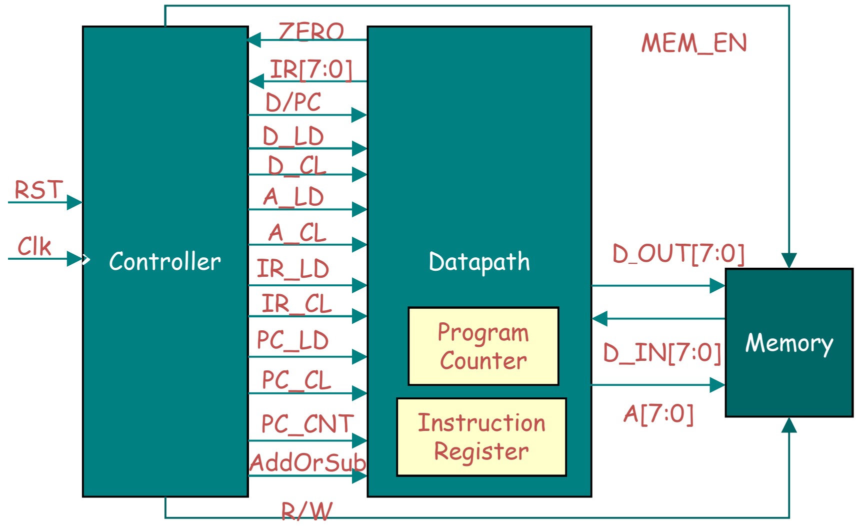
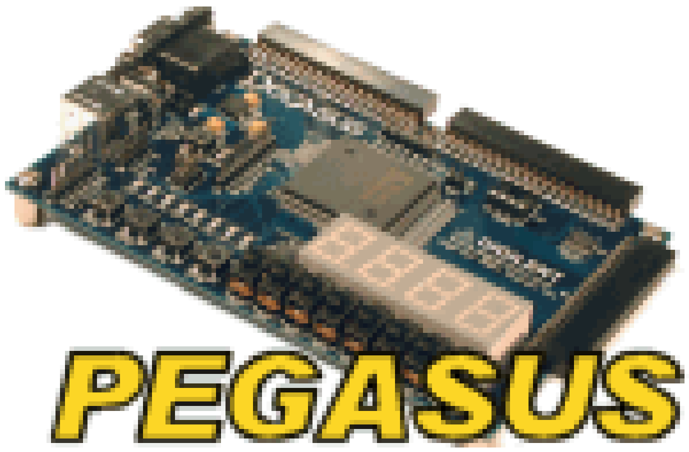
EE/CE/TE 1202 / Spring 2015:

Introduction to Electrical Engineering

Instructor: Dr. Amir Khoobroo

Come and Learn how Computers Compute!!!



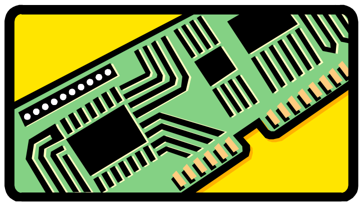


BUILD YOUR OWN TOY PROCESSOR ON AN FPGA!

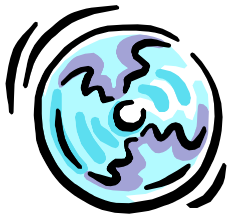
The Computer

Main Unit:

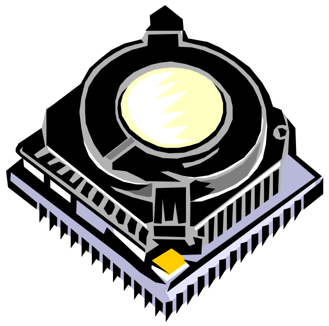
Memory:



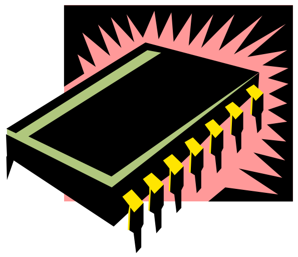
Fan:



CPU:

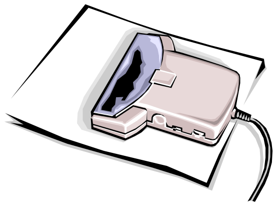


Glue Logic:

