Computer Organization

Douglas Comer

Computer Science Department
Purdue University
250 N. University Street
West Lafayette, IN 47907-2066

http://www.cs.purdue.edu/people/comer

II

Fundamentals Of Digital Logic

Our Goal

- Understand
  - Fundamentals and basics
  - Concepts
  - How computers work at the lowest level
- Avoid whenever possible
  - Complexity
  - Implementation details
  - Engineering design rules

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)*

Analog For Electricity

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

Voltage

- Device used to measure called *voltmeter*
- Can only be measured as difference between two points
- To measure voltage
  - Assume one point represents zero volts (known as *ground*)
  - Express voltage of second point wrt ground

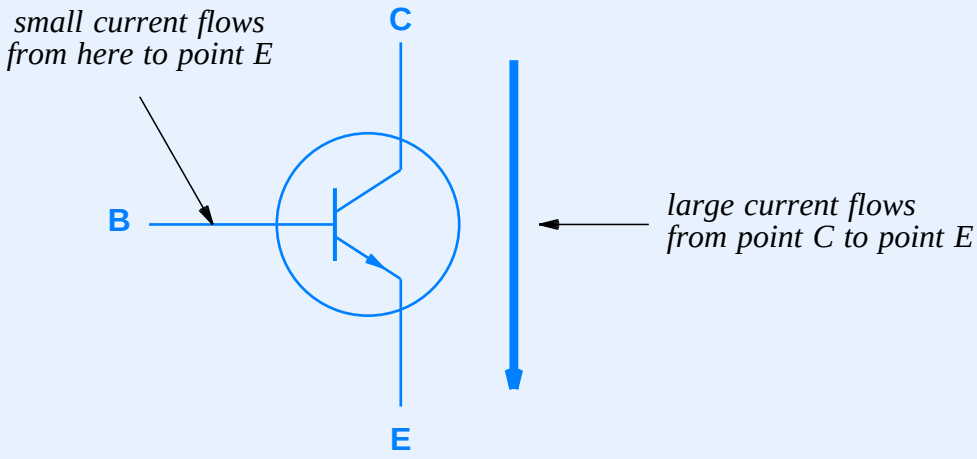
In Practice

- Typical digital circuit operates on five volts
- Two wires connect each chip to *power supply*
  - Ground (zero volts)
  - Power (five volts)
- Digital logic diagrams do not usually show power and ground connections

Transistor

- Basic building block of digital circuits
- Operates on electrical current
- Acts like a miniature switch — small input current controls flow of large current
- Three external connections
  - Emitter
  - Base (control)
  - Collector
- Current between base and emitter controls current between collector and emitter

Illustration Of A Transistor



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

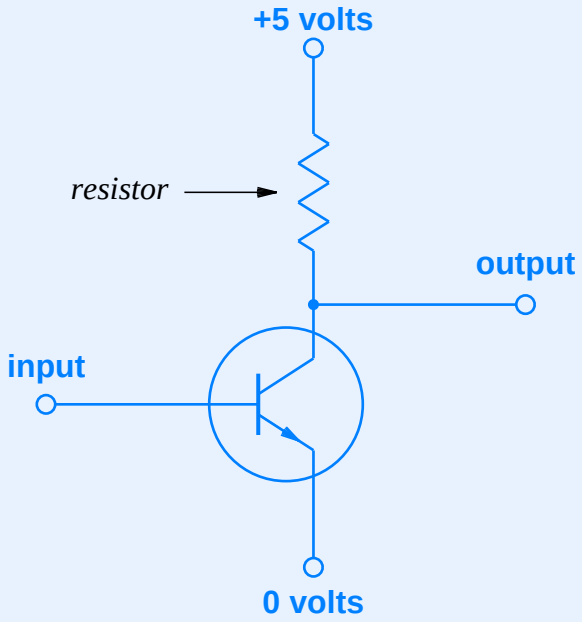
A	B	A or B
0	0	0
0	1	1
1	0	1
1	1	1

A	not A
0	1
1	0

Digital Logic

- Can implement Boolean functions with transistors
- Five volts represents Boolean *1*
- Zero volts represents Boolean *0*

Transistor Implementing Boolean Not



- When input is zero volts, output is five volts
- When input is five volts, output is zero volts

