Chapter 3

Connecting with Computer Science
Greg Anderson
David Ferro
Robert Hilton

Computer Architecture
Objectives

- Learn why you need to understand how computers work
- Learn what a CPU is, and what it is made of
- Learn how digital logic circuits are constructed
- Learn the basic Boolean operators
- Understand how the basic logic gates operate and are used to build complex computer circuits
Objectives (continued)

• Learn the importance of von Neumann architecture

• Understand how the computer uses memory

• Learn what a system bus is and what its purpose is

• Understand the difference between memory and storage
Objectives (continued)

- Be able to describe basic input/output devices
- Understand how a computer uses interrupts and polling
Why You Need to Know About... Computer Architecture

• Computer: hardware designed to run software
  – Purpose: accomplish desired tasks

• Professional’s need: understand logical connection between hardware and software

• Computer architecture: organization of hardware components into a computer system
Inside the Box

• Computer system: an external view
  – Monitor
  – Keyboard and mouse
  – Computer case

• CPU (central processing unit)
  – Resides in case on main board or motherboard
  – Computational center served by all other parts
  – Touch point for the study of computer architecture
Figure 3-1, Typical personal computer system

Courtesy of Fujitsu

Connecting with Computer Science
Figure 3-2, Main board with labeled components

- BIOS chip
- PCI bus slots
- AGP video connector
- Power supply connector
- Serial and parallel ports
- CPU clock
- CPU
- Memory slots
- EIDE device connectors
The CPU

- The CPU is the computer
  - Contains digital components that do processing
  - Fundamental component: transistor

- Transistor
  - Electronic switch accommodates binary values
  - Millions of transistors per chip
  - Organized into a higher level called a circuit
The CPU (continued)

- Four basic functions: Adding, Decoding, Shifting, Storing

- Four corresponding transistor circuits
  - Adder: adds, subtracts, multiplies, divides
  - Decoder: reacts to specific bit patterns
  - Shifter: moves bits to right or left
  - Flip-flops (latches): used to store memory bits
How Transistors Work

- Material composition: silicon or germanium
- Logically organized into three parts: emitter, collector, and base
- Transistor as electronic switch
  - Base used to turn current on and off
  - Capacity to control current translates into capacity to manipulate binary values of 1 and 0
- Size considerations: Typical transistor 130 nanometers wide (Pentium IV)
Figure 3-3, Transistors are used to build logic circuits such as this NOT gate.

- Power supply
- Base
- Collector
- Emitter
- Input
- Ground

When a voltage is placed on the base, the collector voltage goes toward the ground.

The transistor can only conduct electricity if a voltage is placed on the base.
Digital Logic Circuits

• Logic circuit
  – Next level of organization above transistor
  – Leverages switching function of transistor
  – Performs operations of Boolean algebra

• Boolean algebra
  – Functions relating binary input and output
  – Chief operators: AND, OR, NOT
  – Boolean variables are true (1) or false (0)
  – Boolean expressions use Boolean operators and variables
Digital Logic Circuits (continued)

- Truth tables
  - Convenient tabular representations of Boolean expressions
  - Column(s) represent inputs and output(s)
  - Rows correspond to each possible combination of inputs
    - \(2^n\) rows needed for \(n\) inputs (\(n\) is positive integer)
    - Example: two inputs require \(2^2 = 4\) rows
The Basic Boolean Operators

- Three Basic Operators: AND, OR, NOT
- AND operator
  - Takes two values as input (x and y) and generates one output (z)
  - Both inputs must be true (1) for output to be true (1)
    - When x = 1 and y = 1, z = 1
  - Any other combination yields output of false (0)
  - Equivalent Boolean Expression: $xy = z$
Figure 3-4 Truth table for the AND operator

<table>
<thead>
<tr>
<th>inputs</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
The Basic Boolean Operators (continued)

- OR operator
  - Takes two values as input (x and y) and generates one output (z)
  - Either input valued true (1) will cause output to be valued true (1)
  - When both inputs are valued false (0), output will be valued false (0)
  - Equivalent Boolean expression: $x + y = z$
Figure 3-5, Truth table for the OR operator

<table>
<thead>
<tr>
<th>inputs</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
The Basic Boolean Operators (continued)

- NOT operator
  - Takes one value as input (x) and generates one output (z)
  - Reverses value of input.
    - When \( x = 1 \), \( z = 0 \)
    - When \( x = 0 \), \( z = 1 \)

Equivalent Boolean expression: \( x' = z \) or \( x = z \)
Figure 3-6, Truth table for the NOT operator

<table>
<thead>
<tr>
<th>input</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>z</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Digital Building Blocks

• Circuit Hierarchy
  – Gates: transistor circuits implement Boolean operators
  – Gates grouped into more complex circuits that carry out computer tasks

• Reliability
  – Binary values maintained with consistent voltage levels
  – Output of gates completely determined by input

• Six fundamental gates
  – AND, OR, NOT (primitive or basic)
  – NAND, NOR, XOR (constructed from AND, OR, NOT)
Digital Building Blocks (continued)

- **AND gate**
  - Allows for two inputs and has one output
  - Truth table identical to that of AND Boolean Operator
Digital Building Blocks (continued)

