Digital Logic Design

Chapter 1

Introduction

Introduction

- This course is about Design Techniques for **Digital System**, a more exact name will be **Synchronous Digital Hardware System**.
- Synchronous means clocked i.e. all changes in the system are controlled by a global clock and happen at the same time
- Digital means all values (input, output, and internal) can take on discrete values.
- A/D if the input is analog (voice or music).

- **Text**: Digital Design, Mano and Ciletti 4th ED Prentice Hall
- **References**:
- Digital Design: Principles and Practices, Wakerly, Prentice Hall
- Advanced Digital Design with the Verilog HDL, M. Ciletti, Prentice hall
- Contemporary Logic Design, Katz and Borriello, Prentice Hall

HW	0%
3 quizzes	10%
Lab	10%
Project	10%
Midterm	25%
Final	45%

Course contents

- Number system and how to represent things digitally.
- Boolean algebra and logic circuits.
- Combinational design
- Sequential design
- This is not a course on transistor physics or circuits, but we need to know these to better understand the building blocks of the system.
- Not a course on computer organization, but we will look at these as example of what we can do

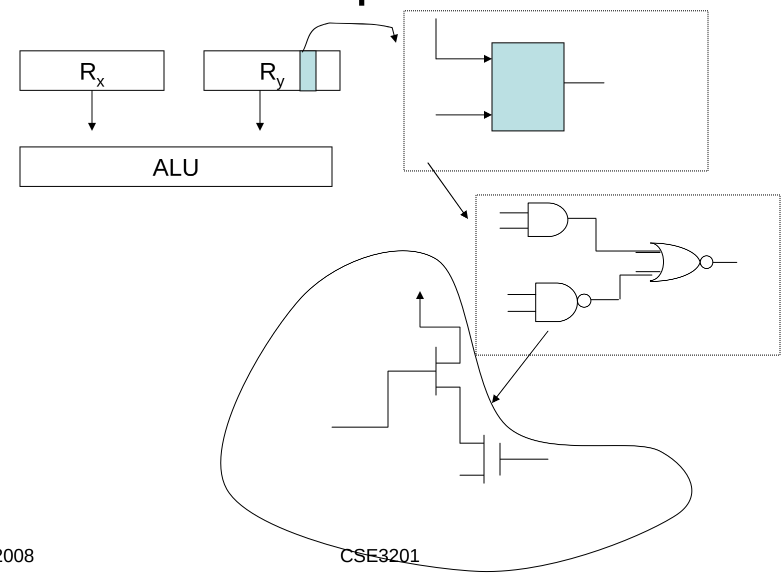
Good Design

- Digital systems are very complex and large, in order to do a good design, consider these issues:
  - Modularize your design
  - Top-down Design
  - Bottom-up Design
- Design issues: **Speed**, **Cost**, **Power**
- Usually these are contradictory (a fast system is not cheap.
- Design is more of an art than science, but luckily we have measures for the design (cost, speed, .)

Specifications

- Like in any other design, we start with the specification.
- The specification is basically what do we want to achieve.
  - High level specification
  - Binary level specification
  - Algorithmic level specification

