ECE 274 Digital Logic

**Instructor:** Roman Lysecky, rlysecky@ece.arizona.edu

**Office Hours:** MW 1:00-2:00 PM, ECE 320F

**Lecture:** MWF 12:00-12:50 PM, ILC 140

**Course Website:** [http://www.ece.arizona.edu/~ece274/](http://www.ece.arizona.edu/~ece274/)

---

**ECE 274 - Digital Logic (Labs)**

**TAs:**
- Haiyong Zhang, hzhang@email.arizona.edu
- Julian Sosa, jsosamol@email.arizona.edu
- Annapoorna Krishnaswamy, annakris@email.arizona.edu

**Lab Sections:**
- Section 1: M 2:00PM-4:50PM, ECE 301, TA: Annapoorna Krishnaswamy
- Section 2: T 8:00AM-10:50AM, ECE 301, TA: Julian Sosa
- Section 3: T 11:00AM-1:50PM, ECE 301, TA: Haiyong Zhang
- Section 4: W 2:00PM-4:50PM, ECE 301, TA: Annapoorna Krishnaswamy
- Section 5: T 11:00AM-1:50PM, ECE 301, TA: Haiyong Zhang
- Section 6: R 2:00PM-4:50PM, ECE 301, TA: Haiyong Zhang
- Section 7: R 2:00PM-4:50PM, ECE 301, TA: Julian Sosa
- Section 8: F 2:00PM-4:50PM, ECE 301, TA: Annapoorna Krishnaswamy

---

**ECE 274 - Digital Logic (Textbook)**

**Fundamentals of Digital Logic with Verilog Design**

**Authors:** Stephen Brown and Zvonko Vranesic

**ISBN:** 0072838787

**Website:** [http://highered.mcgraw-hill.com/sites/0072838787/](http://highered.mcgraw-hill.com/sites/0072838787/)

---

**ECE 274 - Digital Logic (Optional Textbook)**

**Digital Design**

**Author:** Frank Vahid

**ISBN:** 0471467847

**Website:** [http://www.cs.ucr.edu/~vahid/dd/](http://www.cs.ucr.edu/~vahid/dd/)

*Highly Recommended*

---

**ECE 274 - Digital Logic (Syllabus)**

**Course Breakdown:**
- Final 25%
- Midterms 40%
- Quizzes 5%
- Homework 10%
- Lab Assignments 20%

**Grading:**
- 90 – 100% A
- 80 – 90% B
- 70 – 80% C
- 60 – 70% D
- Below 60% F

*All grades are assigned on an individual basis.*
ECE 274 - Digital Logic (Course Policies)

- 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 homework or lab assignments.

Reading:
- Be prepared, read over material BEFORE class.

Regrades:
- All requests for regrades must be submitted in writing within one week of the distribution of graded material.

Digital Design
Silicon Wafer

Individual IC (die)

Digital Design
Pentium Processor (Die Photo)

Digital Design
Field-Programmable Gate Array (Spartan3 Die Photo)

Digital Design
Moore’s Law
- Predicted that number of transistors per chip would grow exponentially (double every 18 months).
- Exponential improvement in technology is a natural trend: steam engines, dynamos, automobiles.
What of the following is the largest (in terms of number of transistors)?

A) Pentium 4 Extreme Edition  
B) Xilinx FPGA  
C) Geforce 6800 Ultra

Answer:  
A) Pentium 4 Extreme Edition (178 million)  
B) Xilinx FPGA (1 billion)  
C) Geforce 6800 Ultra (222 million)
Digital Design

Analog & Digital Signals (converting analog to digital)

For this Keypad:
Inputs?
- User presses one button
Outputs?
- Encoding for that button

Digital Design

Encodings

Sample ASCII encodings

Encoding: Number Systems

Base 10 (decimal) Number System

Uses the ten numbers from 0 to 9
Each column represents a power of 10

Encoding: Number Systems

Base 10 (decimal) Arithmetic

5 2 3

10^4 10^3 10^2 10^1 10^0

1999_{10} = 1 \times 10^3 + 9 \times 10^2 + 9 \times 10^1 + 9 \times 10^0
Encoding: Number Systems

