



Digital systems



Course Details

Lecturer - Ido Bergman
 TA - Yehuda Arav
 Telephone
 Silverman : 6585956
 Ross: 6585297
 Email: dlocs@cs.huji.ac.il


Course Site: www.huji.ac.il/~dlocs




Course Structure - Grading

Exercises grading

Exercise Size	Number	Weight	Total Weight
Small Exercise	8	7%	56%
Large Exercise	4	11%	44%




Course Structure - Grading

$$\text{Total Grade} = \max \left(\begin{array}{l} \text{Total Exercise} * 0.2 + \text{Exam Grade} * 0.8 \\ \text{Exam Grade} \end{array} \right)$$


Registering the course

Registered in:

<http://httpdyn.cs.huji.ac.il/course-admin/dlocs/register>



Course Structure

Introduction - representing data

- Representation of strings (Binary Codes).
- Representation of Numbers in different bases.
- Representation of Integer numbers (1 and 2 complement).
- Representation of Fixed point numbers
- Representation of Floating Numbers
 - Addition and subtraction
 - Multiplication and division.
 - Rounding errors and overflow

Course Structure

Boolean Algebra.

1. Axioms
2. Basic Theorems of boolean algebra.
3. Literals and Boolean functions.
4. Various gates (nand,nor,xor and ect.) (including multiple gates).
5. Canonical forms.
6. Completeness of systems
 1. Explanation of completeness
 2. Proving that AND,NOT or OR,NOT is complete.
 3. Examples of other complete systems:
 1. NAND
 2. Friedkin and Toffoli
 4. Proving that a system is not complete.
7. Simplification of representation
 1. Karnaugh maps.
 1. Maps of 3,4,5,6 variable.
 2. Indifferent combination.
 2. The table method.

Course Structure

Basic Combinatorial Circuits

1. Representation of binary system in computers
 1. Positive and negative logics.
 2. Noise (the bounds of logic).
2. Switches
 1. P and N switches.
 2. Construction of the elements: NOT, AND and OR.
3. Timing of the single element (From Wakerly - Digital Design) :
 1. Objective
 2. Different concepts and terms in timing.
 3. Hazards - Static and Dynamic.
4. Timing of circuit.

Course Structure

Advanced Combinatorial Circuits

1. Example 1 - simple Digit decoder
2. Example 2 - semi and full adders.
3. Presenting the synthesis procedure for sequential circuits.
4. Analysis.
5. Presenting various components (not necessarily a full list):
 1. subtractors.
 2. 4-bit adder
 3. Ripple add
 4. Comparator
 5. Fast addition with look ahead carry.
 6. BCD adder.
 7. Multiplexer (with and without enable) and demultiplexer
 8. Encoder and Decoder (and priority decoder) with or without enable.
 9. Multiplication of binary numbers.
 10. Decimal Adder.
 11. Three State Device.

Course Structure

Basic Sequential Circuits

1. Synchronous and a-synchronous systems.
2. Latches.
 1. Basic SR-Latch implementation
 2. Time diagram
 3. D-latch
 4. JK-latch
 5. T-latch.
3. Problems with a-synchronous circuits
 1. Solution 1. Time controlled Latch (Gate Latch)
 1. 5.1.1 Disadvantages
 2. Solution 2. Flip-Flops
4. Flip-Flops
 1. Triggering FlipFlops
 2. Edge-triggered Flip-Flops.
 1. Positive edge
 2. Negative edge (master slave).
5. Triggering tables.

Course Structure

Advanced Sequential Circuits I

1. Synthesis of sequential+combinatorial circuits.
 1. State machines - Mealy and moore
 2. The synthesis procedure.
 3. Analysis procedure.
2. Registers
 1. Introduction
 2. Simple registers
 3. Registers with parallel loadings
 4. Shift registers
 5. Shift register with parallel loading.
3. Counters
 1. Introduction
 2. Counting up and down.
 3. Example - Counting module 6.
 4. Ripple Counter.

Course Structure

Advanced Sequential Circuits II

1. Control structures.
 1. parallel control (Johnson counter)
 2. sequential control.
2. ROM and RAM
3. Memory BUS
 1. Introduction
 2. Three states device.
 3. Bus implementation.

Course Structure

Computer organization - Basics.

1. Computer design
 1. CPU
 2. Control unit
 3. ALU

Digital system

Machines that handles discrete information units

For example:

Computers
Calculators
Telephone Switches
Cell phones

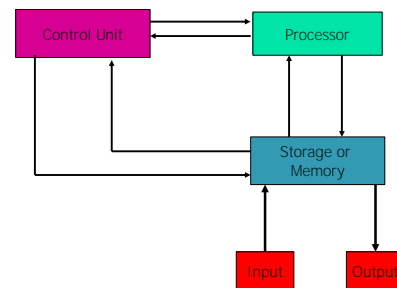
Implementantion Constraint

A reliable electric system has two modes:

On
Off

All data must be represented under These constraints

Scheme of a Computer



Representing data

Bit – Binary digit.

Can have values:

1 (on)
0 (off)

A string of bits can represent data

Representing numbers

Number	BCD representation
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Weight : 421

More number codings

Number	8 4 -2 -1	2421
0	0000	0000
1	0111	0001
2	0110	0010
3	0101	0011
4	0100	0100
5	1011	1011
6	1010	1100
7	1001	1101

} Same value, different Coding

Error Detecting Codes

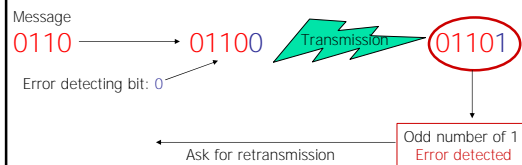
Binary transmission are prone to errors

We add some information to allow us to detect error

Parity bit – odd or even

Message	P (odd)	P (even)
0000	1	0
0001	0	1
0010	0	1

Error detecting (even)



Alphanumeric code

Character	8-bit EBCDIC	ASCII <small>American Standard Code for Information Interchange</small>
a	1000 0001	0110 0001
A	1100 0001	0100 0001
1	1111 0001	0011 0001
\$	0101 1011	0010 0100

Gray code

Gray Code	Number
000	0
001	1
011	2
010	3
110	4