Instructor: Roman Lysecky, rlysecky@ece.arizona.edu
Office Hours: MW 1:00-2:00 PM, ECE 356F
Lecture: MWF 12:00-12:50 PM, ILC 150
Course Website: http://www.ece.arizona.edu/~ece274/

**ECE 274 - Digital Logic**

*Textbook - Required*

**Digital Design**

Author: Frank Vahid
ISBN: 978-0-470-04437-7
Website: http://www.cs.ucr.edu/~vahid/dd/

*Textbook - Optional*

**Verilog for Digital Design**

Author: Frank Vahid, Roman Lysecky
ISBN: 978-0-470-05262-4

**ECE 274 - Digital Logic**

*Labs*

TAs: Haiyong Zhang, hzhang@email.arizona.edu
Lance Saldanha, lance@email.arizona.edu
Lab Sections:
- Section 1: M 2:00PM-4:50PM, ECE 301, TA: TBD
- Section 3: T 2:00PM-4:50PM, ECE 301, TA: TBD
- Section 5: T 11:00AM-1:50PM, ECE 301, TA: TBD
- Section 6: R 11:00AM-1:50PM, ECE 301, TA: TBD
- Section 7: R 2:00PM-4:50PM, ECE 301, TA: TBD

**ECE 274 - Digital Logic**

*Syllabus - Grading*

- Course Breakdown:
  - Final: 20%
  - Midterms (2): 40%
  - Quizzes (3): 15%
  - Homework: 5%
  - Lab Assignments: 20%
- Grading:
  - 90 – 100%: A
  - 80 – 90%: B
  - 70 – 80%: C
  - 60 – 70%: D
  - Below 60%: E
- All grades are assigned on an individual basis.

**ECE 274 - Digital Logic**

*Syllabus - Grading*

- Punctuality:
  - Don’t be late!
- Cell Phones:
  - Please turn your cell phone off before coming to class!
- Academic Dishonesty:
  - Any academic dishonesty will not be tolerated, please consult the UA Code of Academic Integrity.
  - ALL course work should be completed entirely on your own
  - You are allowed to discuss general concepts and ideas
  - But you should not discuss specific homework or lab assignments
Why Study Digital Design?

- Look “under the hood” of computers
  - Solid understanding → confidence, insight, even better programmer when aware of hardware resource issues
- Electronic devices becoming digital
  - Enabled by shrinking and more capable chips
  - Enables:
    - Better devices: Better sound recorders, cameras, cars, cell phones, medical devices,...
    - New devices: Video games, PDAs, ...
  - Known as “embedded systems”
    - Thousands of new devices every year
    - Designers needed: Potential career direction

- Predicted that number of transistors per chip would grow exponentially (double every 18 months).

What Does “Digital” Mean?

- Analog signal
  - Infinite possible values
  - Ex: voltage on a wire created by microphone
- Digital signal
  - Finite possible values
  - Ex: button pressed on a keypad
Digital Signals with Only Two Values: Binary

- Binary digital signal -- only two possible values
  - Typically represented as 0 and 1
  - One binary digit is a bit
  - We'll only consider binary digital signals
- Binary is popular because
  - Transistors, the basic digital electric component, operate using two voltages (more in Chpt. 2)
  - Storing/transmitting one of two values is easier than three or more (e.g., loud beep or quiet beep, reflection or no reflection)

Digitized Audio: Compression Benefit

- Digitized audio can be compressed
  - e.g., MP3s
  - A CD can hold about 20 songs uncompressed, but about 200 songs compressed
- Compression also done on digitized pictures (jpeg), movies (mpeg), and more
- Digitization has many other benefits too

How Do We Encode Data as Binary for Our Digital System?

- Some inputs inherently binary
  - Button: not pressed (0), pressed (1)
- Some inputs inherently digital
  - Just need encoding in binary
    - e.g., multi-button input, encode red=001, blue=010, ...
  - Some inputs analog
    - Need analog-to-digital conversion
    - As done in earlier slide -- sample and encode with bits

How to Encode Text: ASCII, Unicode

- ASCII: 7- (or 8-) bit encoding of each letter, number, or symbol
- Unicode: Increasingly popular 16-bit bit encoding
  - Encodes characters from various world languages

How to Encode Numbers: Binary Numbers

- Each position represents a quantity; symbol in position means how many of that quantity
  - Base ten (decimal)
    - Ten symbols: 0, 1, 2, ..., 8, and 9
    - More than 9 -- next position
      - So each position power of 10
    - Nothing special about base 10 -- used because we have 10 fingers
  - Base two (binary)
    - Two symbols: 0 and 1
    - More than 1 -- next position
      - So each position power of 2

Example of Digitization Benefit

- Analog signal (e.g., audio) may lose quality
  - Voltage levels not saved/copy/transmitted perfectly
- Digitized version enables near-perfect save/copy/trans.
  - “Sample” voltage at particular rate, save sample using bit encoding
  - Voltage levels still not kept perfectly
  - But we can distinguish 0s from 1s

Example of Compression Scheme

```
0 00 00 10000001111
0000000000 000000000 0000000000 000000000
0011111111 111111111
00 00 10000001111 01
```

How fix -- higher, lower,?

