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/

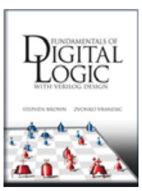
ECE 274 - Digital Logic *(Labs)*

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	2:00PM-4:50PM,	ECE 301,	TA: Julian Sosa
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	11:00AM-1:50PM,	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/0072823151/

ECE 274 - Digital Logic *(Optional Textbook)*

Digital Design
Author: Frank Vahid
ISBN: 0471467847

Website: 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%

ECE 274 - Digital Logic *(Grading)*

- Grading:
  - 90 – 100% A
  - 80 – 90% B
  - 70 – 80% C
  - 60 – 70% D
  - Below 60% F
- All grades are assigned on an individual basis.

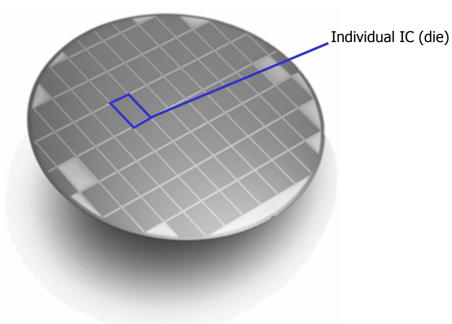
ECE 274 - Digital Logic
(Course Policies)

- Punctuality:
  - Don't be late!
- Cell Phones:
  - Please turn your cell phone off before coming to class!
- Academic Dishonesty:
  - Any academic dishonesty will no 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

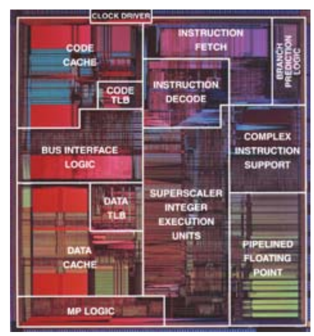
ECE 274 - Digital Logic
(Course Policies)

- 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.

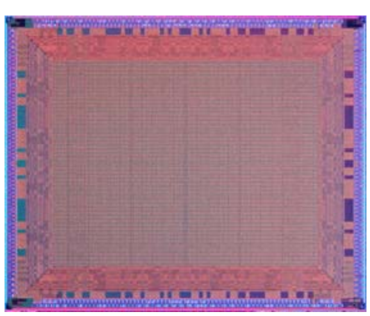
Digital Design
Silicon Wafer



Digital Design
Pentium Processor (Die Photo)



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



Digital Design
Moore's Law

- Gordon Moore: co-founder of Intel.
- 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.

Digital Design
Moore's Law

Courtesy of Intel

Digital Design
Moore's Law

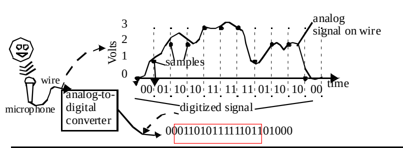
What of the following is the largest (in terms of number of transistors)?

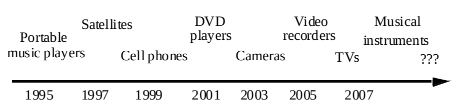
A) Pentium 4 Extreme Edition
B) Xilinx FPGA
C) Geforce 6800 Ultra

Answer:

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

Digital Design
Analog & Digital Signals

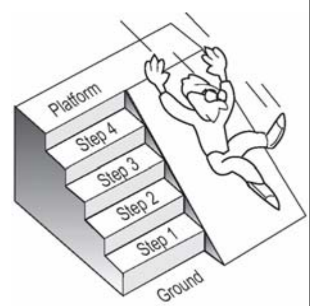




More and more analog products are becoming digital.

Digital Design
Analog vs. Digital

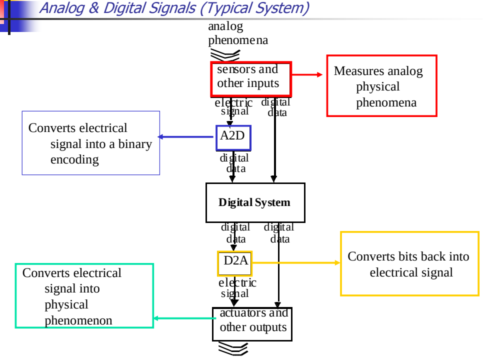
- Analog
  - Continuous
  - E.g., Radio Antenna on Cell Phone
- Digital
  - Discrete
  - E.g., Pentium Processor