Logic Gate

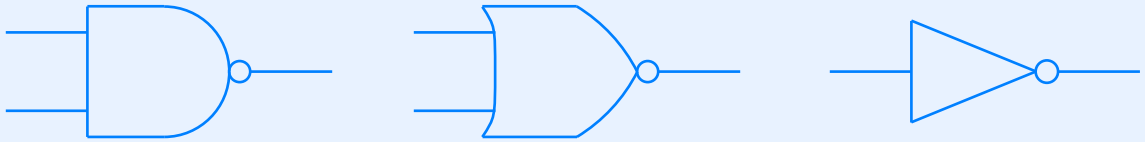
- Hardware component
- Consists of integrated circuit
- Implements an individual Boolean function
- To reduce complexity, provide inverse of Boolean functions
  - Nand gate implements *not and*
  - Nor gate implements *not or*
  - Inverter implements *not*

Truth Tables For Nand and Nor 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

Symbols Used In Schematic Diagrams

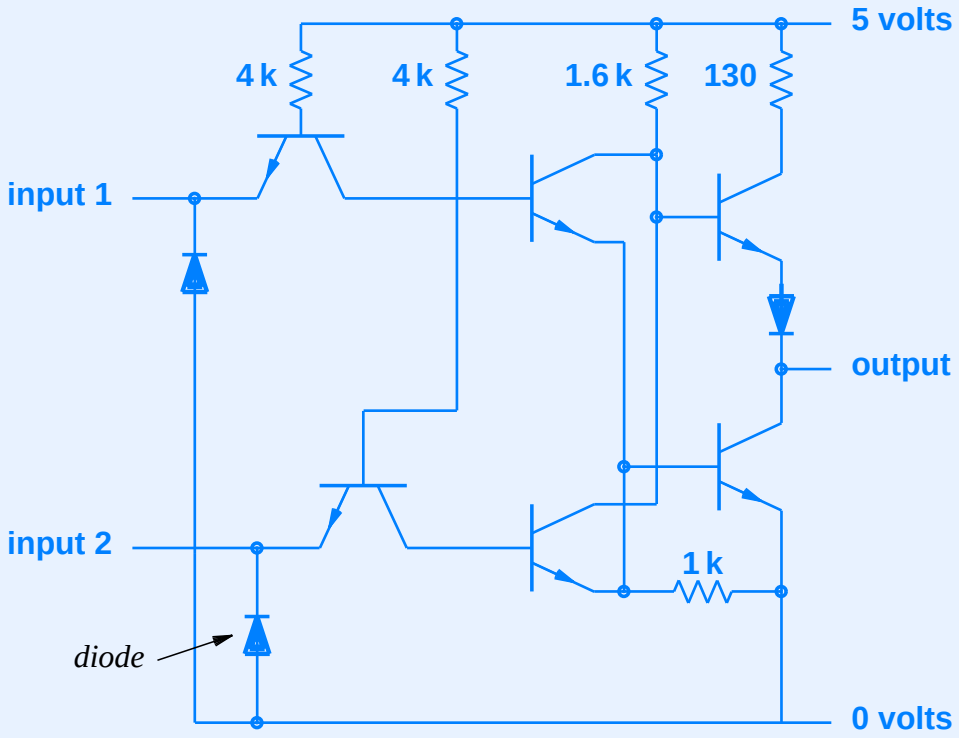


nand gate

nor gate

inverter

Example Of Internal Gate Structure (Nor Gate)



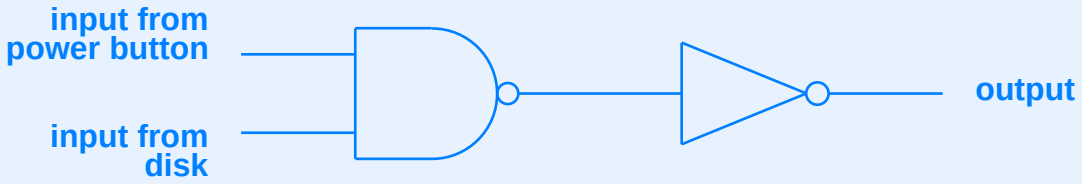
- Solid dot indicates electrical connection

