CS 250: Computer Architecture

Final Exam

Spring 2025

Benjamin Lobos Lertpunyaroj

May 8th, 10:30AM – 12:30PM

Exam contents and details for referencing

- Final exam is held in Fowler Hall on May 8th (Thursday), from 10:30 AM to 12:30 PM.
- Previous cumulative book chapters
  - Chapter 1 sections 1, 2, and 3.
  - Chapter 2, sections 1, 2, 3, 4, 5, 6, and 7.
  - Chapter 3, sections 1, 2, and 5.
  - Chapter 4, sections 1, 2, 3, 4, 5, 6, 7, and 8.
  - Chapter 5, sections 1, 2, 3, 4, 7, and 8.
  - Chapter 8 (Appendix A), sections 1, 2, 3 (but not PLAs or ROMs), 5, 7 (lightly), and 8.
- All lecture notes and lecture slides.
- All labs (1 - 11).

Appendix A

Logic & Gates

An *asserted* signal is logically true, the *deasserted* is the opposite.

Two types of logic systems

Combinational logic: No memory in components, hence same output given same input.

Sequential logic: Memory in components, hence output depends on input and current memory state.

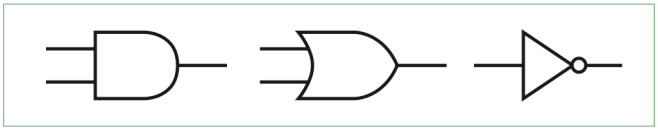


Figure 1: *AND gate, OR gate, and inverter*

The gates can be combined to form different forms of logic. An example of this is $\overline{\overline{A} + B}$ which is equivalent to $A \cdot \overline{B}$ by De Morgan's law, seen in Figure 2.

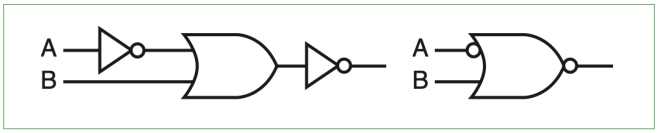
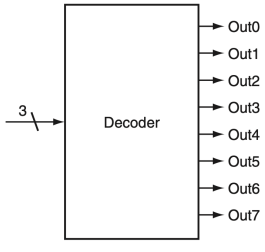


Figure 2: *Logic gate implementation of example formula*

Decoders & Multiplexors

A **decoder** is a logic block that has an n-bit input and 2^n outputs, where there is one unique true bit as output from a unique set of bytes of input.



a. A 3-bit decoder

Inputs			Outputs							
I2	I1	I0	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

Figure 3: *3-bit input decoder that generates* $2^3 = 8$ *different outputs (Out0 – Out7)*

$$2^n \text{ outputs } \therefore \log_2(\text{output}) = \text{input bits}$$

Encoders are the other way around.

Multiplexors have a selector input (or control value), that will determine which inputs will become outputs.

In the case of the two-input MUX, its representation is the following, $C = (A \cdot \overline{S}) + (B \cdot S)$, using n (data inputs) AND gates, and one OR gate.

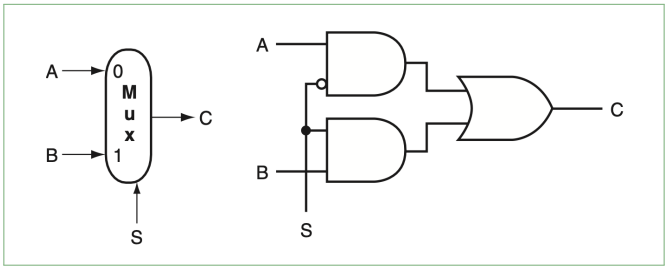


Figure 4: *Two-input multiplexor that generates one output depending on the selector input S*

$$n \text{ (data inputs)} \therefore \log_2 n = S \text{ selector bits required to represent all inputs}$$

Often times a decoder generates n bits for a MUX, to be used as a selector signal.