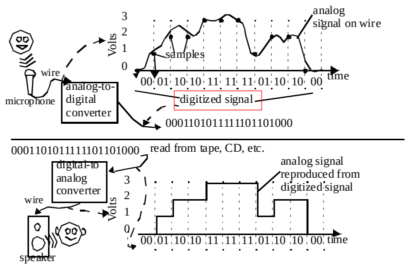
Digital Design
Analog to Digital

Analog Radio data must Translated into a Digital format for the processor to compute the data transmissions

Digital Design
Analog & Digital Signals (Typical System)



Digital Design
Analog & Digital Signals (converting analog to digital)



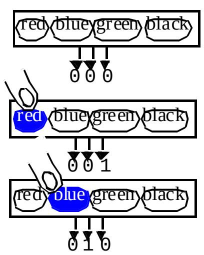
Digital Design
Encodings

For this Keypad:

Inputs?
- User presses one button

Outputs?
- Encoding for that button

Digital Design
Encodings



Digital Design
Encodings

Symbol	Encoding	Symbol	Encoding
R	01010010	r	01110010
S	01010011	s	01110011
T	01010100	t	01110100
L	01001100	l	01101100
N	01001110	n	01101110
E	01000101	e	01100101
O	00110000	9	00111001
.	00101110	!	00100001
<tab>	00001001	<space>	00100000

Sample ASCII encodings

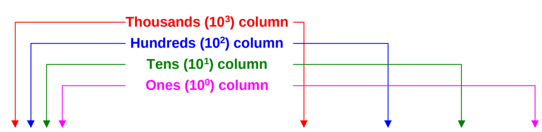
Encoding: Number Systems
Base 10 (decimal) Number System

_ _ 5 2 3

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

Encoding: Number Systems
Base 10 (decimal) Arithmetic

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



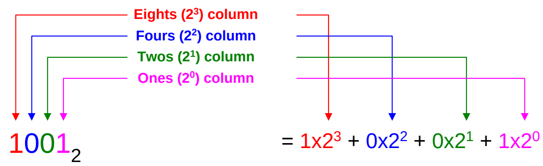
1999_{10} $= 1x10^3 + 9x10^2 + 9x10^1 + 9x10^0$

Encoding: Number Systems
Counting correctly in base 10

0 to 9	As usual: "zero," "one," "two," etc.
10 to 99	10, 11, 12, ... 19: "one ten," "one ten one," "one ten two," ... "one ten nine"
	20, 21, 22, ..., 29: "two ten," "two ten one," "two ten two," ... "two ten nine"
	30, 40, ... 90: "three ten," "four ten," ... "nine ten"
100 to 900	As usual: "one hundred," "two hundred," ... "nine hundred." Even better would be to replace the word "hundred" by "ten to the power of 2."
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***



Encoding: Number Systems
Base 2 (binary) Number System

1 0 1

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

Encoding: Number Systems
Base 2 (binary) Number System

1 0 1

16 8 4 2 1

Encoding: Number Systems
Positional Number Systems

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

100110_2 = ? in Decimal

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

Encoding: Number Systems
Positional Number Systems

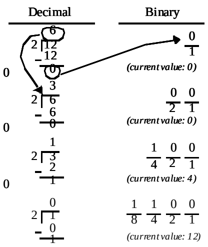
What is the highest value you can count to using your 5-fingers? 10-fingers?

Digital Design

Encoding: Number Systems: System Conversion

- Converting the decimal number 12 to binary using the divide-by-2 method.

1. Divide decimal number by 2
   Add remainder to binary number
   Continue since quotient (6) is greater than 0
2. Divide quotient by 2
   Add remainder to binary number
   Continue since quotient (3) is greater than 0
3. Divide quotient by 2
   Add remainder to binary number
   Continue since quotient (1) is greater than 0
4. Divide quotient by 2
   Add remainder to binary number
   Quotient is 0, done



Digital Design

Encoding: Number Systems: System Conversion

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

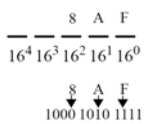
54_{10} = ? in Binary

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

NOTE: Generally, a number can be converted from one base to another by 1) converting the number to base 10, then 2) converting the base ten number to the desired base using the divide-by-*n* method. May not always be the easiest way…

Digital Design

Base 16 (hexadecimal) Number System



hex	binary	hex	binary
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

Digital Design

Hexadecimal Numbers

- Each position can represent 16 values
- Why Mentioned:
  - Used heavily in data-sheet descriptions of circuits
  - After 10 digits, we move to alpha characters: A-F

NOTE:
\# of Characters(0-9) = 10
+# of Characters (A-F) = 6
16 values

Binary	Decimal	Hexa-Decimal
0	0	0
1	1	1
10	2	2
11	3	3
100	4	4
101	5	5
110	6	6
111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F
10000	16	?
10001	17	?

