C SC 355
Fall 2016

LillAnne Jackson

Introduction

Today's Overview:

- Course Introduction & Outline
- Digital systems
- Intro Boolean Algebra

Today's Readings:

Mano/Kime/Martin:
Chapter 1

Course Information

https://heat.csc.uvic.ca/coview/outline/2016/Fall/CSC/355

Email: lillanne at uvic.ca
Office: ECS 563
Phone Number: < Please E-mail >
Office Hours:

Tuesday or Wednesday 9:30-10 am (EOW 206 lobby)
Friday 12:30pm-01:45pm (Somewhere in this building)

Textbook: Logic and Computer Design Fundamentals, 4th or 5th Edition, M. Morris Mano, Charles R. Kime, Prentice-Hall

Course Objectives:

At the end of the course, you will be able to:

- design, analyze and implement (build!) a digital circuit.
- understand the fundamentals of digital design.
- appreciate the need for the testing of all digital systems.
- explain the structure of hardware components in microprocessor systems.
- remember an enjoyable and stimulating course!

Course Topics:

- Boolean algebra, logic gates, combinational circuits
- Simplification and design algorithms
- Registers, memory organization and buses
- Mealy and Moore finite state machines, bubble graphs, ASM charts
- Sequential machines analysis and synthesis, counters
- Arithmetic logic units and control units
- Testing of circuits
- Computer organization
- Design of a microprocessor
- Memory organization
- VHDL programming
- Programmable devices (FPGAs)

Your Schedule

Draft Schedule: Subject to change

Week	Reading:	Lecture 1	Topic	Lecture 2	Topic	Lecture 3	Topic	Lab Topic	Assignment
1		Sep-06	No Class	Sep-07	Intro	Sep-09	Combinational Logic I		
2		Sep-13	Combinational Logic II	Sep-14	Cannonical Forms (Minterms)	Sep-16	Cannonical Forms (Maxterms)	Lab 1: Intro to Logic Gates and Design Works	
3		Sep-20	Karnaugh Maps Intro (2-4 Variables)	Sep-21	Combinational Circuits I	Sep-23	Combinational Circuits II	Lab 2: Combinational Circuits	
4		Sep-27	Combinational Circuits II	Sep-28	Karnaugh Maps 5 Variables	Sep-30	Karnaugh Maps 6 Variables	NO LABS this week -Complete Assignment 1	Assignment 1 Due - Thursday in class
5		Oct-04	Arithmethic Ccts: Adders	Oct-05	Arithmetic Ccts Subtractors	Oct-07	Arithmetic Ccts	Lab 3: Multiplexers and Demultiplexers	
6		Oct-11	Review	Oct-12	VHDL	Oct-14	VHDL & Timing	NO LABS: prepare for Midterm	
7		Oct-18	Midterm Exam	Oct-19	Sequential Circuits I	Oct-21	Sequential Circuits II	Lab 4: VHDL for Combinational Circuits	
8		Oct-25	Sequential Circuits III	Oct-25	FSM Analysis	Oct-28	FSM Design I	Lab 5: Intro to Sequential Circuits	
9		Nov-01	FSM Design II	Nov-02	FSM Design III	Nov-04	VHDL3 Sequential	Lab 6: Finite State Machines	
10		Nov-08	Mealy-Moore	Nov-09	HOLIDAY: REading Break	Nov-11	HOLIDAY: REading Break	NO LABS: Complete Assignment 2	Assignment 2 Due - Tuesday in Class
11		Nov-15	FSM Game Day	Nov-16	Registers	Nov-18	Small Computer Systems	Lab 7: VHDL for Sequential Circuits	
12		Nov-22	Small Computer Systems	Nov-23	Testing, Programmable Logic Arrays, Gate Arrays	Nov-25		Lab 8: Very Small Computer System - Implementation	
13		Nov-29	Review	Nov-30	Review	Dec-02	No Class: Day of Rememberance	NO LABS: Complete Assignment 3	Assignment 3 Due -Wednesday in Class
14		FINAL EXAM PERIOD	December 5 through 19	Don't plan anything until the final Final Exam Schedule is posted!					
								October 31: Last day for withdrawing from courses	
		Normally, all pre-lab exercises will be due in the Friday class, for the lab that will be completed the following week.							

