CpE 213
Digital Systems Design
Review

Lecture 2

Wednesday 8/24/2005

UNIVERSITY OF MISSOURI-ROLLA
The Name. The Degree. The Difference.

Overview

- Announcement
- Introduction (continued)
- Review of number systems

Announcement

- Class will be in EECH 104 from now on.
- This is a permanent change.
- Exam locations will be announced in advance.

Basic information

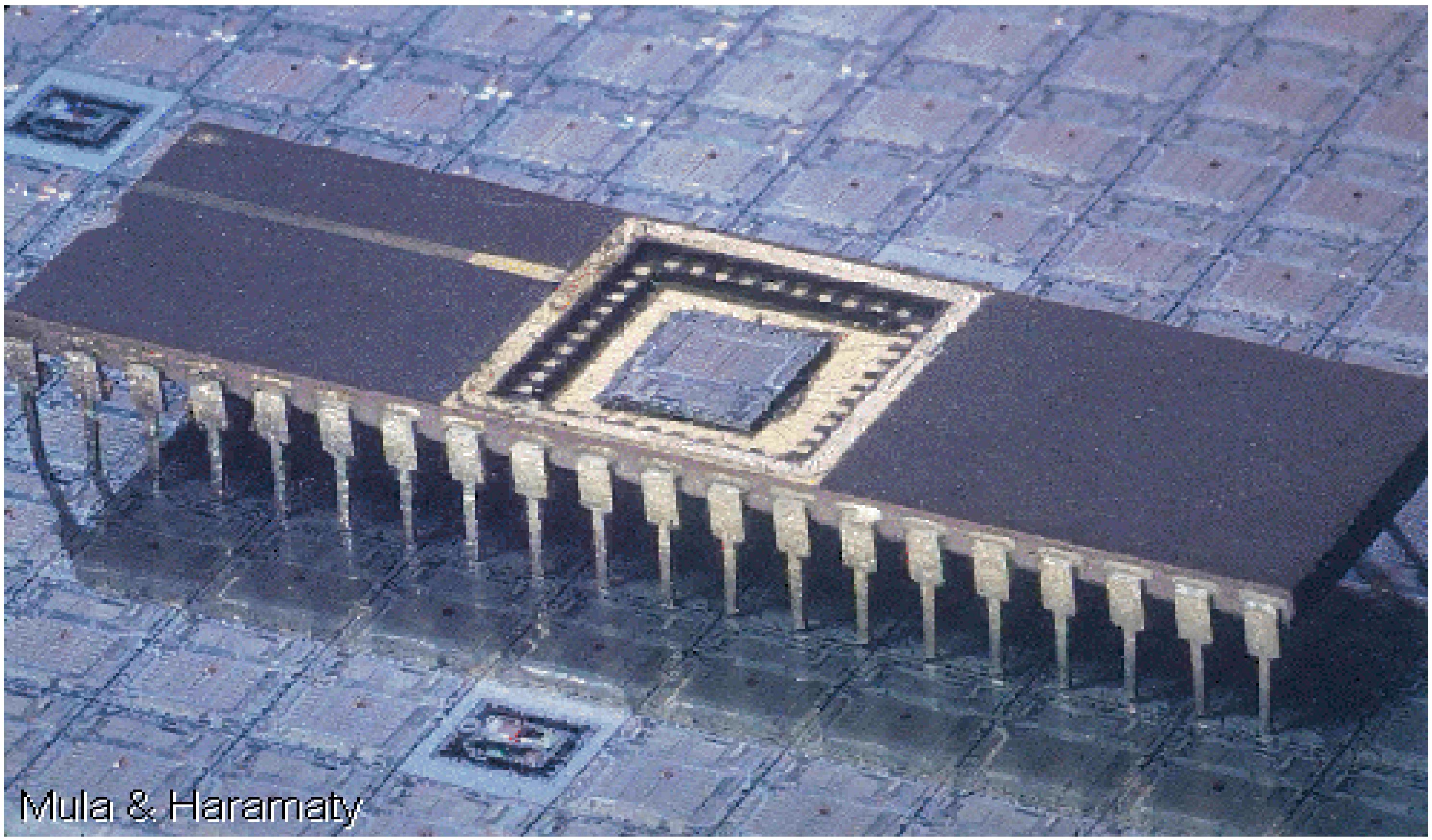
- Instructor: Dr. Sahra Sedigh-Ali
- Email: sedighs@umr.edu
- Phone: 341-7505
- Office: EECH 135
- Office Hours: Tuesdays and Thursdays 1:00-2:30, or by appointment.
- Email is the best way to reach me.

