CpE 213
Digital Systems Design

Lecture 1
Tuesday 8/26/2003

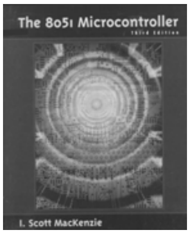
Basic information

- Instructor: Sahra Sedigh-Ali
- Email: sedighali@ieee.org
- Phone: 341-7505
- Office: EECH 219
- Office Hours: TR 1:30-3:00, or by appointment.
- Email is the best way to reach me.

Course prerequisites

- CpE 111 (Intro to Comp Eng)
- a C programming course (CS 53 or CS 74)
- CpE 214 is strongly recommended as a co-requisite.

Required textbook



The 8051 Microcontroller
I. Scott MacKenzie
3rd edition
Prentice Hall, 1999.

A number of recommended supplemental texts are listed in your syllabus.

Homework

- You are expected to read appropriate sections of the textbook before presentation in class.
- Homework problems will be assigned in class.
- Assignments are due at the beginning of class.
- No late homework will be accepted.
- While you are expected to complete all homework assignments, the grader may randomly select only certain problems to grade.

Quizzes

- There will be 3-5 brief announced or unannounced quizzes in class.
- A score of zero will be given for a quiz in case of absence.
- No makeup quizzes will be given.

Grading (tentative)

- Exam 1: 25%
- Exam 2: 25%
- Project (tentative): 10%
- Final Exam (Comprehensive): 30%
- Assignments & Quizzes: 10 %

Exam dates

- Exam 1: Tuesday, September 30
- Exam 2: Thursday, November 6
- Final Exam: Friday, December 19
  (8:00-10:00 am).
  Location to be announced.

Academic Honesty

- Any action that might unfairly improve a student's score on homework, quizzes, or examinations will be considered cheating, and will not be tolerated.
- Cheating on assignments or exams can result in a zero score for the assignment or exam, or a reduced or failing grade for the course, at the discretion of the instructor.
- Instances of cheating will be reported to university administrative officials for further action and possible suspension or expulsion from the University.

Feedback

- Your feedback is critical to my success as an instructor.
- Please return the index cards that I will periodically distribute.
- Feedback can be provided anonymously.
- Your comments are appreciated and are welcome throughout the semester.

Important reminder

- The course syllabus is a <u>legally binding agreement</u> between you and your instructor.
- Portions have been skipped in class.
- <u>Please read it in its entirety</u>.

Bad ways of learning design

- Relying solely on lectures on how to solve design problems
- Reading about how to solve design problems
- Watching someone else solve the design problem, or reading their solution
- atching someone else solve the design problem, or reading their solution
- You will forget most of what you hear, but will remember most of what you put into practice.

Slide adapted from Dr. David Meyer.

Before we begin …

- Every one of you can do well on this course.
- Do not jeopardize your success by:
  - skipping classes
  - failing to do the homework problems
  - procrastinating
  - expecting to “learn by osmosis”
  - being dishonest in any way.

Slide adapted from Dr. David Meyer.

Definitions

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

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 an 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:

- Analyze and design hardware and software for small digital systems involving microcontrollers.
- Understand the organization of a simple digital computer.
- Use the 8051 microcontroller and its standard peripherals.
- Apply the C language in embedded computer systems.

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 embedded systems.

Slide adapted from Dr. David Meyer.

Global microcontroller market growth

- From the June 2003 midyear forecast of the Semiconductor Industry Association:

  “The global microcontroller market, driven by consumer and automotive applications, will increase 9.9 percent to $10.3 billion in 2003, and then grow 14.0 percent to $11.7 billion in 2004, 3.2 percent to $12.1 billion in 2005, and 18.5 percent to $14.3 billion by 2006.”

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

C programming

```
/* The infamous 'Hello World'
  program */
void main(){
 printf("Hello World.\n");
 }
```

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 newline to stdio output device (console display)
- returns to operating system when finished
- That is also what is wrong with it!
  - Embedded systems have no console
  - No body to see it if it did have one
  - No operating system to return to

An 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 read only memory (ROM)
- Started at power on
- Runs forever
- Reaction time is important
  - predictability 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

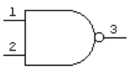
Other embedded applications

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

Another embedded application:

```
void main(){
unsigned int pc;
char rom[2048] = (0,1,2);
char ir;
pc = 0;
  while(1){
    ir= rom[++pc];
    switch (ir)
    {case 0: nop(); break;
     case 1: add(); break;
     /*...*/
    }}}
```

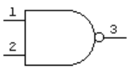
Hardware review



P1	P2	P3
0	0	
0	1	
1	0	
1	1	

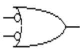
Name:
Logic equation:
DeMorgan equivalent symbol:

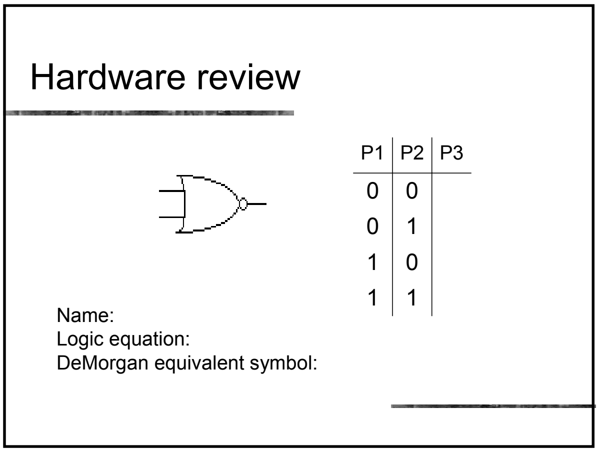
Hardware review



P1	P2	P3
0	0	1
0	1	1
1	0	1
1	1	0

Name: NAND2
Logic equation: P3 <= not (P1 and P2);
DeMorgan equivalent symbol:
P3 <= not P1 or not P2

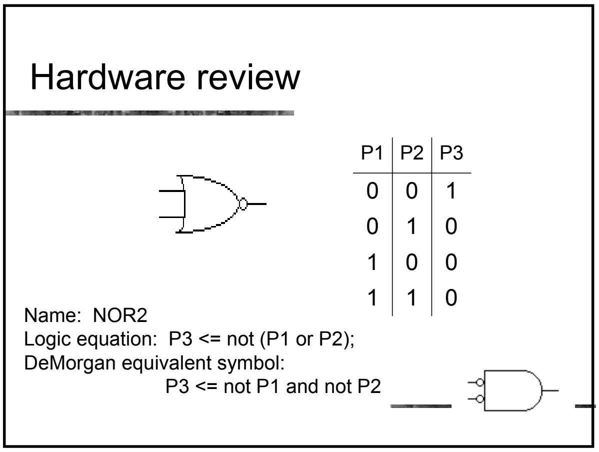




Hardware review

P1	P2	P3
0	0	
0	1	
1	0	
1	1	

Name:
Logic equation:
DeMorgan equivalent symbol:



Hardware review

P1	P2	P3
0	0	1
0	1	0
1	0	0
1	1	0

Name: NOR2
Logic equation: P3 <= not (P1 or P2);
DeMorgan equivalent symbol:
P3 <= not P1 and not P2