Technology For Logic gates

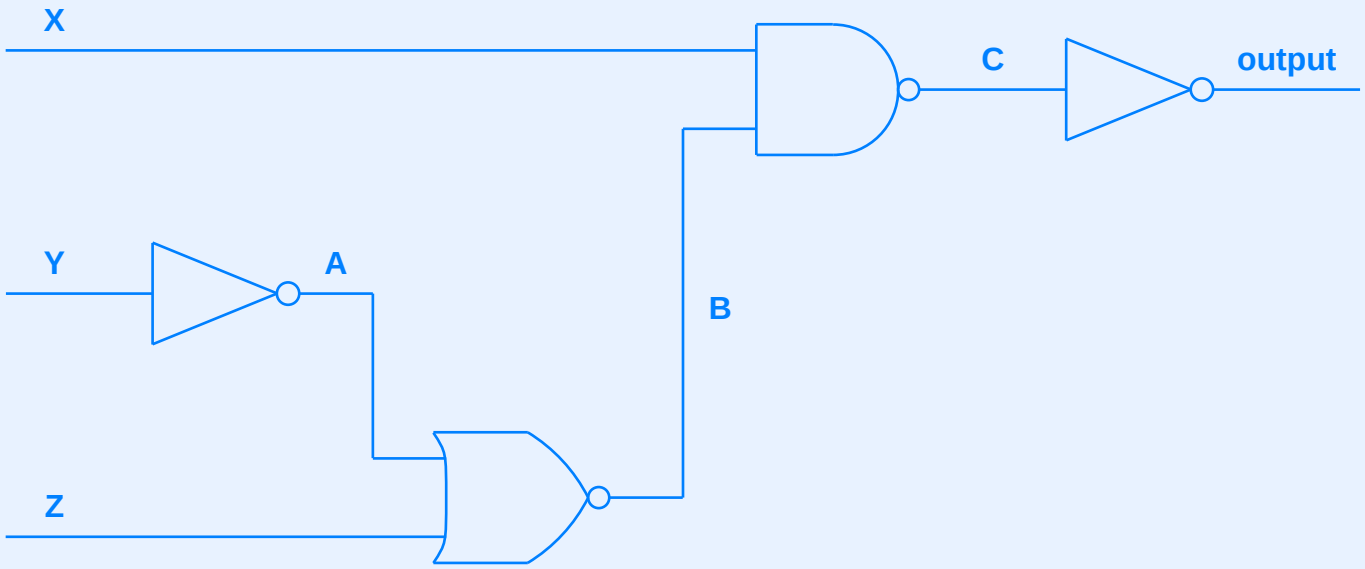
- Most popular technology known as *Transistor-Transistor Logic* (*TTL*)
- Allows direct interconnection (a wire can connect output from one gate to input of another)
- Single output can connect to multiple inputs
  - Called *fanout*
  - Limited to a small number

Example Interconnection Of TTL Gates

- Two logic gates needed to form logical *and*
  - Output from nand gate connected to input of inverter



Consider The Following Circuit

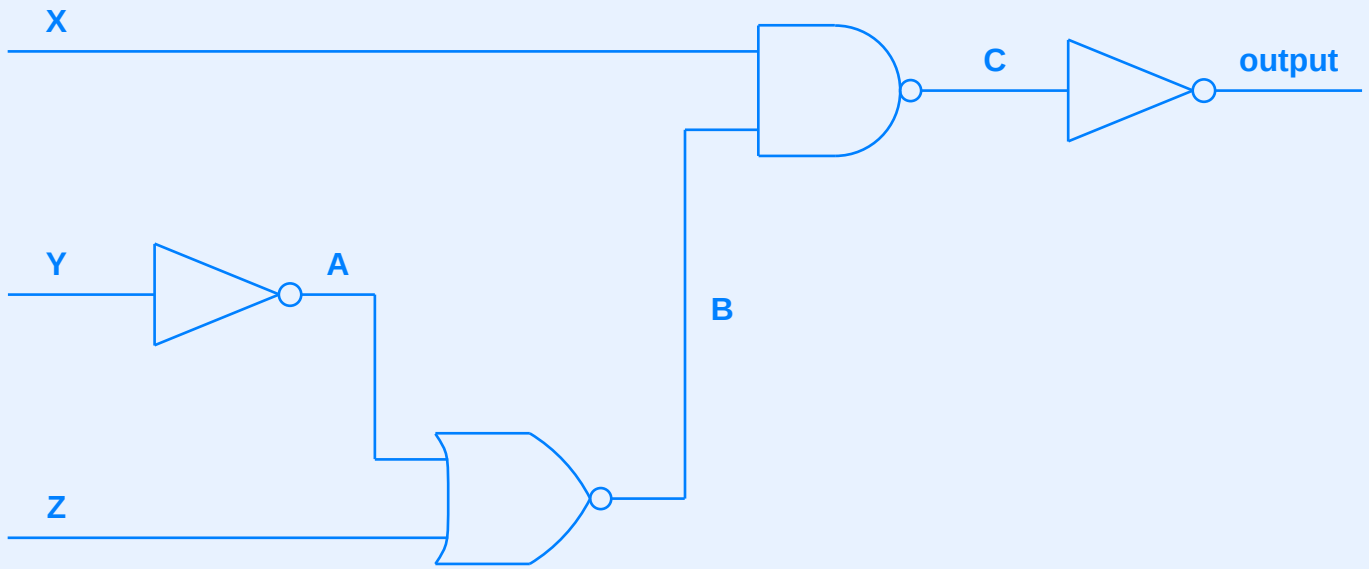


- Question: what does the circuit implement?