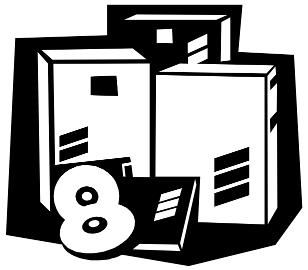


Peripherals:

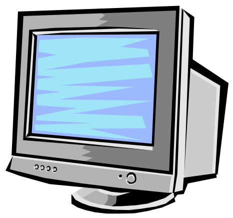
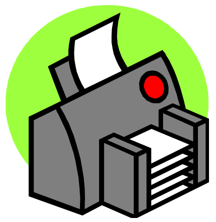
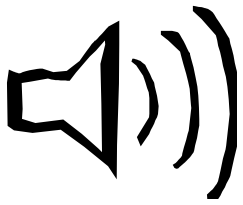
Input:



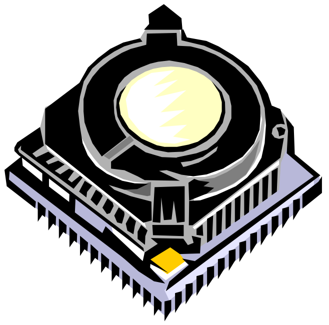
Storage:



Output:

The Central Processing Unit (CPU)

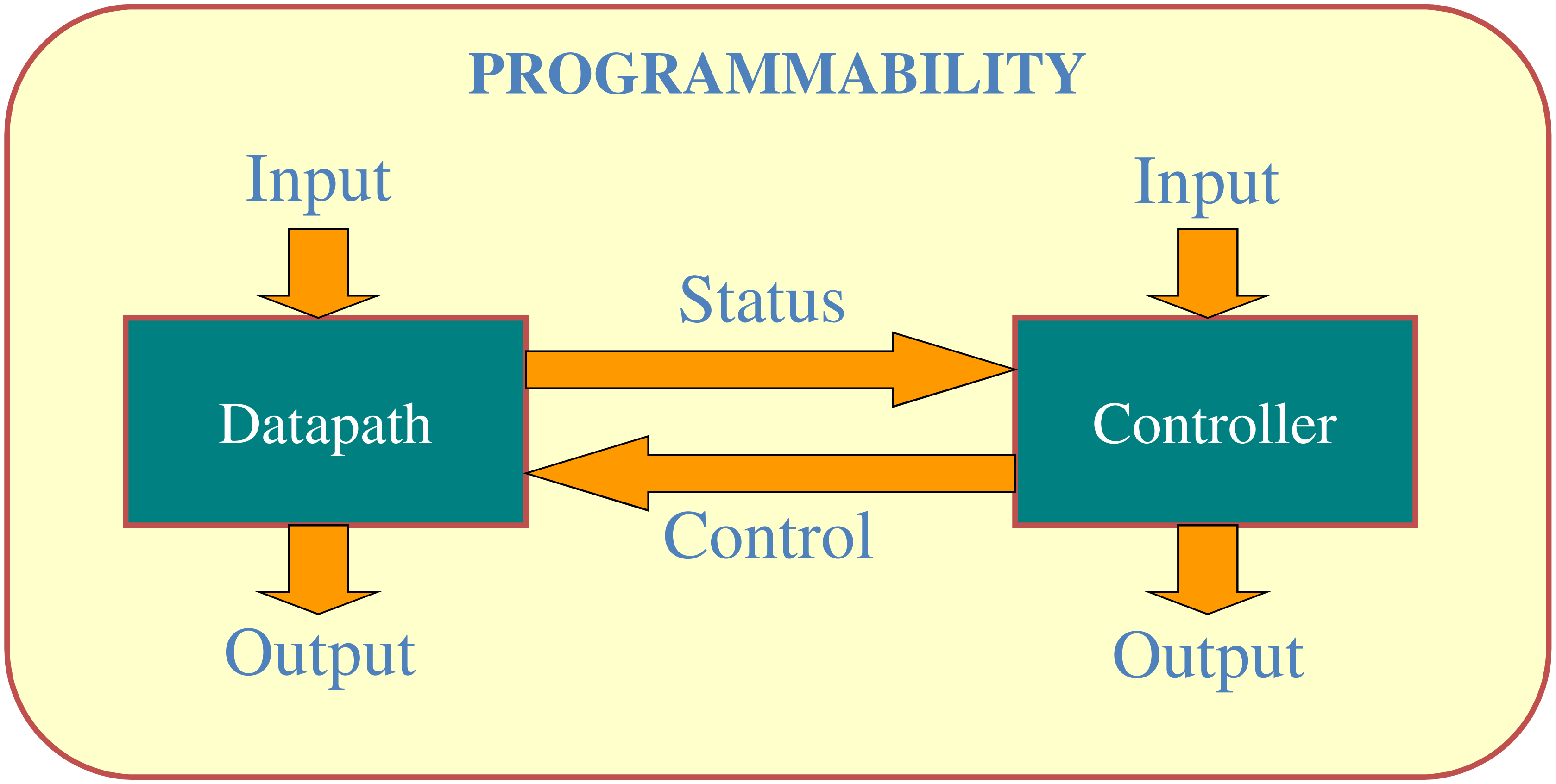


Main Task: Perform Computation

Key Characteristics:

- **DATAPATH:** A collection of fine-grained computational capabilities (load, store, add, subtract, shift, …)
- **CONTROLLER:** A mechanism to arrange and apply these capabilities in a specific sequence
- **PROGRAMMABILITY:** The ability to instruct the datapath/controller pair to perform a desired computation

Datapath/Controller Pair



General Purpose vs. Application Specific

Datapath

Mathematical Abstractions:

Information Representation, Boolean Functions

Basic Building Blocks:

- Logic Gates & Flip-Flops
- Register Transfer Level Components

Controller

Mathematical Abstraction:

Finite State Machines

Basic Building Blocks:

- Logic Gates & Flip-Flops
- State Registers and Next State Components

Programmability

- Instruction Set Design
- Addressing Modes

Putting it All Together…

Laboratory

1) Familiarization with Tools:

• Design of Datapath Circuits via Schematic Capture And Controllers via State Machine Editing

• Simulation of Datapath and Controller Circuits

2) Processor Design & Simulation:

• Design of a Toy Processor with a Minimal Instruction Set

a) Design and Simulation of the Datapath

b) Design and Simulation of the Controller