Counting correctly in base 10

| 0 to 9 | As usual: “zero,” “one,” “two,” etc. |
| 10 to 99 | 10, 11, 12, ... 99: “ten,” “ten one,” “ten ten,” “ten ten nine” |
| 100 to 999 | 20, 21, 22, ... 99: “two ten,” “two ten one,” “two ten two,” “two ten nine” |
| 100 to 999 | 30, 40, ... 90: “three ten,” “four ten,” “five ten,” “six ten,” “seven ten,” “eight ten,” “nine ten” |
| 1000 and up | As usual |

Encoding: Number Systems

Base 2 (binary) Arithmetic

- Uses the two numbers from 0 to 1
- Every column represents a power of 2

1001₂ = \(1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0\)

Encoding: Number Systems

Base 2 (binary) Number System

<table>
<thead>
<tr>
<th>1</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2⁴</td>
<td>2³</td>
<td>2²</td>
</tr>
</tbody>
</table>

Encoding: Number Systems

Positional Number Systems

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

\[1001₁₀₂ = ? \text{ in Decimal}\]

Choose your answer:
A) 100,110
B) 22
C) 38

Encoding: Number Systems

Positional Number Systems

What is the highest value you can count to using your 5-fingers? 10-fingers?
Convert the following hexadecimal value to a binary (zero’s and one’s) value

CAB_{16} = ? in Binary

Choose your answer:
A) 110111101010
B) 110001011001
C) 110010101011
**Digital Design**

**Microprocessor: The Digital Workhorse**

- [Image of microprocessor components]

**Digital Design**

**Microprocessor: Software Implementation**

**Circuit Description:** using a microprocessor

**Inputs:** 2 sensors (one light, one motion)

**Outputs:** 1 signal to lamp

**Functional Description:** Lamp illuminated when it is dark and motion is detected.

```c
void main()
{
    while(1)
    {
        P0 = I0 && !I1;
    }
}
```

**Digital Design**

**Microprocessor: Timing Diagrams (Motion in Dark Detector System)**

- [Diagram showing timing and sensor inputs and outputs]

**Circuit Description:** using a microprocessor

**Inputs:** 3 motion sensors

**Outputs:** 1 signal to buzzer

**Functional Description:** System activates a buzzer when any of the three motion sensors is activated.

```c
void main()
{
    while(1)
    {
        P0 = I0 || I1 || I2;
    }
}
```

**Digital Design**

**Microprocessors: Variety of Processors**

- PIC ($1-5$)
- 8051 ($1-10$)
- Pentium ($>100$)
- Pentium (>$100$)

**Digital Design**

**Microprocessors: Trouble in Paradise**

- With microprocessors so readily available, why would anyone ever need to design new digital circuits?

**When analyzing needs for a particular system:**
- Software may be too slow
- May be too much circuitry than needed
- Can be costly for simple circuits
- Power hungry

**Solution?**
Sample digital camera task execution times (in seconds) on a microprocessor versus a digital circuit.

<table>
<thead>
<tr>
<th>Digital Camera Task</th>
<th>Microprocessor</th>
<th>Custom digital circuit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>5</td>
<td>0.1</td>
</tr>
<tr>
<td>Compress</td>
<td>8</td>
<td>0.5</td>
</tr>
<tr>
<td>Store</td>
<td>1</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Digital Circuit: Motion Detector Implementation

Circuit Description: using a microprocessor
Inputs: 2 sensors (one light, one motion)
Outputs: 1 signal to lamp
Functional Description: Lamp illuminated when it is dark and motion is detected.

\[ F = a \& \& \neg b; \]

When analyzing needs for a particular system:
- Custom circuit may be too costly
- A processor might be just as fast, and cheaper

Design Partitioning: Possible Digital Camera Implementations

- Design Partitioning
  - Deciding which tasks to implement on the microprocessor and which to implement as a custom digital circuit
  - Digital camera implemented with:
    - (a) a microprocessor,
    - (b) custom circuits, and
    - (c) a combination of custom circuits and a microprocessor.

Digital Camera implemented with:
(a) a microprocessor,
(b) custom circuits, and
(c) a combination of custom circuits and a microprocessor.

Where do we go from here?

ECE 274 Course Goals:
- Combinational Logic Design
- Sequential Logic Design
- Design of Common Components
- Register-Transfer Level (RTL) Design
  - Modern approach to Digital Design
- Optimization of Digital Circuits
- Digital Design using HDL (Verilog)
There are 10 types of people in the world: Those who get binary and those who don't.