Introduction (continued)

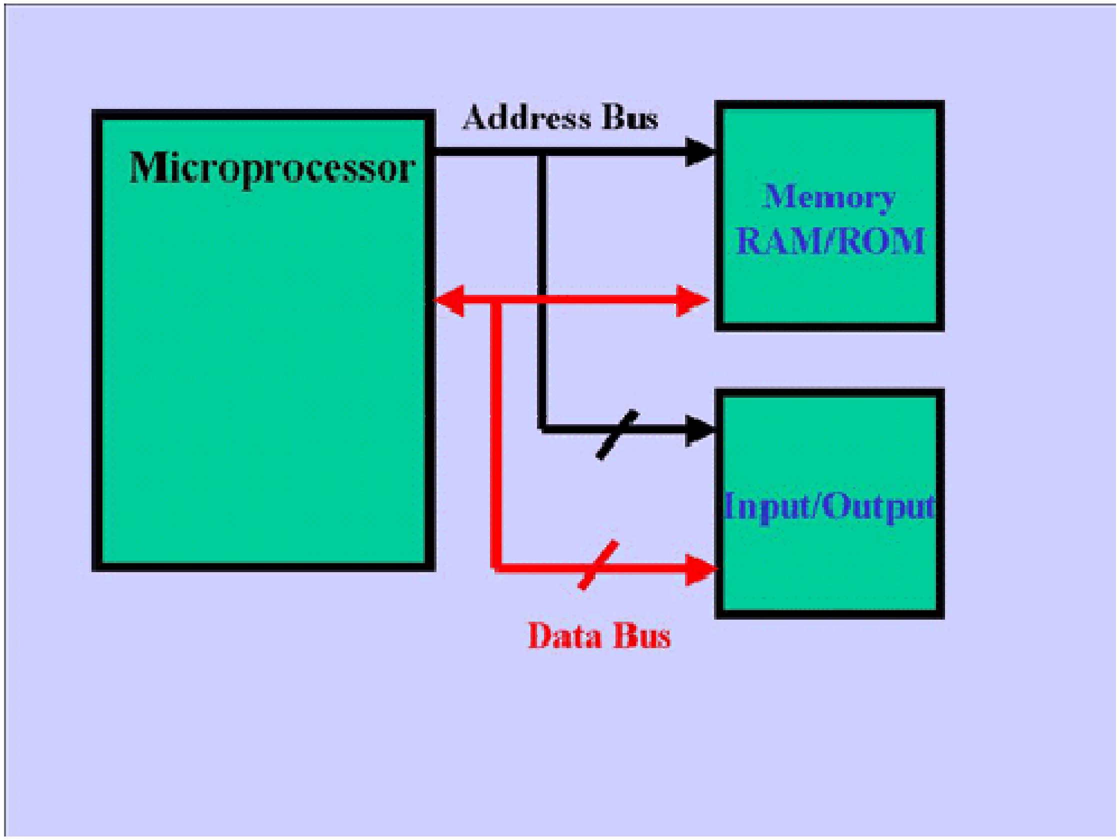
Definitions

- : A device that stores and retrieves data and sequentially executes a stored program without human intervention.
- : a computer that is contained in a single integrated circuit.
- : A microprocessor with a number of integrated peripherals, typically used in control-oriented applications.

A Typical Microprocessor



General Layout of a Basic μP System



- Most of these components are common to all processors.