Digital Design

Encoding: Number Systems: System Conversion

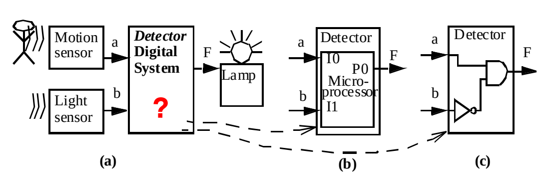
- Convert the following hexadecimal value to a binary (zero's and one's) value

CAB_{16} = ? in Binary

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

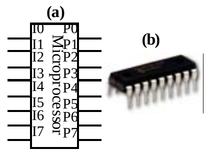
Digital Design

System Implementation Options (Motion in Dark Detector System)



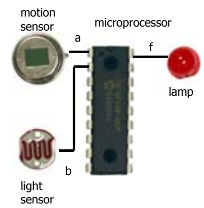
System Block Diagram *Microprocessor Implementation* *Custom Digital Circuit*

Digital Design
Microprocessor: The Digital Workhorse



Digital Design
Microprocessor: Software Implementation

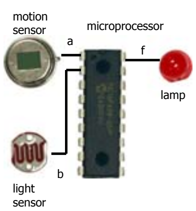
Circuit Description: using a microprocessor
Inputs: 2 sensors (one light, one motion)
Outputs: 1 signal to lamp
Functional Description:
Lamp illuminated when it is dark and motion is detected.

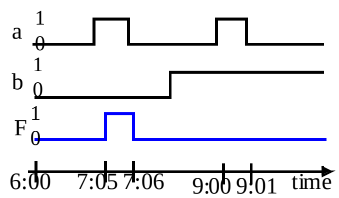



```
void main()
{
  while(1)
  {
   P0 = I0 && !I1;
  }
}
```


Digital Design
Microprocessor: Timing Diagrams (Motion in Dark Detector System)

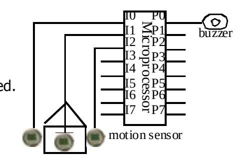
P0 = I0 && !I1;





Digital Design
Microprocessor: Motion Detector: Software Implementation

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.

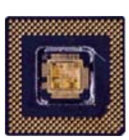



```
void main()
{
  while(1)
  {
   P0 = I0 || I1 || I2;
  }
}
```


Digital Design
Microprocessors: Variety of Processors

PIC ($1-5)

8051 ($1-10)



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?

Digital Design

Process Analysis: Microprocessor vs. Digital Circuit

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

Digital Camera Task	Microprocessor	Custom digital circuit
Read	5	0.1
Compress	8	0.5
Store	1	0.8

Digital Design

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 && !b;

When analyzing needs for a particular system:

- Custom circuit may be too costly
- A processor might be just as fast, and cheaper

Digital Design

Partitioning: Possible Digital Camera Implementations

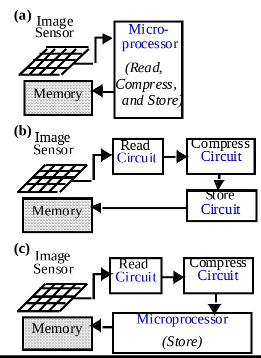
- Design Partitioning
  - Deciding which tasks to implement on the microprocessor and which to implement as a custom digital circuit

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

Digital Design

Partitioning: Possible Digital Camera Implementations

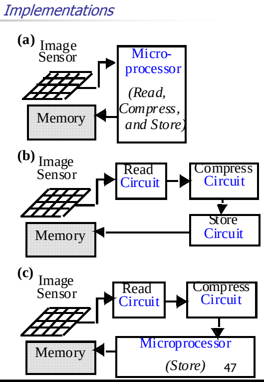
Digital camera implemented with:
(a) a microprocessor,
(b) custom circuits, and
(c) a combination of custom circuits and a microprocessor.



Digital Design

Partitioning: Possible Digital Camera Implementations

Digital Camera Task	μP	Custom digital circuit	Hybrid: μP + Custom Digital
Read	5	0.1	0.1
Compress	8	0.5	0.5
Store	1	0.8	1
Total	14	1.4	1.6



Digital Design

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)

Digital Design
(Humor)

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