Grading

Coursework	Weight
Assignments (3 @ 4%)	12%
Lab Exercises(8 @ 2%)	16%
Midterm Exam	25%
Finite State Machine Game	2%
Final Exam	45%

Administrivia

- You have to attend <u>your</u> lab
- failure to submit the prelab implies you cannot attend the lab: Grade for the lab = 0.
- Late assignments will be evaluated (eventually) but not included in your grade
- Do not make plans for Final Exam period until the final final exam schedule has been posted.
- to pass the course, you must obtain pass:
  (a) the course overall;
  (b) the final exam; and
  (c) the labs overall.
- Web site: connex.csc.uvic.ca
  Use your UVic NetLink ID to log in

Please find your Course Outline:

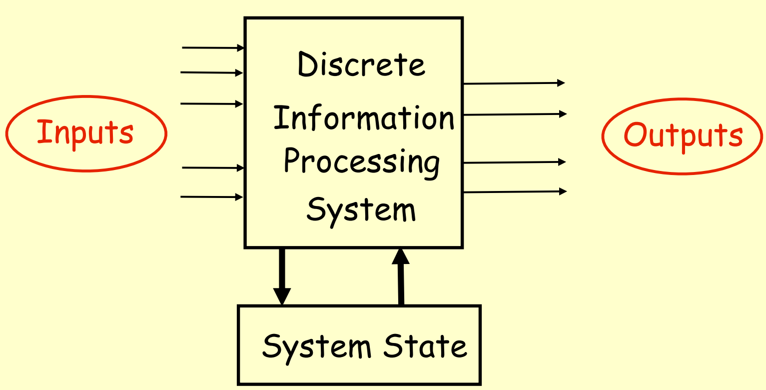
- www.csc.uvic.ca
  - Current students
  - Undergraduate
  - Undergraduate Courses
  - CSC 355
  - Fall 2016 Outline

And now for some Digital Logic !!

QUESTIONS?

A Digital System

Converts a set *inputs* (information) into a set of *outputs (information)*



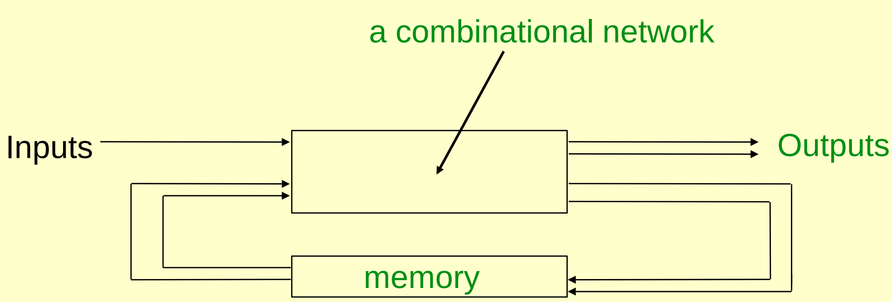
AKA: A Switching Network

switching? Were once based on switches,
open and closed

COMBINATIONAL:
current output of the network depends entirely on current inputs ==> no memory

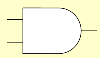
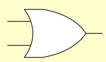
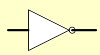
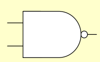
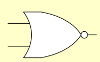
SEQUENTIAL:
network with memory of any type
==> output depends on current inputs plus previous state of inputs

A Sequential Network



For now, lets focus on Combinational Logic

GATES :

- AND 
- OR 
- NOT 
- NAND  (NOT AND)
- NOR  (NOT OR)

Signal:

A variable that represents a physical quantity. Two discrete levels, or binary values, are the usual in digital systems.

