CS 315: Computer Logic and Digital Design

- An Overview of Computer Organization
- Switches and Transistors
- Boolean Algebra and Logic
- Binary Arithmetic and Number Systems
- Combinational Logic and Circuits
- Sequential Logic and Circuits
- Memory Logic Design
- The DataPath Unit
An Overview of Computer Organization

Chapter 1
Some Definitions

- *Computer architecture* deals with the functional behavior of a computer system as viewed by a programmer (like the size of a data type – 32 bits to an integer).

- *Computer organization* deals with structural relationships that are not visible to the programmer (like clock frequency or the Type of the physical memory).

- There is a concept of *levels* in computer architecture. The basic idea is that there are many levels at which a computer can be considered, from the highest level, where the user is running programs, to the lowest level, consisting of transistors and wires.
Structure & Function

- Structure is the way in which components relate to each other
- Function is the operation of individual components as part of the structure
Function

• All computer functions are:
  – Data processing
  – Data storage
  – Data movement
  – Control
Functional view

- Functional view of a computer

![Diagram showing the functional view of a computer with nodes labeled Data Movement Apparatus, Control Mechanism, Data Storage Facility, and Data Processing Facility connected by arrows.]
Operations (1)

• Data movement
  – e.g. keyboard to screen
Operations (2)

- Storage
  - e.g. Internet download to disk
Operation (3)

- Processing from/to storage
  - e.g. updating bank statement
Operation (4)

- Processing from storage to I/O
  - e.g. printing a bank statement
Structure - Top Level

- Computer
- Main Memory
- Input Output
- Systems Interconnection
- Communication lines
- Peripherals
Structure - The CPU

Structure:
- Computer
  - I/O
  - System Bus
  - Memory
  - CPU

CPU:
- Registers
- Arithmetic and Login Unit
- Internal CPU Interconnection
- Control Unit
Structure - The Control Unit

- CPU
- ALU
- Internal Bus
- Registers
- Control Unit
- Sequencing Login
- Control Unit Registers and Decoders
- Control Memory
The Von Neumann Model

- The von Neumann model consists of five major components: (1) input unit; (2) output unit; (3) arithmetic logic unit; (4) memory unit; (5) control unit.
The System Bus Model

- A refinement of the von Neumann model, the system bus model has a CPU (ALU and control), memory, and an input/output unit.
- Communication among components is handled by a shared pathway called the system bus, which is made up of the data bus, the address bus, and the control bus. There is also a power bus, and some architectures may also have a separate I/O bus.
The Level of Machines

- There are a number of levels in a computer (the exact number is open to debate), from the user level down to the transistor level.
- Progressing from the top level downward, the levels become less abstract as more of the internal structure of the computer becomes visible.

<table>
<thead>
<tr>
<th>High Level</th>
<th>User Level: Application Programs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>High Level Languages</td>
</tr>
<tr>
<td></td>
<td>Assembly Language / Machine Code</td>
</tr>
<tr>
<td></td>
<td>Microprogrammed / Hardwired Control</td>
</tr>
<tr>
<td></td>
<td>Functional Units (Memory, ALU, etc.)</td>
</tr>
<tr>
<td></td>
<td>Logic Gates</td>
</tr>
<tr>
<td>Low Level</td>
<td>Transistors and Wires</td>
</tr>
</tbody>
</table>
The Mother Board

- The five von Neumann components are visible in this example motherboard, in the context of the system bus model.

System
NAND Gate Example

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Truth Table

Logic Gate Representation

Transistor (Switch) Implementation
System
Hardware Level Description

SYSTEM LEVEL:  
- Processors, memories, peripherals  
- Words, files, records, programs  
- HDL, natural language

REGISTER-TRANSFER LEVEL:  
- Registers, ALUs, buses, CCUs  
- Bytes, words, double words  
- Block diagrams, state diagrams

LOGIC LEVEL:  
- Gates, flip-flops  
- 1, 0, X (unknown); Strong, weak, Z  
- Logic diagrams, boolean equations

CIRCUIT LEVEL:  
- R, C, L, Diodes, Transistors  
- Voltage, current, temperature  
- Schematic diagrams, circuit equations

SILICON LEVEL:  
Elements:  
- nPN & PNP transistors, CMOS  
Values:  
- Voltage, current, temp., fields  
Description:  
- Device models, interconnects
System
Levels of Integration