Definition of a Microcontroller

- A is a single-chip computer that contains many of the same items that a desktop computer has, such as μP, memory, etc., but does not include any "human interface" devices like a monitor, keyboard, or mouse.
- Microcontroller =
- Microcontrollers are designed for , rather than human interaction.
- Micro suggests that the device is small, and controller tells you that the device might be used to control objects, processes, or events.

What is an Embedded Controller?

- Another term to describe a microcontroller is embedded controller, because the microcontroller and its support circuits are often built into, or embedded in, the devices they control.

Topics covered in this course

1. Introduction to microprocessor organization and operation, emphasizing the 8051 microprocessor subset (the WIMP51).
2. Introduction to computer architecture, with emphasis on systems involving the 8051 microcontroller.
3. Machine and assembly language programming for the Intel 8051 and variants.
4. C language programming for embedded systems.
5. More later …

Course Objectives

- By the end of the course, you should be able to:
  - Understand the organization of a simple microcontroller.
  - Analyze and design hardware and software for small digital systems involving microcontrollers.
  - Program embedded computer systems in Assembly and C.
  - Use the 8051 microcontroller and its standard peripherals.

Why this course is important

- Microcontrollers are used extensively in process control, instrumentation, home appliances, automobiles, etc. – they represent a *basic building block* of modern digital systems design.
- If you go into virtually any form of engineering design, there is a high probability that knowledge of microcontrollers will be required.
- Microcontrollers are the basis of <u>embedded systems.</u>

Slide adapted from Dr. David Meyer.

Microprocessors vs. Microcontrollers

	Microprocessor	**Microcontroller**
Data formats		
Instruction types/modes		
Hardware Architecture		
Applications		
I/O		

Pros and Cons of Microcontrollers

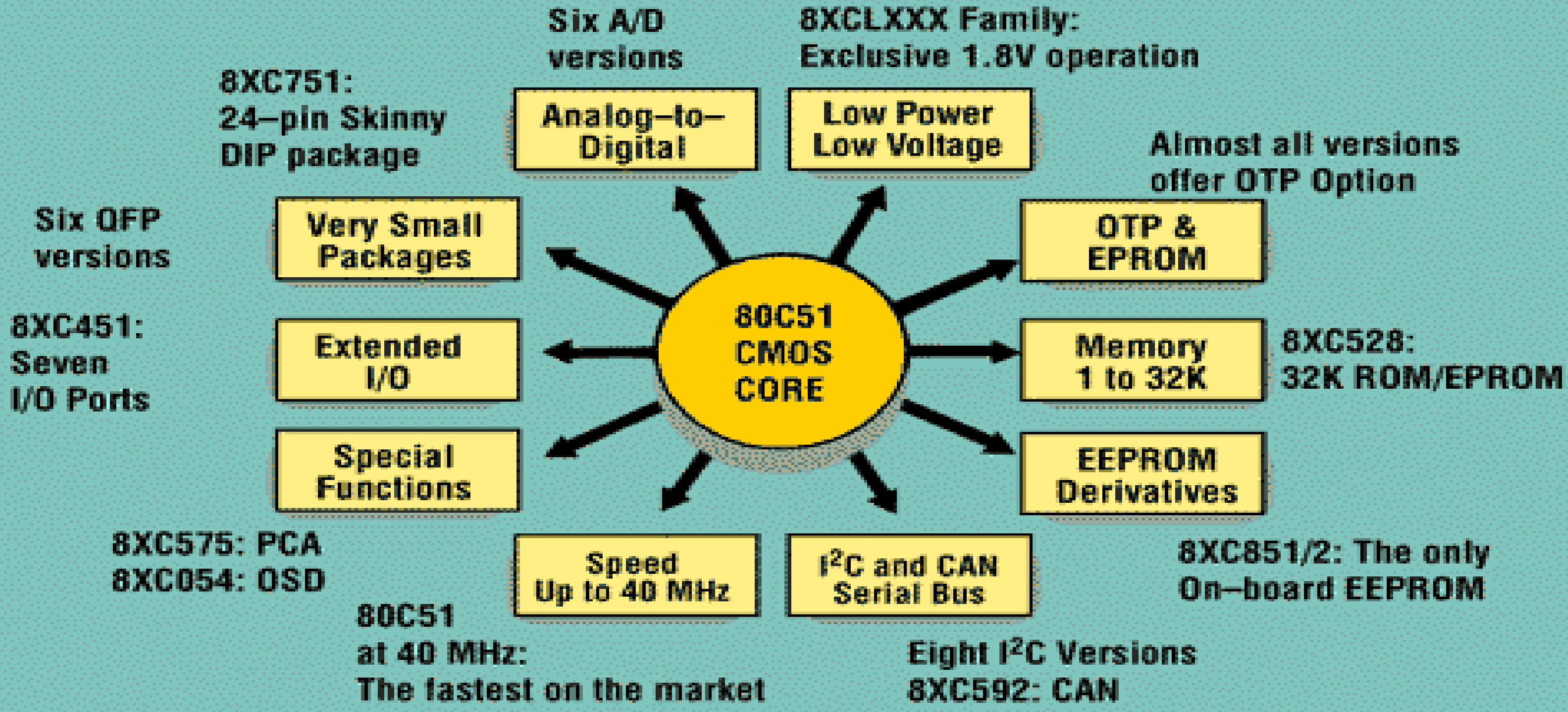
Advantages	Disadvantages
Low cost	Poor versatility
response (distributed and controlled applications)	response (compared to discrete circuits)
Space, size of system	Limited space
Efficiency	Limited instruction set
“Embeddability”	
Low power usage	

