CSC 411

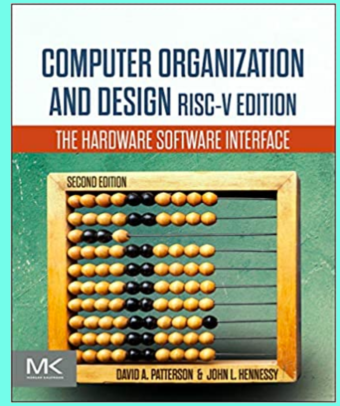
Computer Organization (Spring 2024)
Lecture 19: Introduction to logic design

Prof. Marco Alvarez, University of Rhode Island

Disclaimer

Some figures and slides are adapted from:

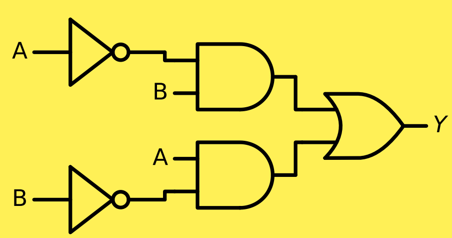
- Computer Organization and Design (Patterson and Hennessy)
  - The Hardware/Software Interface



Practice

- Which gate is this?
  - not, or, and, xnor, xor, nand, nor

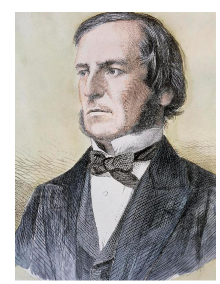
A	B	Y
0	0	
0	1	
1	0	
1	1	



Boolean algebra

- Uses 1s and 0s
- Defines properties, laws and theorems
  - useful for manipulating and simplifying boolean expressions
  - critical for digital design

George Boole (1815–1864) was an English mathematician, philosopher, and logician. He is best known as the author of The Laws of Thought (1854) which contains Boolean algebra, laying the foundations for the Information Age.



https://en.wikipedia.org/wiki/George_Boole

The following slides are just a brief introduction to boolean algebra, not covering all topics in the field, there are many more complex concepts and applications that are beyond the scope of this brief overview.

Axioms

identity	$1x$	$=$	x
	$x + 0$	$=$	x
complementary	xx'	$=$	0
	$x + x'$	$=$	1
commutative	xy	$=$	yx
	$x + y$	$=$	$y + x$
distributive	$x(y + z)$	$=$	$xy + xz$
	$x + yz$	$=$	$(x + y)(x + z)$
associative	$(xy)z$	$=$	$x(yz)$
	$(x + y) + z$	$=$	$x + (y + z)$

https://introcs.cs.princeton.edu/java/71boolean/

Deriving laws

negation	$0'$	$=$	1
	$1'$	$=$	0
double negation	$(x')'$	$=$	x
annihilation	$0x$	$=$	0
	$1 + x$	$=$	1
absorption	$x(x + y)$	$=$	x
	$x + xy$	$=$	x
DeMorgan's laws	$(xy)'$	$=$	$x' + y'$
	$(x + y)'$	$=$	$x'y'$

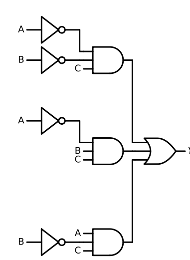
https://introcs.cs.princeton.edu/java/71boolean/

Sum of products

- Starting from a truth table …
  - any boolean function can be represented with an expression that uses AND, OR, and NOT operators
- a.k.a. DNF (disjunctive normal form)

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

$$Y = \overline{A}\,\overline{B}C + \overline{A}BC + A\overline{B}C$$

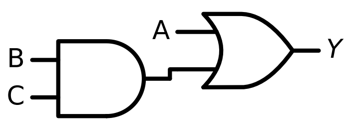


always two-level logic

Simplifying sums of products

$$\begin{aligned} Y &= \overline{A}BC + A\overline{B}\,\overline{C} + A\overline{B}C + AB\overline{C} + ABC \quad \text{canonical form} \\ &= \overline{A}BC + A\overline{B}(\overline{C} + C) + AB(\overline{C} + C) \\ &= \overline{A}BC + A\overline{B} + AB \\ &= \overline{A}BC + A(\overline{B} + B) \\ &= \overline{A}BC + A \\ &= BC + A \quad \text{minimal form} \end{aligned}$$

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



Boolean algebra also defines product of sums, i.e., expressing a function as the product of sums of literals. Both sums of products and products of sums are canonical forms used to simplify and represent Boolean expressions.