Two Ways To Describe Circuit

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

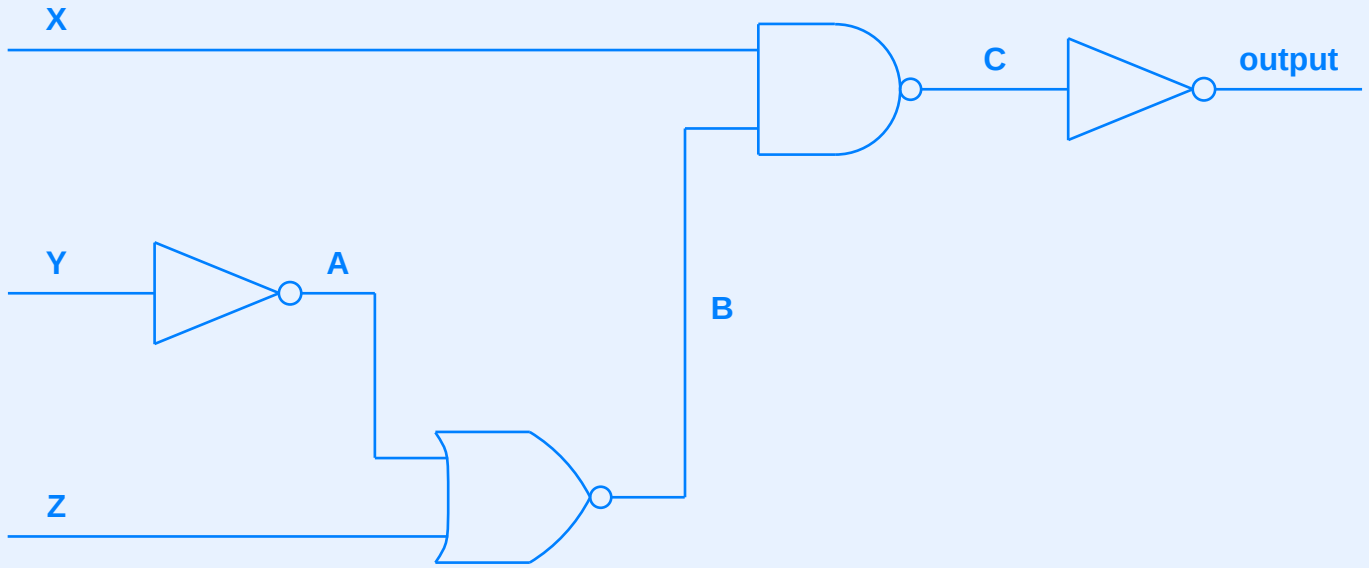
Describing A Circuit With Boolean Algebra



- Value at point *A* is *not Y*
- Value at *B* is:

Z nor (not Y)

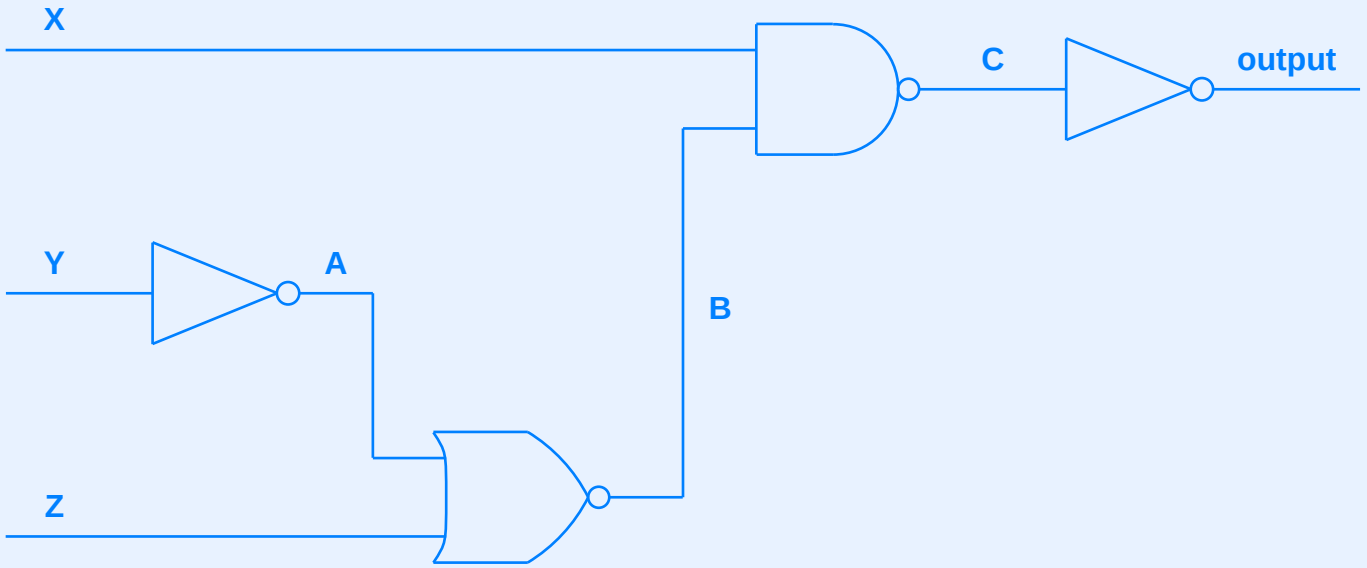
Describing A Circuit With Boolean Algebra (continued)



