UNITED STATES AIR FORCE ACADEMY

ECE 281

Lesson 13

Notes

Objectives:

- Understand how to create an adder, subtractor, or comparator
- Know the tradeoffs of the three common carry propagate adders
- Demonstrate ability to use digital building blocks (adders, subtractors, comparators, and muxes) to build more complex combinatorial circuits

Tradeoffs of the three common carry propagate adders: In your zyBooks lesson, you were essentially presented with two common forms of an adder logic circuit:

- **Carry-Ripple Adder**
- **Carry-Lookahead Adder**

In addition to these forms, there are other more advanced forms such as the Radix-3 or Sklansky & KoggeStone Adder (This form was not covered in zyBooks, and is beyond the scope of this class).

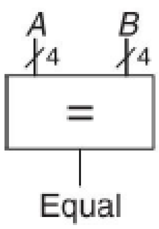
What I want you all to understand, is that your design requirements will drive your choice of logic design in many cases. For instance:

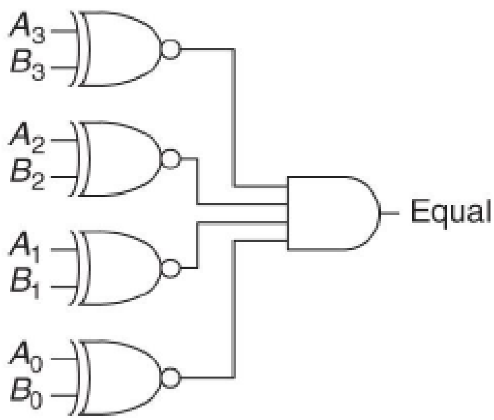
Example 1) Which adder would you use (Carry-Ripple or Carry-Lookahead) if you wanted to minimize the number of logic gates required?

Example 2) Which adder would you use (Carry-Ripple or Carry-Lookahead) if you wanted to minimize the delay from input to output?

Comparators – compares two numbers, indicating whether the numbers are equal, or which number is greater.

Equality Comparator

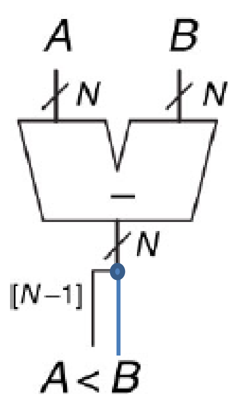




XNOR

A	B	Y
0	0	
0	1	
1	0	
1	1	

Magnitude Comparator



NOTE: get used to working with a higher level of abstraction going forward. We will increasingly start to work with blocks that pertain to a specific function (i.e. muxes, decoders, adders, comparators, etc.)

If A-B = 0:

If A-B < 0:

If A-B > 0:

Emphasis Item: The magnitude comparator will only prove A is "Less than" B without additional logic…. Why?

<u>In-Class Practice Problem:</u> Design an N-Bit comparator using the equality and magnitude comparator (along with necessary additional logic) to determine if A is less than, greater than or equal to B:

Build the same system with a single full-adder, 2x NOR gates and an Inverter:

Design Practice Problem #1: Can you implement the function below given that a, y, z, 3, 4 and 7 are all 8-bit numbers?

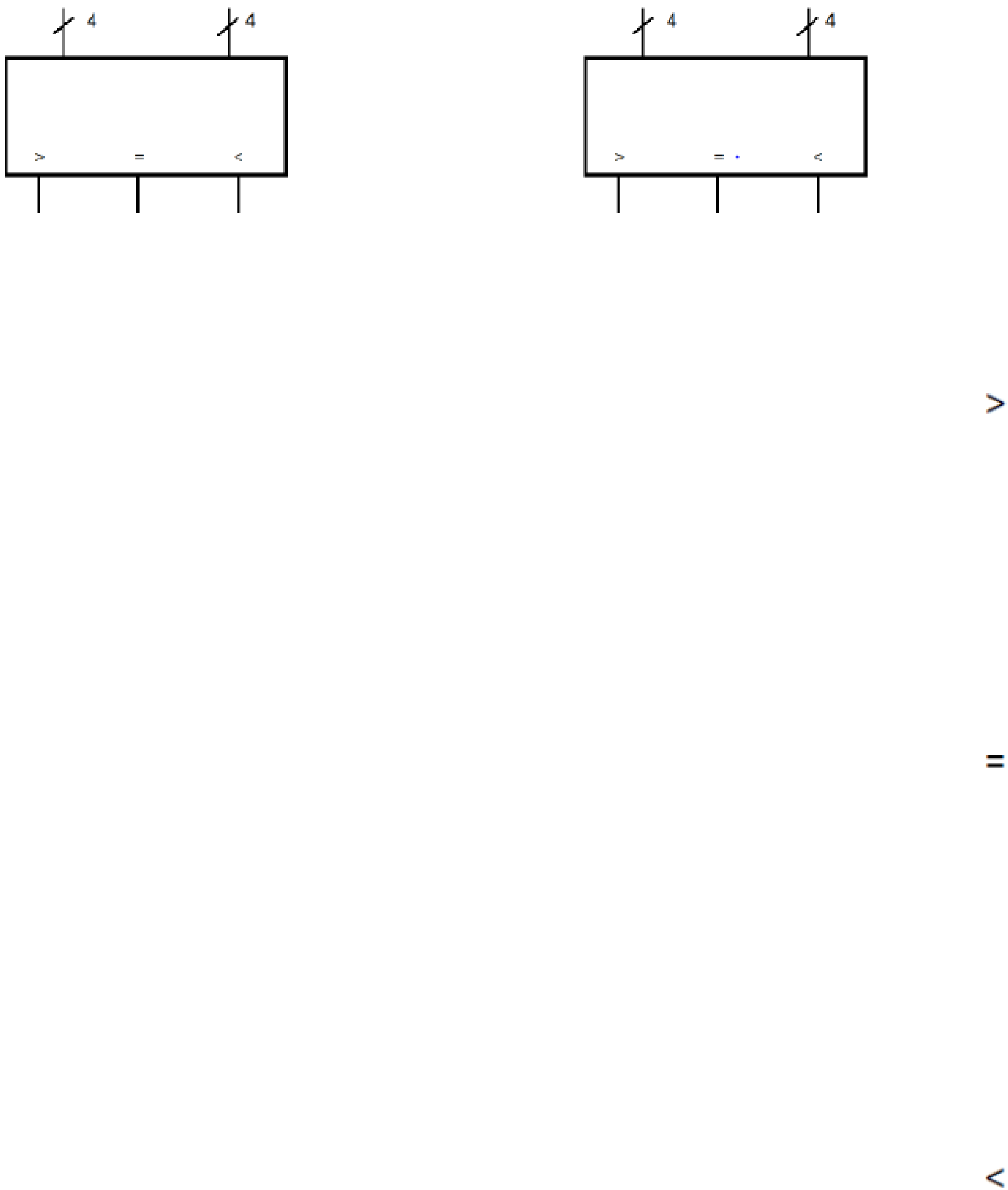
If a<4:

z=y+3

Else:

z=y+7

Design Practice Problem #2: Given two four-bit magnitude comparators, add some combinational logic gates and create an eight-bit magnitude comparator where A and B are your 8-bit inputs to compare. Ensure the inputs (and associated bits are clearly labeled). Note that the outputs for the 8-bit comparator are given on the right.



Design Practice Problem #3: Given the following adder, add combinational logic to determine whether the addition overflowed assuming 2's-complement addition. (Hint: refer back to page 4 of lesson 2 notes). Then, what simple addition could you make to the circuit below to detect either signed or unsigned overflow given a control signal "Sign"