Microcontroller Types

- Microcontroller models vary in data size from 4 to 32 bits.
- 4-bit units are produced in huge volumes for very simple applications.
- 8-bit units are the most versatile.
- 16 and 32-bit units are used in high-speed control and signal processing applications.

Why the 8051 microcontroller?

- Classic
- Most popular
- Plenty of applications, peripherals, and development tools
- More than 150 variants of 8051 are offered by more than 20 vendors
  - over 126 million components sold annually
- We will learn about what's inside, how to program, and how to design around the 8051.
- These concepts are fundamental to digital systems design in general.



Some Key 80C51 Derivatives

C programming

- A typical ‘desktop C’ program
- What does it do?
- What’s wrong with it?

Hello world in Desktop C

- Invoked by a console command
- Writes ‘Hello World’ followed by new line to stdio output device (console display)
- Returns to operating system when finished
- That is also what is wrong with it!

Embedded Hello World Program

```
#include <dos.h>            /* outportb prototype */
void main(){
#define LPT1 0x378          /* normal address for LPT1 */
#define PIN2 1     /* pin 2 of 25 pin D connector */
 while(1){         /* embedded pgms never return */
  outportb(LPT1,PIN2);  /* set bit 0, pin 2 */
  delay(500);           /* delay 500 mSec */
  outportb(LPT1,0);     /* clear bit */
  delay(500);           /* delay 500 mSec */
  }
}
```

Embedded software attributes

- Stored in
- Started at power on
- Runs forever
- is important
  - is often more important than the actual value
- Cost is important - use minimal resources to run (memory, clock speed, power, parts)

Some embedded applications

- Automotive Applications
- Telecommunications
- Consumer Electronics
- Industrial Controls
- Aerospace

Automotive Applications

- As many as 22 micros in GM cars and trucks
- Over 60 micros and 4 networks in late model Volvo
- Automatic climate control
- Anti-lock brakes, traction control
- Stability enhancement
- Driver information centers
- Supplemental restraint systems (SRS)
- Real time damping
- Navigation systems
- Remote keyless entry

Telecommunications

- Satellite communications
- Cordless phones
- Cellular phones, pagers, infrastructure
- An 8051 (or variant) in EVERY Adtran product
- Cost about $1 in large quantities
- Used to control ASICs