- Output is:

X and (Z nor (not Y))

Describing A Circuit With Boolean Algebra (continued)



- Output is (alternative):

X and not (Z or (not Y))

Describing A Circuit With A Truth Table (continued)

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

Avoiding Nand / Nor Operations

- Circuits use nand and nor gates
- Sometimes easier for humans to use *and* and *or* operations
- Example circuit or truth table output can be described by Boolean expression:

X and Y and (not Z))

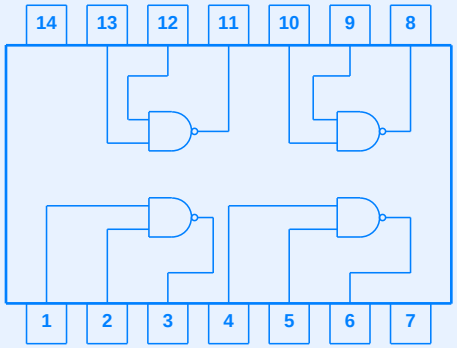
In Practice

- Only a few connections needed per gate
- Chip has many pins for external connections
- Result: can package multiple gates placed on each chip

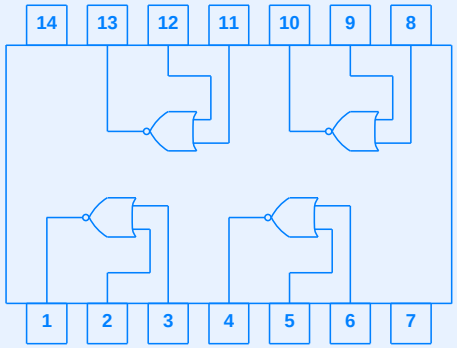
Example Of Logic Gates

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

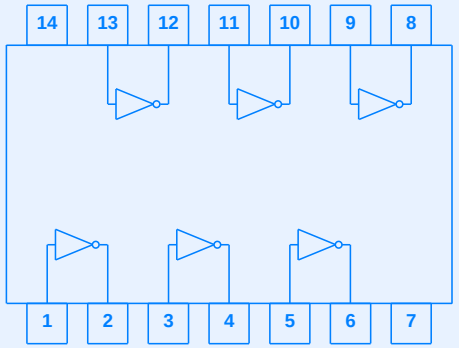
Examples Of Gates On 7400-Series Chips