3) Processor Construction and Implementation:

• Construction and Simulation of the Datapath/Controller

• Implementation on an FPGA and Interfacing to the World

Information Representation

Information:

Numbers, Text, Audio, Video, etc

$\Downarrow$

Represented and Stored as Binary Digits (BITS):

• A Bit takes values from the set {0,1}

• Combinations of Bits are used to represent information

BIT Combinations – Representation Capacity:

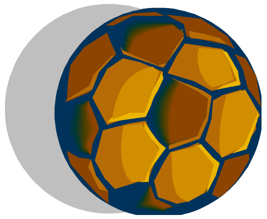
• How many distinct objects can be represented with "n" bits?

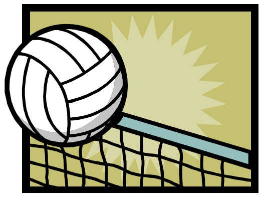
Answer:

• 2^n (Each bit doubles the amount of choices)

Examples

1 bits – 2 Combinations:





0	1

2 bits – 2×2=4 Combinations:

🟦	🟥	🟥	🟧
00	01	10	11

3 bits – 2×2×2=8 Combinations:

C	O	M	P	U	T	E	R
000	001	010	011	100	101	110	111

Number Representation

Positional Number System:

Each number is represented by a string of digits, in which the position of each digit has an associated weight

Example of Decimal Number:

$1234.56 = 1\times1000+2\times100+3\times10+4\times1+5\times0.1+6\times0.01$

General Decimal Number Form:

$$\underbrace{\text{Most Significant Digit (MSD)}}_{} \longrightarrow \mathbf{d_{m-1}} d_{m-2}\ldots d_0 \,.\, d_{-1} d_{-2}\ldots \mathbf{d_{-n}} \longleftarrow \text{Least Significant Digit (LSD)}$$

Radix Point ↑ (under the point)

Corresponding Value:

$$D=d_{m-1}\times10^{m-1}+d_{m-2}\times10^{m-2}+\ldots+d_0\times10^0+d_{-1}\times10^{-1}+\ldots+d_{-n}\times10^{-n}$$
$$=\sum_{i=-n}^{m-1} d_i\times10^i$$