- **Small scale integration (SSI)**
  - ~10 transistors
  - Individual gates, flip-flops
- **Medium scale integration (MSI)**
  - 10-100 transistors
  - Adders, encoders/decoders, multiplexers, shift registers, counters
- **Large scale integration (LSI)**
  - 100-10,000 transistors
  - small memories, ROMs, PLAs, small memories
- **Very-large scale integration (VLSI)**
  - > 10,000 transistors
  - microprocessors, DSP chips, large memories
Analog vs. Digital
A-To-D Conversion
Analog vs. Digital

Binary Signals
Binary

Binary Logic Levels

TYPICAL VALUES

HIGH (1)

FORBIDDEN ZONE

LOW (0)

V_{Hmax}  \quad 3.3 \text{ V} \quad 5.5 \text{ V}

V_{Hmin}  \quad 2.0 \text{ V} \quad 4.0 \text{ V}

V_{Lmax}  \quad 0.8 \text{ V} \quad 1.0 \text{ V}

V_{Lmin}  \quad 0.0 \text{ V} \quad -0.5 \text{ V}

\{ \text{noise margin} \}

\{ \text{noise margin} \}
Binary Numbers

- Binary numbers are base 2 as opposed to base 10 typically used.
- Instead of decimal places such as 1s, 10s, 100s, 1000s, etc., binary uses powers of two to have 1s, 2s, 4s, 8s, 16s, 32s, 64s, etc. places.

Examples:

\[101_2 = (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = 4_{10} + 1_{10} = 5_{10}\]

\[10111_2 = (1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) = 23_{10}\]

\[00101111_2 = (1 \times 2^5) + (1 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) = 47_{10}\]

- We will discuss binary numbers and binary arithmetic in a little more depth later.
### TABLE 1-2
Numbers with Different Bases

<table>
<thead>
<tr>
<th>Decimal (base 10)</th>
<th>Binary (base 2)</th>
<th>Octal (base 8)</th>
<th>Hexadecimal (base 16)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0000</td>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>0001</td>
<td>01</td>
<td>1</td>
</tr>
<tr>
<td>02</td>
<td>0010</td>
<td>02</td>
<td>2</td>
</tr>
<tr>
<td>03</td>
<td>0011</td>
<td>03</td>
<td>3</td>
</tr>
<tr>
<td>04</td>
<td>0100</td>
<td>04</td>
<td>4</td>
</tr>
<tr>
<td>05</td>
<td>0101</td>
<td>05</td>
<td>5</td>
</tr>
<tr>
<td>06</td>
<td>0110</td>
<td>06</td>
<td>6</td>
</tr>
<tr>
<td>07</td>
<td>0111</td>
<td>07</td>
<td>7</td>
</tr>
<tr>
<td>08</td>
<td>1000</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>09</td>
<td>1001</td>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>1010</td>
<td>12</td>
<td>A</td>
</tr>
<tr>
<td>11</td>
<td>1011</td>
<td>13</td>
<td>B</td>
</tr>
<tr>
<td>12</td>
<td>1100</td>
<td>14</td>
<td>C</td>
</tr>
<tr>
<td>13</td>
<td>1101</td>
<td>15</td>
<td>D</td>
</tr>
<tr>
<td>14</td>
<td>1110</td>
<td>16</td>
<td>E</td>
</tr>
<tr>
<td>15</td>
<td>1111</td>
<td>17</td>
<td>F</td>
</tr>
</tbody>
</table>
Binary-Coded Decimal (BCD)

<table>
<thead>
<tr>
<th>Decimal Symbol</th>
<th>BCD Digit</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000</td>
</tr>
<tr>
<td>1</td>
<td>0001</td>
</tr>
<tr>
<td>2</td>
<td>0010</td>
</tr>
<tr>
<td>3</td>
<td>0011</td>
</tr>
<tr>
<td>4</td>
<td>0100</td>
</tr>
<tr>
<td>5</td>
<td>0101</td>
</tr>
<tr>
<td>6</td>
<td>0110</td>
</tr>
<tr>
<td>7</td>
<td>0111</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
</tr>
<tr>
<td>9</td>
<td>1001</td>
</tr>
</tbody>
</table>

Table 1-3 Binary-Coded Decimal (BCD)
### Table 1-4: American Standard Code for Information Interchange (ASCII)