7400



7402



7404

- Pins 7 and 14 connect to ground and power

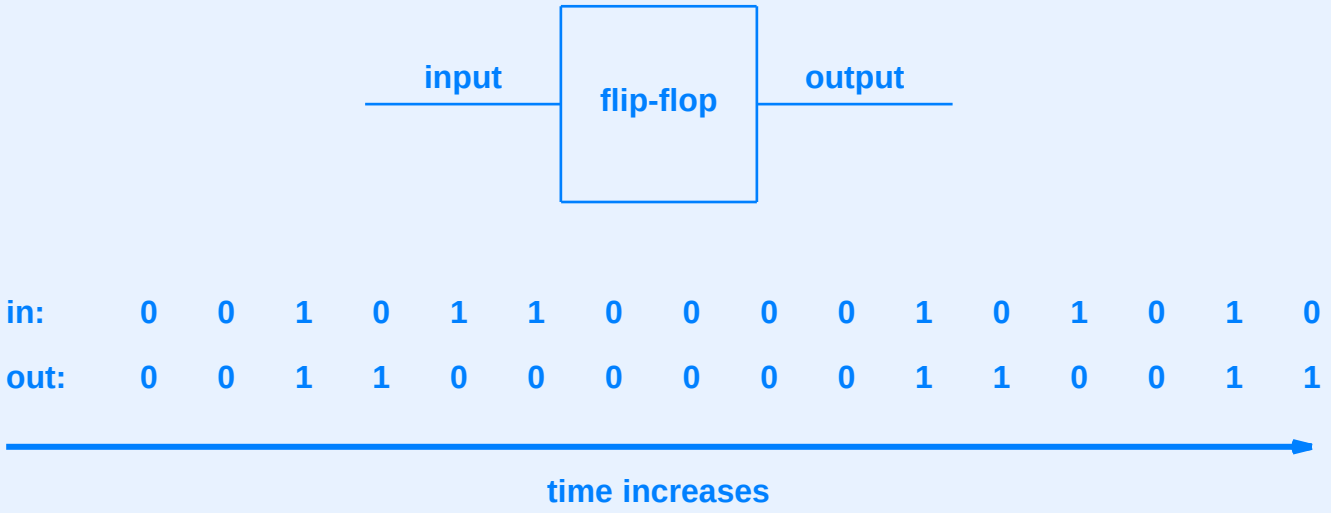
Circuits That Maintain State

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

Example Of Circuit That Maintains State

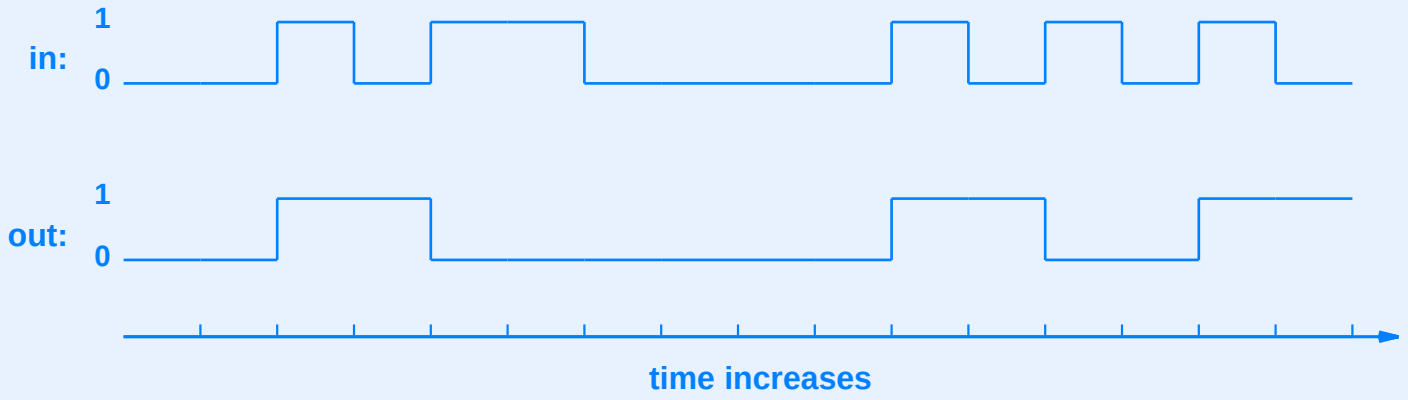
- Basic *flip-flop*
- Analogous to push-button power switch
- Each new 1 received as input causes output to reverse
  - First input pulse causes flip-flop to turn on
  - Second input pulse causes flip-flop to turn off

Output Of A Flip-Flop



- Note: output only changes when input makes a transition from zero to one

Flip-Flop Action Plotted As Transition Diagram

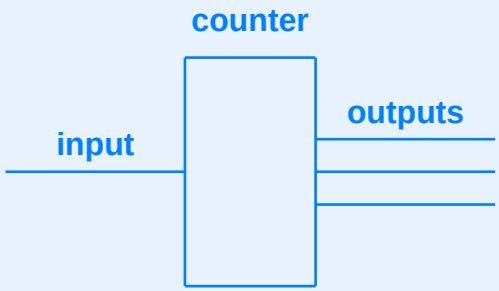


- Output changes on *leading edge* of input
- Also called *rising edge*

Binary Counter

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

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)