Binary Number System

General Binary Number Form:

Most Significant Bit (MSB) $\rightarrow$ $\mathbf{b_{m-1}} b_{m-2} \ldots b_0 \,.\, b_{-1} b_{-2} \ldots \mathbf{b_{-n}}$ $\leftarrow$ Least Significant Bit (LSB)

$\uparrow$ Radix Point

Corresponding Value:

$$B = b_{m-1} \times 2^{m-1} + b_{m-2} \times 2^{m-2} + \ldots + b_0 \times 2^0 + b_{-1} \times 2^{-1} + \ldots + b_{-n} \times 2^{-n}$$
$$= \sum_{i=-n}^{m-1} b_i \times 2^i$$

Examples of Binary Numbers:

$$10101_2 = 1\times16+0\times8+1\times4+0\times2+1\times1=21_{10}$$
$$10.101_2 = 1\times2+0\times1+1\times0.5+0\times0.25+1\times0.125=2.625_{10}$$

Hexadecimal Number System

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Conversions

Binary to Hexadecimal Conversion:

Starting from the radix point and going leftwards, separate the bits into groups of four and replace each group with the corresponding hexadecimal digit. Repeat from the radix point going rightwards

Example:

$1010011100.10101_2 = 0010\ 1001\ 1100.\ 1010\ 1000 = 29C.A8_{16}$

Hexadecimal to Binary Conversion:

Replace each hexadecimal number with the corresponding 4-bit binary number

Example:

$FED.23_{16} = 1111\ 1110\ 1101\ .\ 0010\ 0011_2$

Decimal to Binary Conversion

General Binary Form:

$$D = ((\dots((b_{m-1}\times 2)+b_{m-2})\times 2+\dots+b_1)\times 2+b_0 = \sum_{i=0}^{m-1} b_i\times 2^i$$

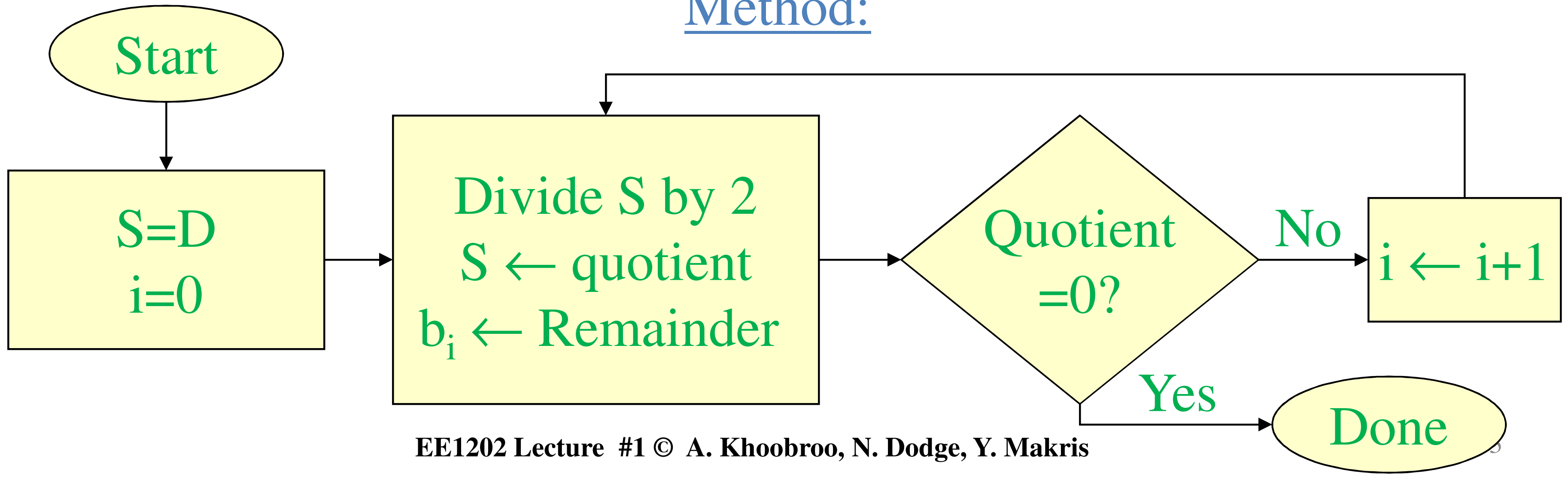
Dividing this equation by the radix 2 we obtain:

Quotient $Q = ((\dots((b_{m-1}\times 2)+b_{m-2})\times 2+\dots+b_2)\times 2+b_1$

and

Remainder $R = b_0$

Method:



Decimal to Binary Conversion

Example:

Convert 179_{10} to binary:

179 / 2 => Quotient = 89 and Remainder = 1 (LSB)
89 / 2 => Quotient = 44 and Remainder = 1
44 / 2 => Quotient = 22 and Remainder = 0
22 / 2 => Quotient = 11 and Remainder = 0
11 / 2 => Quotient = 5 and Remainder = 1
5 / 2 => Quotient = 2 and Remainder = 1
2 / 2 => Quotient = 1 and Remainder = 0
1 / 2 => Quotient = 0 and Remainder = 1 (MSB)

Result From MSB to LSB:

10110011_2

Addition of Binary Numbers

Addition of binary digits

$x_i + y_i + c_i$			c_{i+1}	s_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

		512	256	128	64	32	16	8	4	2	1
x(987)		1	1	1	1	0	1	1	0	1	1
y(123)		0	0	0	1	1	1	1	0	1	1
Carries		1	1	1	1	1	1	0	1	1	
x+y (1110)	1	0	0	0	1	0	1	0	1	1	0
	s_{10}	s_9	s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0

Subtraction of Binary Numbers

Subtraction of binary digits

x_i	- y_i	- b_i	b_{i+1}	d_i
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

		512	256	128	64	32	16	8	4	2	1
x(987)		1	1	1	1	0	1	1	0	1	1
y(123)		0	0	0	1	1	1	1	0	1	1
Borrows		0	0	1	1	0	0	0	0	0	
x-y(864)		1	1	0	1	1	0	0	0	0	0
	d_{10}	d_9	d_8	d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0

Is Engineering Right for You?

- "Toto, I have a feeling we're not in Kansas anymore."
- Now that you are here, did you make the right choice?
- Electrical/Computer engineering is a challenging and satisfying profession. That does not mean it is easy. In fact, with the possible exceptions of medicine or law, it is the MOST difficult.
- There are some things you need to consider if you really, really want to be an engineer.
- Why did you decide to be an electrical/computer engineer?
  - Parents will pay for your education so it's what they want.
  - You like math and science.
  - A relative is an engineer and you like him/her.
  - You want to challenge yourself, and engineering seems challenging.
  - You think you are creative and love technology.
  - You want to make a difference in the society.

The High School “Science Student” Problem

- In high school, you were **far above** the average and you probably didn’t study too hard, right?
- You liked science and math, and they weren’t terribly hard.
- You figured out that the professions that make the “big bucks” are law, medicine, and engineering.
  - But those lawyers have to memorize a bunch of dry facts!
  - And who wants to see all that blood and cut up bodies?
  - Engineers get to make design and build cool stuff.
- Science and engineering are NOT THE SAME.
- Generally, scientists try to discover new facts and add to human knowledge.
- Engineers use scientific knowledge to produce cost-effective technological answers to societal problems.
  - The cost is as important as the solution (sometimes MORE important!).
  - Engineering is not doing something once, but producing something that can be replicated inexpensively and **make profit** for the company.

You Should Be an Engineer If:

- The thought of creating something totally new is very exciting to you (think iPhone!).
- You work and play well with others (NO engineer EVER works alone!).
- Hard work and long hours are fine with you so long as the work is not repetitive or boring (and yes, even the best engineering tasks have boring parts – try negotiating price and delivery with a vendor!).
- You really, really loved trigonometry and you can't wait to study calculus!
- You are patient and enjoy the challenge of different courses, even though you are not sure why they are in the curriculum.
  - English? Wait until that first presentation to your boss or that first report to the VP of engineering!
  - Chemistry? Wait until you get that first job in a wafer fab at Texas instruments!
- Building circuits in the lab is so cool you just cannot wait to get there.
- The idea of contributing to society sounds exciting or fulfilling.
- You like the idea of having responsibility and taking a major role in an important project.