<table>
<thead>
<tr>
<th>B.B.B.B.</th>
<th>0000</th>
<th>0010</th>
<th>0100</th>
<th>0110</th>
<th>1000</th>
<th>1010</th>
<th>1100</th>
<th>1110</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>NULL</td>
<td>DLE</td>
<td>SP</td>
<td>@</td>
<td>P</td>
<td>~</td>
<td>p</td>
<td></td>
</tr>
<tr>
<td>0010</td>
<td>SOH</td>
<td>DC1</td>
<td>!</td>
<td>1</td>
<td>A</td>
<td>Q</td>
<td>a</td>
<td>q</td>
</tr>
<tr>
<td>0100</td>
<td>STX</td>
<td>DC2</td>
<td>*</td>
<td>2</td>
<td>B</td>
<td>R</td>
<td>b</td>
<td>r</td>
</tr>
<tr>
<td>0110</td>
<td>ETX</td>
<td>DC3</td>
<td>#</td>
<td>3</td>
<td>C</td>
<td>S</td>
<td>c</td>
<td>s</td>
</tr>
<tr>
<td>1000</td>
<td>EOT</td>
<td>DC4</td>
<td>$</td>
<td>4</td>
<td>D</td>
<td>T</td>
<td>d</td>
<td>t</td>
</tr>
<tr>
<td>1010</td>
<td>ENQ</td>
<td>NAK</td>
<td>%</td>
<td>5</td>
<td>E</td>
<td>U</td>
<td>e</td>
<td>u</td>
</tr>
<tr>
<td>1100</td>
<td>ACK</td>
<td>SYN</td>
<td>&amp;</td>
<td>6</td>
<td>F</td>
<td>V</td>
<td>f</td>
<td>v</td>
</tr>
<tr>
<td>1110</td>
<td>BEL</td>
<td>ETB</td>
<td>'</td>
<td>7</td>
<td>G</td>
<td>W</td>
<td>g</td>
<td>w</td>
</tr>
<tr>
<td>1111</td>
<td>BS</td>
<td>CAN</td>
<td>(</td>
<td>8</td>
<td>H</td>
<td>X</td>
<td>h</td>
<td>x</td>
</tr>
<tr>
<td>1011</td>
<td>HT</td>
<td>EM</td>
<td>)</td>
<td>9</td>
<td>I</td>
<td>Y</td>
<td>i</td>
<td>y</td>
</tr>
<tr>
<td>1010</td>
<td>LF</td>
<td>SUB</td>
<td>:</td>
<td>J</td>
<td>Z</td>
<td>j</td>
<td>z</td>
<td></td>
</tr>
<tr>
<td>1011</td>
<td>VT</td>
<td>ESC</td>
<td>+</td>
<td>K</td>
<td>[</td>
<td>k</td>
<td>[</td>
<td></td>
</tr>
<tr>
<td>1100</td>
<td>FF</td>
<td>FS</td>
<td>&lt;</td>
<td>L</td>
<td>\</td>
<td>I</td>
<td>l</td>
<td></td>
</tr>
<tr>
<td>1101</td>
<td>CR</td>
<td>GS</td>
<td>=</td>
<td>M</td>
<td>]</td>
<td>m</td>
<td>]</td>
<td></td>
</tr>
<tr>
<td>1110</td>
<td>SO</td>
<td>RS</td>
<td>&gt;</td>
<td>N</td>
<td>^</td>
<td>n</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>1111</td>
<td>SI</td>
<td>US</td>
<td>/</td>
<td>O</td>
<td>_</td>
<td>o</td>
<td>DEL</td>
<td></td>
</tr>
</tbody>
</table>

**Control Characters:**

- **NULL**: NULL
- **SOH**: Start of heading
- **STX**: Start of text
- **ETX**: End of text
- **EOT**: End of transmission
- **ENQ**: Enquiry
- **ACK**: Acknowledge
- **BEL**: Bell
- **BS**: Backspace
- **HT**: Horizontal tab
- **LF**: Line feed
- **VT**: Vertical tab
- **FF**: Form feed
- **CR**: Carriage return
- **SO**: Shift out
- **SI**: Shift in
- **SP**: Space
- **DLE**: Data link escape
- **DC1**: Device control 1
- **DC2**: Device control 2
- **DC3**: Device control 3
- **DC4**: Device control 4
- **NAK**: Negative acknowledge
- **SYN**: Synchronous idle
- **ETB**: End of transmission block
- **CAN**: Cancel
- **EM**: End of medium
- **SUB**: Substitute
- **ESC**: Escape
- **FS**: File separator
- **GS**: Group separator
- **RS**: Record separator
- **US**: Unit separator
- **DEL**: Delete