Binary Storage

- Binary Cell: A Device that Possesses Two Stable States
- Cell Input: Receives Data and Control Signals that Set it to One of the States
- Cell Output: Physical Quantity Indicating Which State the Cell is in
- States are Encoded as Binary Digits $\{0,1\}$

Registers

- Group of n Binary Cells: an n-bit Register
- Register has 2^n States: All Possible n-bit Strings
- Register State (or content) can be *INTERPRETED* as a Value, ASCII, etc.
- Registers Classified as to Type of Input and Output – Serial and/or Parallel

Much More on This Later in the Class

Register Transfer

- General Way to Describe a Digital Circuit (including a computer)
- Registers are Interconnected
- At a Given Time Content of One Register *Transferred* to Another
- Content May be *Transformed/Changed* During Transfer
- Transforming Circuit is a *Data Processing* or *Data Path* Element/Circuit

Register Transfer (cont)

- General Way to Describe a Digital Circuit
  - Can Use Diagrams
  - Can Use *Register Transfer Languages*
- Register Transfer Descriptions are Often the First Descriptions Developed in Design Process
- Example: Keyboard to Registers to Memory
  - Key Strike: 8-bit ASCII to Right-most Register
  - Key Strike: Register Content Shifted Left
  - 4 Key Strikes: Content Transferred to Memory

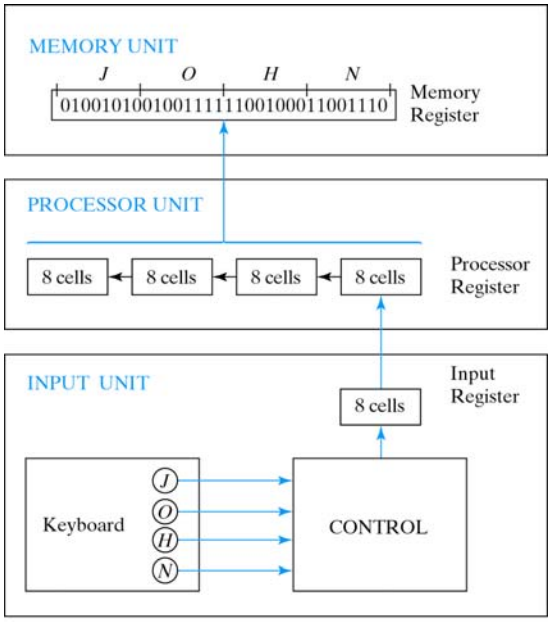


Fig. 1-1 Transfer of information with registers

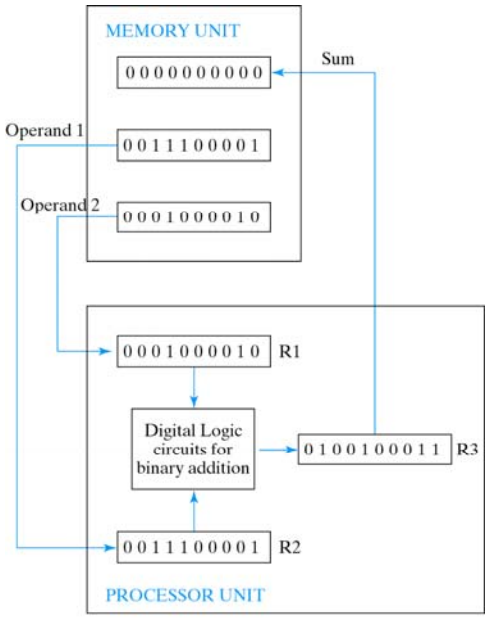


Fig. 1-2 Example of binary information processing

Binary Logic

- Deals With Variables that have Two Different Values – Typically BITs $\{0,1\}$
- Logic is Based on *Boolean Algebra* Operators
- Boolean Algebra DOES NOT Necessarily Have to Be Based on Binary Values
- Formal Introduction in Following Sections

Binary Logic

- Variables Denoted by Letters of Alphabet
  - *EXAMPLES*

    $A, B, C, a, b, c, x_1, x_2, x_3$, etc.
- Three Basic Operations Considered (for now)
  - AND (Boolean Algebra Multiplication)
  - OR (Boolean Algebra Addition)
  - NOT (Boolean Algebra Inversion)

Binary AND Operation

- Represented by a Dot • *OR* Absence of Operator Between Adjacent Variables or Constants