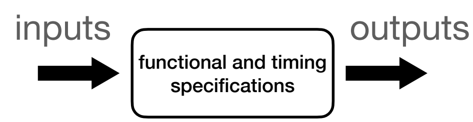
Practice

- Simplify this function

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

Building logic circuits

Logic circuits

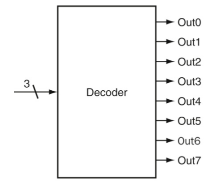


- Composition
  - inputs and outputs
  - functional specification
    - mapping between inputs and outputs (same inputs produce same outputs every time)
    - represented by (usually simplified) boolean expressions, different expressions lead to different design (hardware area, cost, latency, power, etc.)
  - timing specification
    - latency between inputs changing and outputs responding
- Types
  - **Combinational logic**
    - elements operate on data, output is a function of the inputs, memoryless
  - **State (sequential) logic**
    - elements store information, outputs determined by memory and current inputs

Decoders

Decoder

- Input pattern detector
  - combinational logic circuits that convert binary information from n input lines to a maximum of 2^n unique output lines
  - only one of the output lines is active (equal to 1) at any given time, while all the others are 0
  - decoders are used for tasks like memory addressing, or interpreting opcodes



a. A 3-bit decoder

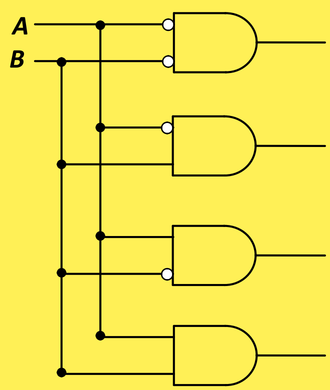
Inputs			Outputs							
12	11	10	Out7	Out6	Out5	Out4	Out3	Out2	Out1	Out0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

b. The truth table for a 3-bit decoder

Practice

- Complete the following truth table

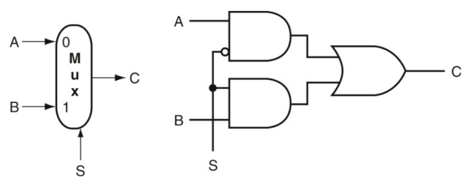
A	B	Y1	Y2	Y3	Y4
0	0				
0	1				
1	0				
1	1				



Multiplexers

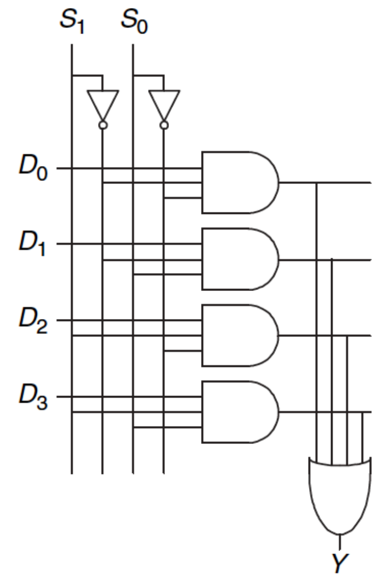
Multiplexer (MUX)

- Selector
  - combinational logic circuit , a device that selects one of several input signals and forwards it to a single output line
  - selection of the input is controlled by a set of selection lines, which determine which input gets transmitted to the output
  - commonly used in data selection, address decoding, etc.



2-to-1 mux

4-to-1 mux

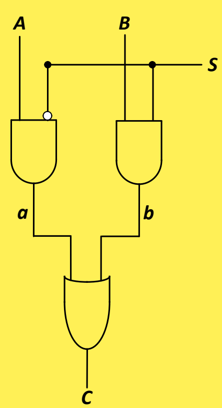


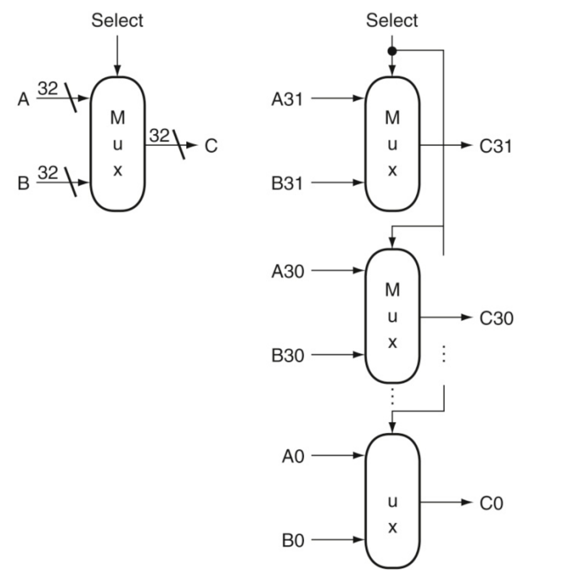
Credit: Digital Design & Computer Architecture, Our Mutlu, ETH, 2023

Practice

- Complete the following truth table

S	A	B	Y





a. A 32-bit wide 2-to-1 multiplexor

b. The 32-bit wide multiplexor is actually an array of 32 1-bit multiplexors