- OR gate
  - Allows for two inputs and has one output
  - Truth table identical to that of Boolean OR

![Figure 3-8, Symbol and truth table for the OR](image)
Digital Building Blocks (continued)

- NOT gate
  - Allows for one input and one output
  - Truth table identical to Boolean NOT

Figure 3-9, Symbol and truth table for the NOT gate
Digital Building Blocks (continued)

- NAND gate
  - Reverses output of AND gate with NOT gate
  - Output opposite that of AND gate

![Figure 3-10, Symbol and truth table for the NAND gate](image)
Digital Building Blocks (continued)

- NOR gate
  - Reverses output of OR gate with NOT gate
  - Truth table opposite of OR gate truth table
Digital Building Blocks (continued)

- XOR gate
  - Exclusive OR (one or the other, but not both)
  - Inputs with opposite values generate output of 1

Figure 3-12, Symbol and truth table for the XOR gate
Gate Behavior

• Predictability of Gates
  – Predicated on corresponding truth table
  – Output for given input derived from truth table

• Scalability
  – Complex circuits constructed with basic gates
  – Output of one gate is connected as input to another
  – Example: 3-input AND gate from two 2-input AND gates
Figure 3-13, Constructing a 3-input AND gate from two 2-input AND gates

<table>
<thead>
<tr>
<th>w</th>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Complex Circuits

- Four Fundamental circuits of CPU
  - ADDER, DECODER, SHIFTER, FLIP-FLOP
  - Constructed from basic gates

- ADDER
  - Function: adds two binary numbers, outputs result
  - Inputs: two bits (x, y) to add and one carry-in (ci)
  - Outputs: sum bit (s) and one carry-out bit (co)
Figure 3-14, Truth table for adding two bits with carry-in and carry-out

<table>
<thead>
<tr>
<th>inputs</th>
<th>outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Figure 3-15, Adder circuit
Complex Circuits (continued)

- Decoder
  - Functions: Address memory, select I/O devices
  - Given input pattern, output line selected
  - Circuit with n inputs can control $2^n$ lines
    - Example: 32 inputs could control 4 billion lines
  - Illustrate Decoder with two inputs
    - Has four possible outputs
    - Truth table incorporates four basic truth tables
Figure 3-16, Decoder with two input lines controlling four output lines

<table>
<thead>
<tr>
<th>inputs</th>
<th>outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Complex Circuits (continued)

- Flip-flop
  - Special form of latch circuit
  - Function: holds value at output even if input changes
  - Inputs: S (set) and R (reset)
  - Outputs: Q and Q'
  - Ideal for bit storage
    - Used for high-speed memory in CPU
    - Static RAM, SRAM, cache
Figure 3-17, A basic SR (set and reset) flip-flop circuit implemented using NOR gates
Complex Circuits (continued)

- Shifter
  - Purpose: support math operations such as multiplication and division
  - Function: shifts input bits to the left or right

Figure 3-18, Inputs and outputs of a shifter circuit (one-bit right shift)
Complex Circuits (continued)

• Other circuits: multiplexer, parity generator, counter

• Three part design process
  – Construct truth table relating inputs and outputs
  – Build Boolean expression equivalent to truth table
  – Represent Boolean expression in a circuit diagram
Complex Circuits (continued)

- Modern design efforts supported with software

- Integrated Circuits (ICs)
  - Whole circuits etched onto single semiconductor chip
  - VLSI (Very Large-Scale Integration) chip
    - Contains millions of transistors making up CPU circuits
Von Neumann Architecture

• Multipurpose machine with following characteristics:
  – Instruction cycle: Fetch-Decode-Execute
  – Instructions and data are stored in main memory
  – Instruction execution carried out by central processing unit (CPU)
Von Neumann Architecture (continued)

- CPU consists of:
  
  - Control unit (CU)
  
  - Arithmetic logic unit (ALU)
  
  - Registers (small storage areas).

- The CPU interacts with input and output devices
Figure 3-19, Von Neumann architecture

- Input device
- Central processing unit
  - Control unit
  - Arithmetic/logic unit
  - Registers
  - Main memory
- Output device
- Auxiliary storage device
Von Neumann Architecture (continued)

• Breakdown of typical fetch-decode-execute cycle:
1. Control unit uses the address in program counter register to fetch an instruction from main memory
2. Instruction decoded
3. Any needed data retrieved from memory and placed into other registers
4. ALU executes the instruction using data in registers, if necessary
5. Input or output operations required by the instruction are performed
Von Neumann Architecture (continued)

• Crystal (system) clock synchronizes steps in instruction sequence
  – Computers measured by clock speed
  – Example: Pentium IV speed = 3 GHz. Processes 3 billion instruction cycles per second

• Trends in clock speed
  – Rising for 60 years
  – Laws of Physics erected speed barrier at 100 MHz
  – Parallelism and caching work around speed limit
Buses

- Bus: set of wires and rules facilitating data transfer
- Components connected via system bus
- Bus wires divided into three separate signal groups:
  - Control
  - Address
  - Data
- Modern bus standard: PCI (Peripheral Component Interconnect)
Peripheral Buses