Level of Implementation



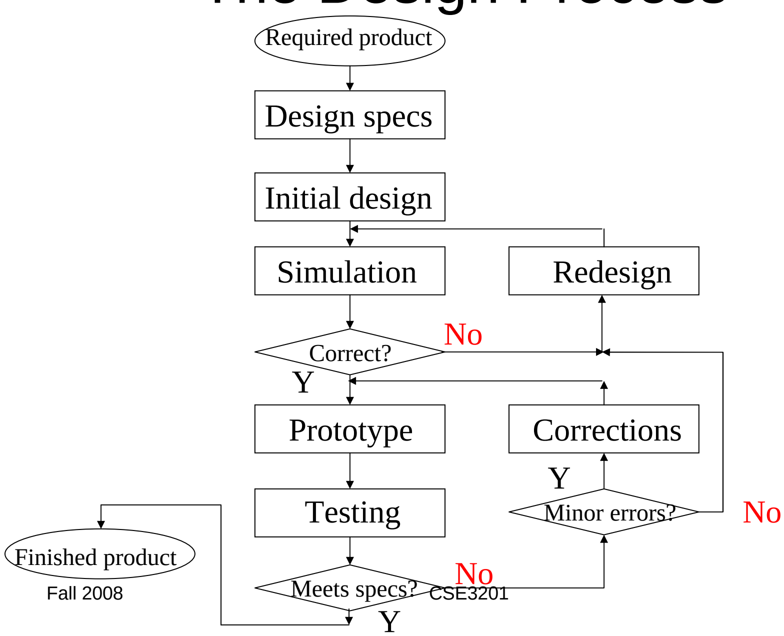
CAD TOOLS

- Nowadays, design is usually automated, we use design tools for our design.
- HDL is used to describe the system (or the proposed system) in a high-level C-like language.
- Synthesis tools are used to map this design to FPGA
- Simulation tools are used to check the design (timing or functional simulation).
- Finally, testing

Implementation

- Integrated Circuits (IC's)
  - Crystalline silicon
  - 1—100's of Millions of transistors
  - Feature size 0.13um or 0.13 90 n (0.09micron) 60n and dropping …
  - CMOS (mostly)
  - Standard microprocessors
  - ASIC (application Specific IC's)
  - FPGA's (Field Programmable Gate Arrays)
- Printed Ciruits Board (PCB)
  - Fiberglass or ceramics
  - Many conductive layers (1-20)
  - Multiple Chips Modules (MCM's) multiple chips directly connected to a substarte
- Chassis

The Design Process



Number System

- Decimal numbers (9735) = $9 \times 10^3 + 7 \times 10^2 + 3 \times 10^1 + 5\ 10^0$
- Binary numbers (101) = 5 decimal
- How to convert 29 to binary (successive division by 2 the answer is 11101

Number	Quotient	Remainder
29/2	14	1
14/2	7	0
7/2	3	1
3/2	1	1
1/2	0	1

Number System

- Octal and Hex
- Conversion is the same idea

Complements—Diminished radix

- Given a number *N*, in base *r* having *n* digits is defined as $(r^n-1)-N$
- For example 9^{th} complement of 456325 is 543674
- The 1's complement of any binary number is obtained by changing every 1 to 0 and every 0 to 1
- The 1's complement of 101100010 is 010011101

Radix Complement

- Given a number *N*, in base *r* having *n* digits is defined as $r^n - N = (r^n - 1) - N + 1$
- For the 10^{th} complement, the rule is
- Leave the least significant 0's unchanged, the first nonzero digit is subtracted from 10, the rest of the digit are subtracted from 9 10^{th} complement of 3451600 is 6548400
- For 2's complement, the LSB zeros are left unchanged, the first 1 un changed, the remaining bits are complemented 010010 110110

Signed Binary Numbers

Decimal	2's complement	1's complement	signed magnitude
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
+0	0000	0000	0000
-0		1111	1000
-1	1111	1110	1001
-2	1110	1101	1010
-3	1101	1100	1011
-4	1100	1011	1100
-5	1011	1010	1101
-6	1010	1001	1110
-7	1001	1000	1111
-8	1000		

Subtraction with complement

- To subtract two n-digit unsigned numbers *M-N* in base *r* is done as follows

1. Add the minuend *M* to the *r's* complement of the subtrahend *N yielding* $M+(r^n - N) = M-N+r^n$
2. *If* $M \geq N$ The sum will produce an end carry, that is basically the r^n
3. *If* $M \leq N$ The sum does not produce a carry and is equal to $r^n - (N-M)$ which is the *r's* complement of (*N-M*). To obtain the answer take the *r's* complement of the sum and place a -ve sign next to it.

Addition (examples)

Addition (Example)

Binary Codes

- **BCD** Note that in adding 2 BCD numbers, the digits are added as if they are 2 binary numbers, if the result is greater than or equal 1010, we add 0110 to obtain the correct BCD digit sum and a carry
- **Gray code** (why do we care?)
- **ASCII** and ASCII with parity

BCD Addition

Gray Codes

- Only I digit change when we go from any number to number+1
- To form a sequence, put the sequence from left to right, followed by the sequence reversed. Add 0 as a MSB to the left sequence and 1 to the right sequence

0	1						
00	01	11	10				
000	001	011	010	110	111	101	100
0	1	3	2	6	7	5	4

Register Transfer Logic

- Example R2 → R1
- What is an R?

Binary Logic

- Low 0-1 volt
- High 1.2-4 volts
- In-between not defined
- AND, OR NOT (EXOR, NAND, NOR)