Tyler Holland

Suzie Morano

**Experiment 7**

**Procedure 1: Half Adder**

Truth Table: Black Box:

|  |  |  |  |
| --- | --- | --- | --- |
| ai | bi | **si** | **ci** |
| 0 | 0 | **0** | **0** |
| 0 | 1 | **1** | **0** |
| 1 | 0 | **1** | **0** |
| 1 | 1 | **0** | **1** |

Expressions:

Sum (si) = Ai XOR Bi

CarryOut (ci) = Ai AND Bi

**Procedure 2: Full Adder**

Truth Table: Black Box:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ai | Bi | Ci - 1 | **Si** | **ci** |
| 0 | 0 | 0 | **0** | **0** |
| 0 | 0 | 1 | **1** | **0** |
| 0 | 1 | 0 | **1** | **0** |
| 0 | 1 | 1 | **0** | **1** |
| 1 | 0 | 0 | **1** | **0** |
| 1 | 0 | 1 | **0** | **1** |
| 1 | 1 | 0 | **0** | **1** |
| 1 | 1 | 1 | **1** | **1** |

Expressions:

Sum (si) = Ai XOR Bi XOR Ci - 1

CarryOut (ci) = (Ai AND Ci - 1) OR (Ai AND Bi) OR (Bi AND Ci – 1)

**Procedure 3:**

Demo 4bitRCA: \_\_\_\_\_\_\_\_\_\_

**Procedure 4:**

Demo 4BitComparator: \_\_\_\_\_\_\_\_\_\_\_

Truth Table: For 2 bit comparator

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A1 | A0 | B1 | B0 | **Output** |
| 0 | 0 | 0 | 0 | **1** |
| 0 | 0 | 0 | 1 | **0** |
| 0 | 0 | 1 | 0 | **0** |
| 0 | 0 | 1 | 1 | **0** |
| 0 | 1 | 0 | 0 | **0** |
| 0 | 1 | 0 | 1 | **1** |
| 0 | 1 | 1 | 0 | **0** |
| 0 | 1 | 1 | 1 | **0** |
| 1 | 0 | 0 | 0 | **0** |
| 1 | 0 | 0 | 1 | **0** |
| 1 | 0 | 1 | 0 | **1** |
| 1 | 0 | 1 | 1 | **0** |
| 1 | 1 | 0 | 0 | **0** |
| 1 | 1 | 0 | 1 | **0** |
| 1 | 1 | 1 | 0 | **0** |
| 1 | 1 | 1 | 1 | **1** |

Equations for 4-bit comparator

EQUAL = (NOT(A0) OR B0) **AND** (NOT(A1) OR B1) **AND** (A0 OR NOT(B0)) **AND** (A1 OR NOT(B1))

**Questions:**

1. The propagation delay for the half adder is 1 AND gate or 1 XOR gate.
2. The propagation delay for the full adder is either 2 XOR gates or 1 AND gate and 1 OR gate
3. 256 rows for a 4-bit adder. There are 8 inputs and 5 outputs. No, because 232 is much too large to make a practical truth table out of.
4. 20 gates. The worst case delay path is having the carry from each adder affect the next adder. The values that cause this are 1111 for A and 0001 for B.
5. 44 gates. Equation: (((# of bits) -1) \* 6) + 2.
6. Use a 2’s complement design and use all Full adders instead of Full adders and one half adder. There would also be a subtraction input that is hooked to each B input with an XOR gate with the original B input. This creates the 2’s complement for each input, thus subtracting the values when they are added.
7. It has a carry out value, so you have to add an extra check to make sure the answer is right.
8. Structural model, we used two 2-bit comparators in the architecture of the 4-bit comparator.
9. The keypad password input. This device would check to make sure the given password is equal to the stored password in the alarm system.

**Conclusion**:

Tyler: In this lab I learned how to make a Binary adder using VHDL and components in VHDL. The parts that gave me trouble was figuring out the best way to implement the adder using the smallest number of gates and getting a nearly glitch free and smallest delay device as possible. It was nice applying what we had learned in CPE 129 into our lab and seeing how it actually works on the NEXYS board.

Suzie: In Today’s lab we used VHDL to create an adder and a comparator. We were able to create less redundant code by using a structural model. We only had to write code for one full adder even though we used 3 full adders. Using this type of coding we were also able to use a 2 bit comparator to make a 4 bit comparator.