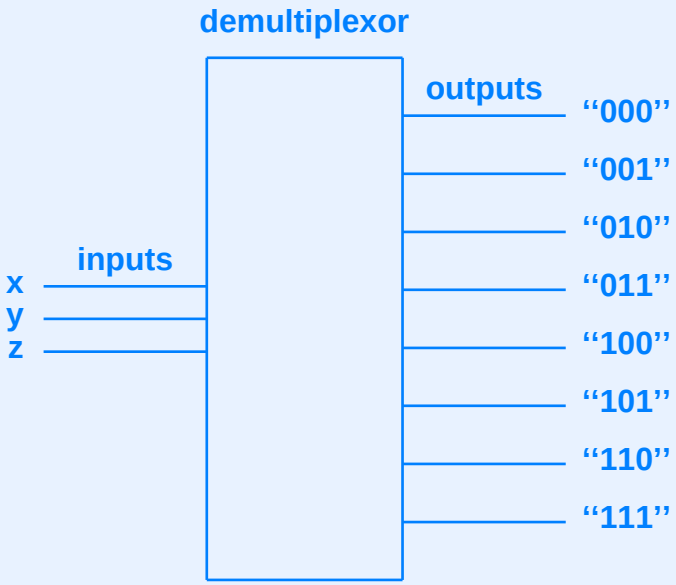
Clock

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

Demultiplexor

- Takes binary value as input
- Uses input to select one output

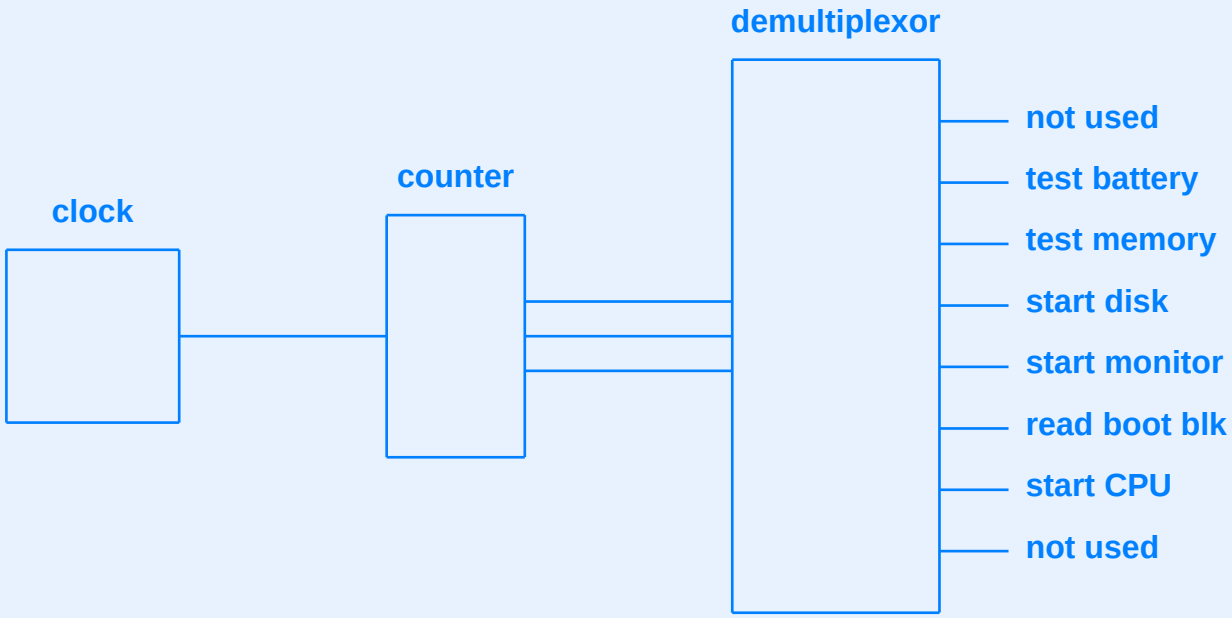
Illustration Of Demultiplexor



Example Circuit That Executes A Sequence Of Steps

- Desired sequence
  - Test the battery
  - Power on and test the memory
  - Start the disk spinning
  - Power up the monitor
  - Read boot sector from disk into memory
  - Start the CPU