  $x \bullet y = z$ $\quad\quad$ $xy = z$

  x AND y is equal to z

  $z=1$ if and only if x AND $y=1$, otherwise $z=0$

- Truth Table Definition:

x	y	$x \bullet y$
0	0	0
0	1	0
1	0	0
1	1	1

Binary OR Operation

- Represented by a Plus Sign + Between Adjacent Variables or Constants

  $x+y=z$

  x OR y is equal to z

  $z=1$ if $x=1$ OR $y=1$ OR both $x=y=1$, otherwise $z=0$

- Truth Table Definition:

x	y	$x+y$
0	0	0
0	1	1
1	0	1
1	1	1

Binary NOT Operation

- Represented by a Prime Sign ’ OR an Overbar
- Unary Operator – Operates on One Variable

  $x'=z$ $\quad\quad$ $\overline{x}=z$

  NOT x is equal to z

  if $x=1$ THEN $z=0$ ELSE both $z=1$
- Truth Table Definition:

x	$z=\overline{x}$
0	1
1	0

Logic Gates

- Electronic Circuits Operate on 1 or More Input Signals
- Produce an Output Signal Corresponding to a Boolean Operation
- Signals are Voltages or Currents
- Signals Interpreted as a 0 or 1
- Defined by a Range of Voltage or Current Values

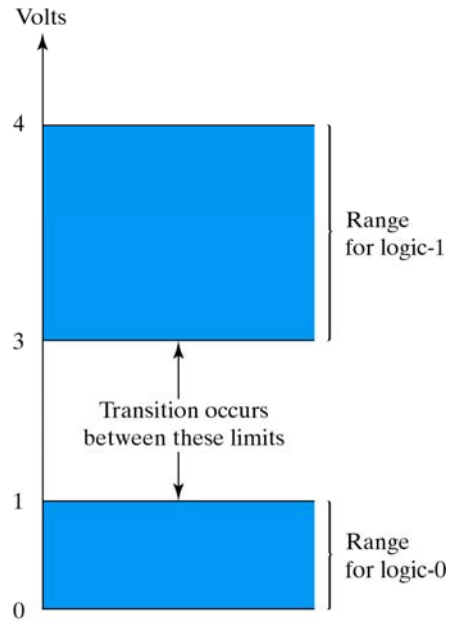
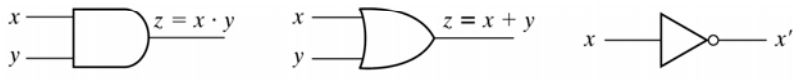


Fig. 1-3 Example of binary signals

Logic Gates

- Circuit Diagrams Use Graphical Symbols for Logic Gates
- Connections of Logic Gates Between Registers can Transform Data
- Logic Gates Represent Interconnections of Transistors and Other Electronic Components



(a) Two-input AND gate (b) Two-input OR gate (c) NOT gate or inverter

Fig. 1-4 Symbols for digital logic circuits

Timing Diagrams

- Depict Input and Output Signals as Functions of Time
- Values Toggle Between Two Discrete Values

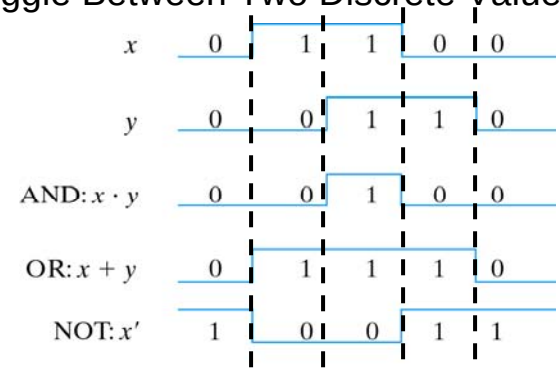
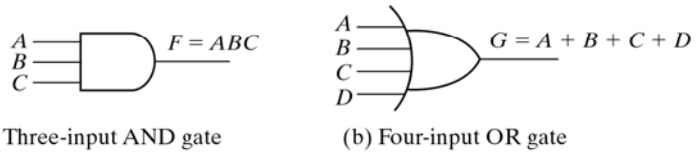


Fig. 1-5 Input-output signals for gates

Multi-input Logic Gates

- Many Logic Gates Can Have More than Two Inputs
- Examples are AND and OR Gates



(a) Three-input AND gate (b) Four-input OR gate

Fig. 1-6 Gates with multiple inputs