ECE 370: Digital Systems-Logic Design

Sample Test: Chapter 6-Part 2: Multiplexers
Spring 2005

Multiplexers

1. Describe the general purpose of a multiplexer. Generally, how many inputs does it have? Generally, how many outputs does it have? What sort of electrical device does a multiplexer mimic?

2. Design an 8-to-1 multiplexer circuit using a minimum SOP implementation. Repeat the design using a minimum POS implementation

3. Design an 8-to1 multiplexer in the following two ways:
   a) using only 2-to-1 multiplexers.
   b) using only 4-to-1 multiplexers.

4. Design a 16-to1 multiplexer in the following three ways:
   a) using only 2-to-1 multiplexers.
   b) using only 4-to-1 multiplexers.
   c) using only 8-to-1 multiplexers.

5. Design a 4-bit 2's complement ripple-carry Adder/Subtractor system, this time using multiplexers and a 4-bit adder component. Repeat the design, this time designing a 4-bit 1's complement ripple carry Adder/Subtractor.

6. Implement the Boolean function below using the algebraic definition of *Shannon's Expansion Theorem*

$$f(A,B,C) = \overline{A}B\overline{C} + BC + A\overline{B}$$

   a) using a 2-to-1 MUX, any extra logic gates, and *B* as the select input to the 2-to-1 MUX.
   b) using a 4-to-1 MUX, any extra logic gates, and *B &C* as the select inputs to the 4-to-1 MUX.
   c) using a 8-to-1 MUX, any extra logic gates, and *A, B, & C* as the select inputs to the 8-to-1 MUX.

PURDUE
UNIVERSITY
CALUMET

7. Implement the Boolean function below using the algebraic definition of *Shannon's Expansion Theorem*

$$f(w,x,y,z) = \left(x\overline{y} + x\overline{z}\right)\left(\overline{z}w + y\right)\left(\overline{y} + \overline{z}\right)$$

a) using a 2-to-1 MUX, any extra logic gates, and *y* as the select input to the 2-to-1 MUX.
b) using a 4-to-1 MUX, any extra logic gates, and *x & z* as the select inputs to the 4-to-1 MUX.
c) using a 8-to-1 MUX, any extra logic gates, and *w, x, & z* as the select inputs to the 8-to-1 MUX.
d) using a 16-to-1 MUX, any extra logic gates, and *w, x, y, & z* as the select inputs to the 16-to-1 MUX.

8. Implement the Boolean function below using *Shannon's Expansion Theorem* and K-Maps.

$$f_{\varepsilon}(A,B,C) = A\overline{B}\overline{C} + A(B \oplus C) + C(\overline{A \oplus C})$$

a) using a 2-to-1 MUX, any extra logic gates, and *A* as the select input to the 2-to-1 MUX.
b) using a 4-to-1 MUX, any extra logic gates, and *C & D* as the select inputs to the 4-to-1 MUX.
c) using a 8-to-1 MUX, any extra logic gates, and *C, B, & A* as the select inputs to the 8-to-1 MUX.

9. Implement the Boolean function below using *Shannon's Expansion Theorem* and K-Maps.

$$f(A,B,C,D) = (\overline{A} + B + C)(B + C + D)(\overline{A} + D)$$

a) using a 2-to-1 MUX, any extra logic gates, and *D* as the select input to the 2-to-1 MUX.
b) using a 4-to-1 MUX, any extra logic gates, and *B & C* as the select inputs to the 4-to-1 MUX.
c) using a 8-to-1 MUX, any extra logic gates, and *D, A, & B* as the select inputs to the 8-to-1 MUX.
d) using a 16-to-1 MUX, any extra logic gates, and *D, B, A, & C* as the select inputs to the 16-to-1 MUX.

10. An Arithmetic Logic Unit (ALU) is a combinational digital system found in microprocessors whose purpose is to perform arithmetic & logical operations (*AND, OR, NOT, ADD, SUBTRACT* etc…) for the processor. Here are the input/output specs for a typical *n*-bit ALU.

- **An n-bit ALU contains two *n-bit* inputs and one *m-bit* input.**

  A: *n-bit* signed number on which to operate.

  B: *n-bit* signed number on which to operate.

  OpCode: *m-bit* code indicating 1 of the 2^m different operations performed by the ALU.

- **An n-bit ALU contains a one *n-bit* output and two single-bit outputs**

  Result: *n-bit* signed number indicating the result of a specific operation.

  Overflow: When adding or subtracting input, the ALU needs to indicate overflow,

  Zero: Indicates if the result is zero.

Design a 1-bit ALU that performs *ADD, SUB, AND, OR, NOR, and XOR* operations. Your design should have all of the inputs and outputs described above, with appropriate lengths (*Hint: assign the most appropriate opCode for each operation).* Use your 1-bit ALU design to implement a hierarchal design of a 4-bit ALU. Show your schematics for each design.