- SCSI: Small Computer System Interface
  - Connects different types of I/O devices to computer
  - Allows CPU to pass control to other devices (bus mastering)
Storage

- Storage: family of components used to store programs and data

- Storage hierarchy
  - Primary memory
  - Secondary memory (mass storage)
Memory

- Two basic flavors
  - ROM (read-only memory)
    - Memory etched into chip
    - Generally cannot be modified
    - BIOS (basic input/output system) in this category
  - RAM (random access memory) basics
    - Allows direct memory reference
    - Allows reading and writing
    - Volatile
    - CPU fetches program instructions from RAM
Memory (continued)

• Types of RAM
  – DRAM (Dynamic RAM)
    • Made of circuits using one transistor per bit
    • Needs to be constantly refreshed to maintain data
    • Access speeds ranging from 10 – 70 ns
  – SRAM
    • Made of flip flop circuits
    • Fastest memory type
    • Used chiefly in registers and cache
Mass Storage

• Secondary memory characteristics
  – Greater storage capacity than RAM or ROM
  – Used for devices such as hard drive or CD-ROMS
  – Cheaper storage per megabyte than RAM or ROM
  – Non-volatile
  – Slower than RAM or ROM
Mass Storage (continued)

- Hard disk drives
  - Most common form of mass storage
  - Magnetic metal platters store information
    - Organized into concentric circles called tracks
    - Tracks divided into sectors
    - Platters spin at about 7200 RPM
    - Read/write heads interface with disk surface
  - Low cost-unit storage ratio relative to RAM
  - RAID (Redundant Array of Independent Discs)
Figure 3-20, Hard drive platters and read/write heads
Mass Storage (continued)

- Optical Storage
  - Popular formats: CDs (compact disks) and DVDs (digital video disks)
  - Stores data using optical (laser) technologies
    - Pits burned into disks interpreted as binary data
    - Data written to disks in continuous spiral
    - Like hard disks, CDs and DVDs spin
    - Read/write heads interface with disk surface
  - Low cost-unit storage ratio
Mass Storage (continued)

- Flash (Thumb) drives
  - Portable storage that plugs into USB (Universal Serial Bus) port
  - Replacing floppy drives
  - Uses non-volatile flash memory
  - Low cost-unit storage ratio
Input/Output Systems

- Final component of Von Neumann architecture
- Input/Output (I/O) devices: computer’s connection to user
Input Devices

- **Keyboard**
  - Chief input device for most users
  - Connected to motherboard through port and then to CPU by controller circuit and system bus
  - Keystrokes are translated into binary signals for CPU consumption

- **Mouse**
  - Used in conjunction with keyboard
  - Senses movement that can be translated into binary code

- **Other devices:** trackballs, styluses, touch pads/screens
Figure 3-21, The motherboard provides numerous ports to connect peripheral devices
Output Devices

- Communication channel to outside world
- Monitors
  - Primary output device
  - CRTs (cathode ray tubes)
    - Utilizes raster scanning techniques
    - Quality based on resolution and refresh rate
  - LCD (Liquid Crystal Display)
    - Thinner and cooler than CRTs
    - Utilizes transistors rather than electron beams
    - Quality based on resolution and refresh rate
Figure 3-22 Comparison of LCD and CRT monitors

Courtesy of NEC Mitsubishi

Connecting with Computer Science
Output Devices (continued)

- Printers
  - Important output device
  - Chief varieties: ink jet and laser printers
  - Quality measured by resolution (dots per inch) and speed (pages per minute)

- Sound cards
  - Fit into PCI expansion slot on main board
  - Used to digitize sound for storage
  - Also converts binary sound files into analog sounds
Interrupts and Polling

• CPU instruction cycle equals clock speed

• CPU commits to cycle based on processing need

• Processing need determined by (2) techniques
  – Polling: CPU interrogates I/O device
  – Interrupt Handling: I/O device initiates request for service
Choosing the Best Computer Hardware

- No one size fits all
- Circumstances drive selection process
- Some criteria
  - Define objectives for machine
  - Ascertain metrics such as clock speed, memory type, bus speed, hard drive speed
  - Compare models
  - Solicit opinions
  - Consider price
One Last Thought

- All computer scientists need architectural concepts
- Advanced hardware studies highly recommended
- Stay current on latest industry developments
Summary

- Computer architecture: hardware design
- Von Neumann architecture: design template for modern machines
- Components of Von Neumann machine:
  - Central Processing Unit (CPU)
  - Memory (organized in a hierarchy)
  - Input/Output Devices
- CPU: ALU/Controller/Registers
- System components are connected via buses
Summary (continued)

• Instruction cycle: fetch-decode-execute

• Instructions processed at clock speed

• Basic circuits: Adder/Decoder/Flip-flop/Shifter

• Integrated circuits unite transistors and other components into single chip

• Logical circuit scheme based on Boolean Algebra
Summary (continued)

• Six fundamental circuits (or gates)
  – AND, OR, NOT, NAND, NOR, XOR

• All circuits can be equivalently represented by truth tables and Boolean expressions