Binary values are represented by:

- binary digits: 0 and 1
- labels: False (F) and True (T)
- heights: Low (L) and High (H)
- Switch values: Off and On

Example #1

IF the garage door is open

AND the car is running

THEN the car can be backed out of the garage

Is garage door open?	Is the car running?	Can the car be backed out?
false (0)	false (0)	false (0)
false (0)	true (1)	false (0)
true (1)	false (0)	false (0)
true (1)	true (1)	true (1)

Boolean Algebra and Logical Operators

Algebra: a branch of mathematics in which symbols (as letters and numbers) are combined according to the rules of arithmetic. (Merriam-Webster Dictionary)

Boolean Algebra: the branch of *algebra* in which the values of the variables are the truth values true and false, usually denoted 1 and 0 respectively. (Wikipedia)

Values used: 0 and 1

Operators: AND, OR, NOT

AND, OR, NOT

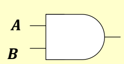
A	B	AND
0	0	0
0	1	0
1	0	0
1	1	1

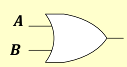
A	B	OR
0	0	0
0	1	1
1	0	1
1	1	1

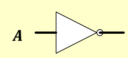
A	NOT
0	1
1	0

Truth Table: a table of all possible input combinations and output values

$A \cdot B$ $A + B$ $\bar{A}$







Example #2

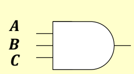
Turn on the furnace if, the internal temperature is less than 20°C and the door and the window are both closed.

A: internal temperature <20

B: door closed

C: window closed

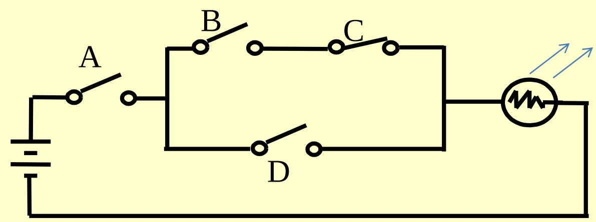
$A \cdot B \cdot C$



A	B	C	Furnace
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Example #1

What is this expression?



F(light on) = A AND [(B AND C) OR D]

Example #2: A Boolean Expression

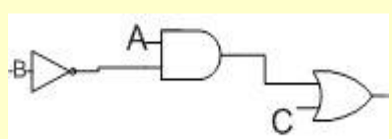
$$A\,\overline{B} + C$$

Build the truth table

3 variables $\Rightarrow 2^3 = 8$ rows

A	B	C	$\overline{B}$	$A\,\overline{B}$	$A\,\overline{B}+C$
0	0	0	1	0	0
0	0	1	1	0	1
0	1	0	0	0	0
0	1	1	0	0	1
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	1	0	0	1

Build the circuit:



Boolean Algebra: The Rules

Precedence:

1. brackets
2. NOT
3. AND
4. OR

For an expression with *n* inputs:

- How many rows in its truth table? 2^n
- How many possible functions? 2^{2^n}

Boolean Algebra

- **The set of elements, B, that contains at least two elements, a, b where $a \neq b$**
- **The binary operations {AND, OR}, also written as $\{\cdot, +\}$**
- **The unary operation {NOT}, also written as $\{\ \bar{}\ \}$**

Type	for a, b, c, 0, 1 ∈ B	
Closure:	$a + b \in B$	$a \cdot b \in B$
Identity:	$a + 0 = a$	$a \cdot 1 = a$
	$a + 1 = 1$	$a \cdot 0 = 0$
Commutative:	$a + b = b + a$	$a \cdot b = b \cdot a$
Distributive:	$a + (b \cdot c) = (a+b)(a+c)$	$a \cdot (b + c) = a \cdot b + a \cdot c$
Complement:	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$
Involution:	$\bar{\bar{a}} = a$	
Idempotent:	$a + a = a$	$a \cdot a = a$
Associative:	$a + (b + c) = (a + b) + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c$
Absorption:	$a + a \cdot b = a$	$a \cdot (a + b) = a$
de Morgan's Law:	$\overline{(a + b)} = \bar{a} \cdot \bar{b}$	$\overline{a \cdot b} = \bar{a} + \bar{b}$
Simplification Laws	$a + ab = a$	$a\bar{b} + b = a + b$

For Friday (pre-lab):

- Please Simplify:

$$F = \left(P\overline{R} + R\right)\left(PR + \overline{P}Q + QR\right)$$

- You should show ALL the steps and indicate which Boolean Algebra rule was used at each step.
- Draw the circuit, before & after simplification.