Other embedded applications

- Consumer electronics
- Industrial Controls
  - Labeling machines
  - Coin Acceptors
- Aerospace
  - Flight control systems
  - Navigation systems

Review of Number Systems

Some slides adapted from Dr. S. Noushabadi

Decimal Numbers: Base 10

- Digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Example:

  3271 =
-

Numbers: positional notation

- Number Base B => B symbols per digit:
  - Base 10 (Decimal): 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
    Base 2 (Binary): 0, 1
- Number representation:
  - $d_{31}d_{30} \ldots d_2d_1d_0$ is a 32-digit number
  - value = $d_{31} \times B^{31} + d_{30} \times B^{30} + \ldots + d_2 \times B^2 + d_1 \times B^1 + d_0 \times B^0$
- Binary: 0,1
  - 1011010 =

  - Notice that a 7-digit binary number converts into a 2-digit decimal number
  - Which base(s) convert(s) to binary easily?

Hexadecimal Numbers: Base 16

- Digits: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
- Normal digits have expected values
- In addition:
  - A → 10
  - B → 11
  - C → 12
  - D → 13
  - E → 14
  - F → 15

Hexadecimal Numbers: Base 16

- Example (convert hex to decimal):
  - B28F0DD =

decimal

- Notice that a 7-digit hex number is converted to a 9-digit decimal number

Decimal vs. Hexadecimal vs.Binary

- Examples:
- 1010 1100 0101 (binary)

= (hex)

- 10111 (binary)

= (binary)

= (hex)

- 3F9(hex)

= (binary)

00	0	0000
01	1	0001
02	2	0010
03	3	0011
04	4	0100
05	5	0101
06	6	0110
07	7	0111
08	8	1000
09	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Hex to Binary Conversion

- HEX is a more compact representation of binary.
- Each hex digit represents 16 decimal values.
- Four binary digits represent 16 decimal values.
- Therefore, each hex digit can replace four binary digits.
- Example:
  - 0011 1011 1001 1010 1100 1010 0000 0000 binary
  - hex
  - C uses notation 0x

Which Base Should We Use?

- Decimal: Great for humans; most arithmetic is done with this base.
- Binary: This is what computers use, so get used to them. Become familiar with how to do basic arithmetic with them (+,-,*,/).
- Hex: Terrible for arithmetic; but if we are looking at long strings of binary numbers, it’s much easier to convert them to hex and look at four bits at a time.

What can we do with binary representations of numbers?

- Everything we can do with decimal numbers.
  - Addition
  - Subtraction
  - Multiplication
  - Division
  - Comparison
- Example: 10 + 7 = 17

```
      1   1
      1   0   1   0
+     0   1   1   1
-------------------------
```

  - so simple to add in binary that we can build circuits to do it
  - subtraction also just as in decimal

Complement Number (CN) Representation

- in CN, the negative of a number is equal to the so-called "complement" of the number
- there are two popular complements
  - radix complement of an n digit number D is given by $r^n - D$, where r is the radix (or base of the number)
  - diminished radix complement – we are not interested in this one for this course

2’s Complement notation

- 2’s complement of a binary number, D
  - the complement equals $r^n - D = 2^n - D$
- alternative convenient way of finding 2’s complement
  - write the number in the binary form
  - flip all bits
  - add one
    - drop any carry out of MSB

8-bit examples

	17_{10} =00010001_2	0_{10} = 0000000_2	1_{10} = 0000001_2
flip bits			
add 1			

fact: a negative number will always have a 1 in the MSB

Conversions in 2's complement

- 10111_2 = ?
  -

- note that if we are not using signed numbers,

- given n bits, the range of representable numbers in 2's complement is -2^{n-1} to $(2^{n-1} - 1)$

2’s complement addition/subtraction

- add 2’s complement numbers just as we would add positive numbers, and ignore the carry out of the “sign bit”
  - sign bit is the MSB

For Friday

- Review today's lecture notes.
- Print lecture notes for Lecture 3.
- Email me names of your group members (one email per group).