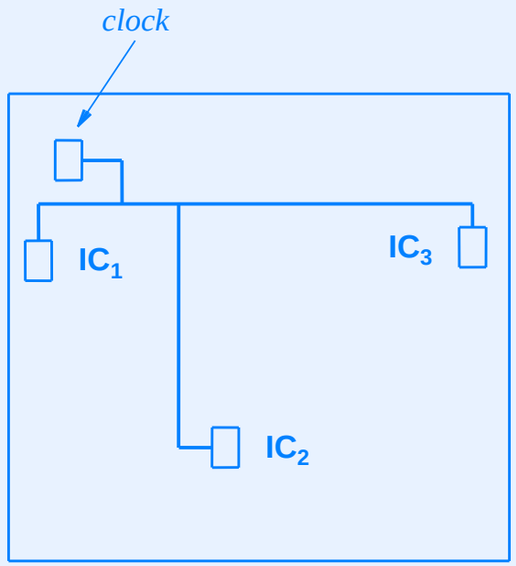
Copyright © 2016 by Douglas Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Practical Engineering Concerns

- Power consumption (wiring must carry sufficient power)
- Heat dissipation (chips must be kept cool)
- Timing (gates take time to settle after input changes)
- Clock synchronization (clock signal must travel to all chips simultaneously)
- Difference in clock signals (*clock skew*) can cause problems

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Illustration Of Clock Skew



- Length of wire determines time required for signal to propagate

Copyright © 2016 by Douglas Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Clockless Logic

- Active circuits built without a clock
- Advantages
  - Possible power savings
  - Avoids clock skew
- Uses two wires to transfer a bit

Wire 1	Wire 2	Meaning
0	0	Reset before starting a new bit
0	1	Transfer a 0 bit
1	0	Transfer a 1 bit
1	1	Undefined (not used)

Copyright © 2016 by Douglas Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Moore's Law And Classifications

- Gordon Moore predicted that the number of transistors on a chip would double each year (revised in 1970 to every 18 months)
- Led to the following classifications

Name	Example Use
Small Scale Integration (SSI)	The most basic logic such as Boolean gates
Medium Scale Integration (MSI)	Intermediate logic such as counters
Large Scale Integration (LSI)	More complex logic such as embedded processors
Very Large Scale Integration (VLSI)	The most complex processors (i.e., CPUs)

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Other Terminology Associated With Chips

- ASIC (Application-Specific Integrated Circuit)
  - Custom design for a specific product
  - Used when higher speed is needed
- SoC (System on Chip)
  - Single IC that contains one or more processors, memories, and I/O device interfaces all interconnected to form a working system
  - Used in many low-end devices

Copyright © 2016 by Douglas Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Levels Of Abstraction

- Digital systems can be described at various levels of abstraction
- Some examples

Abstraction	Implemented With
Computer	Circuit board(s)
Circuit board	Components such as processor and memory
Processor	VLSI chip
VLSI chip	Many gates
Gate	Many transistors
Transistor	Semiconductor implemented in silicon

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Reconfigurable Logic

- Alternative to standard gates
- Allows chip to be configured multiple times
- Can create
  - Various gates
  - Interconnections
- Typical approach: view a gate as an array and inputs as an index
- Most popular form: *Field Programmable Gate Array* (*FPGA*)

Copyright © 2016 by Douglas Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual

Summary

- Computer systems are constructed of digital logic circuits
- Fundamental building block is called a *gate*
- Digital circuit can be described by
  - Boolean algebra (most useful when designing)
  - Truth table (most useful when debugging)
- Clock allows active circuit to perform sequence of operations
- Feedback allows output to control processing
- Practical engineering concerns include
  - Power consumption and heat dissipation
  - Clock skew and synchronization

Copyright © 2016 by Douglas C. Comer. All rights reserved.
Essentials of Computer Architecture 2nd edition Comer Solutions Manual


Questions?