Circuit To Execute Sequence

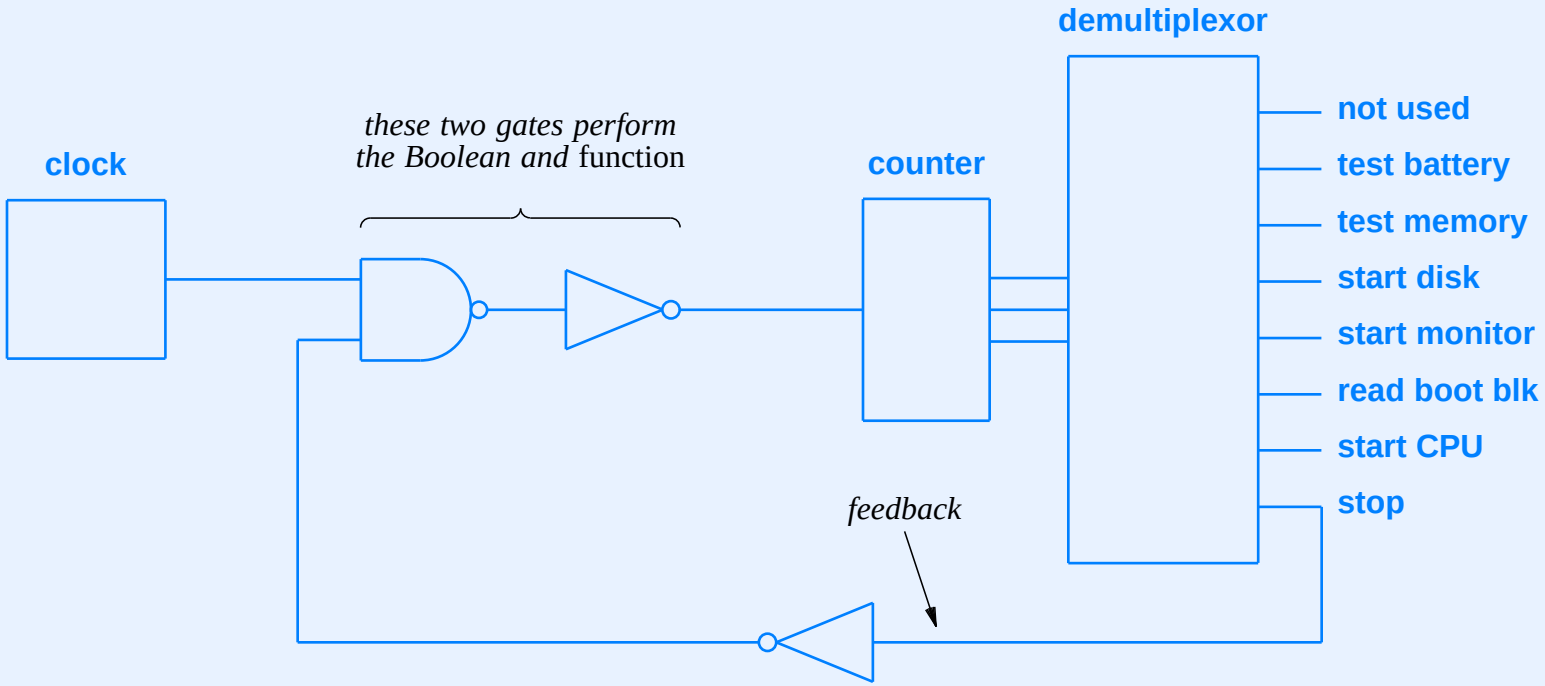


Feedback

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

CLOCK *and* (*not* F)

Illustration Of Feedback For Termination



- Note additional input needed to restart sequence

Spare Gates

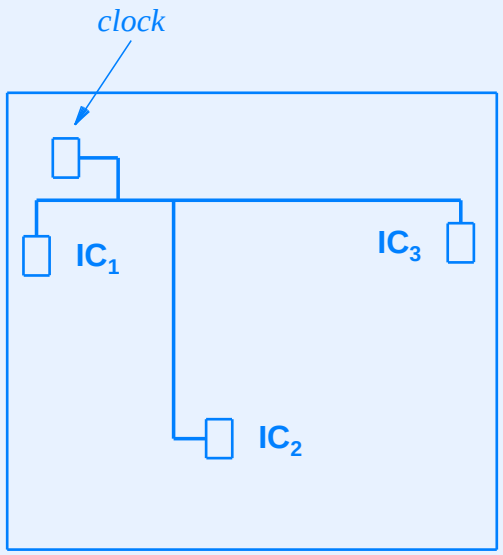
- Note: because chip contains multiple gates, some gates may be unused
- May be able to substitute spare gates in place of additional chip
- Example uses spare nand gate as inverter by connecting one input to five volts:

1 *nand* x = *not* x

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 reach all chips simultaneously)

Illustration Of Clock Skew



- Length of wire determines time required for signal to propagate

Classification Of Technologies

Name	Example Use
Small Scale Integration (SSI)	Basic Boolean gates
Medium Scale Integration (MSI)	Intermediate logic such as counters
Large Scale Integration (LSI)	Small, embedded processors
Very Large Scale Integration (VLSI)	Complex processors

Levels Of Abstraction

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

Summary

- Computer systems are constructed of digital logic circuits
- Fundamental building block is *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


Questions?