Can fix -- easily distinguish 0s and 1s better

Question:

What does this ASCII bit sequence represent?

```
10100011 1001011 11010100
```

Note: small red “a” (a) in a slide indicates animation
How to Encode Numbers: Binary Numbers

• Working with binary numbers
  – In base ten, helps to know powers of 10
    • one, ten, hundred, thousand, ten thousand, ...
  – In base two, helps to know powers of 2
    • one, two, four, eight, sixteen, thirty two, sixty four, one hundred twenty eight
  – (Note: unlike base ten, we don’t have common names, like “thousand,” for each position in base ten — so we use the base ten name)

• Q: count up by powers of two
  2^4 2^3 2^2 2^1 2^0
  16 8 4 2 1

Converting from Decimal to Binary Numbers: Subtraction Method (Easy for Humans)

• Goal
  – Get the binary weights to add up to the decimal quantity
  • Work from left to right
  • (Right to left – may fill in 1s that shouldn’t have been there — try it).

Desired decimal number: 12

<table>
<thead>
<tr>
<th>Binary Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
</tr>
</tbody>
</table>

Remaining quantity: 12

<table>
<thead>
<tr>
<th>Binary Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
</tr>
</tbody>
</table>

Converting from Decimal to Binary Numbers: Division Method (Good for Computers)

• Divide decimal number by 2 and insert remainder into new binary number.
  – Continue dividing quotient by 2 until the quotient is 0.

Example: Convert decimal number 12 to binary

<table>
<thead>
<tr>
<th>Decimal Number</th>
<th>Binary Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>0</td>
</tr>
</tbody>
</table>

2 divide by 2

<table>
<thead>
<tr>
<th>Decimal Number</th>
<th>Binary Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

4 divide by 2

<table>
<thead>
<tr>
<th>Decimal Number</th>
<th>Binary Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

8 divide by 2

<table>
<thead>
<tr>
<th>Decimal Number</th>
<th>Binary Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

6 divide by 2

<table>
<thead>
<tr>
<th>Decimal Number</th>
<th>Binary Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

3 divide by 2

<table>
<thead>
<tr>
<th>Decimal Number</th>
<th>Binary Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

0 divide by 2

<table>
<thead>
<tr>
<th>Decimal Number</th>
<th>Binary Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Done! 25 in decimal is 10111 in binary.
Converting from Decimal to Binary Numbers: Division Method (Good for Computers)

- Example: Convert decimal number 12 to binary (continued)

<table>
<thead>
<tr>
<th>Decimal Number</th>
<th>Binary Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>1100</td>
</tr>
</tbody>
</table>

- Continue dividing since quotient (1) is greater than 0

<table>
<thead>
<tr>
<th>Divide by 2</th>
<th>Insert remainder</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Since quotient is 0, we can conclude that 12 is 1100 in binary.

Encoding: Number Systems

Base 2 (binary) Arithmetic

- Convert the following decimal value to a binary (zero’s and one’s) value

5410 = ???????2

Choose your answer:
A) 110110
B) 100010
C) 1000010

Base Sixteen: Another Base Sometimes Used by Digital Designers

- Nice because each position represents four base two positions
  - Used as compact means to write binary numbers

- Known as hexadecimal, or just hex

<table>
<thead>
<tr>
<th>hex</th>
<th>binary</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>
</tbody>
</table>

Q: Write 11110000 in hex

Digital Design

Encoding: Number Systems: System Conversion

- Convert the following hexadecimal value to a binary (zero’s and one’s) value

CAB16 = ????????????2

Choose your answer:
A) 110111101010
B) 110001011001
C) 110010101011

Implementing Digital Systems: Programming Microprocessors

What is the highest value you can count to using your 5-fingers? 10-fingers?
Digital Design: When Microprocessors Aren’t Good Enough

- With microprocessors so easy, cheap, and available, why design a digital circuit?
  - Microprocessor may be too slow
  - Or too big, power hungry, or costly

Sample digital camera task execution times (in seconds) on a microprocessor versus a digital circuit:

<table>
<thead>
<tr>
<th>Task</th>
<th>Microprocessor</th>
<th>Custom Digital Circuit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>5.8 + 1</td>
<td>1.4 sec</td>
</tr>
<tr>
<td>Compress</td>
<td>0.1 + 0.5</td>
<td>1.4 sec</td>
</tr>
<tr>
<td>Store</td>
<td>0.8</td>
<td>Good compromise</td>
</tr>
</tbody>
</table>

Q: How long for each implementation option?

- Inside computers
- Inside huge variety of other electronic devices (embedded systems)

- Digital systems use 0s and 1s
  - Encoding analog signals to digital can provide many benefits
    - e.g., audio -- higher-quality storage/transmission, compression, etc.
  - Encoding integers as 0s and 1s: Binary numbers

- Microprocessors (themselves digital) can implement many digital systems easily and inexpensively
  - But often not good enough — need custom digital circuits

Digital Design Where do we go from here?

- ECE 274 Course Goals:
  - Combinational Logic Design
  - Sequential Logic Design
  - Design of Common Data Components
  - Register-Transfer Level (RTL) Design
    - Modern approach to Digital Design
  - Optimization of Digital Circuits
  - Digital Design using HDL (Verilog)

Chapter Summary

- Digital systems surround us
  - Inside computers
  - Inside huge variety of other electronic devices (embedded systems)

- Digital systems use 0s and 1s
  - Encoding analog signals to digital can provide many benefits
    - e.g., audio -- higher-quality storage/transmission, compression, etc.
  - Encoding integers as 0s and 1s: Binary numbers

- Microprocessors (themselves digital) can implement many digital systems easily and inexpensively
  - But often not good enough — need custom digital circuits

Digital Design (Humor)

There are 10 types of people in the world:
Those who get binary and those who don’t.