The following is an except for ECED2200 Lab #4. See www.colinoflynn.com/teaching

# Part 1 Observations

**Part 1-A: NOR Based RS Latch**

**Q1)** Complete the state transition table. It may take some experimentation to get from one state to the next. Where two lines are given for the same transition there may be more than one method of getting that transition to occur. The state transition table shows how we move from the state given by Q to the state given by Q+.

**NB: See detailed information in the procedure if you don’t understand what the state transition tables are asking, it IS NOT simply the state of LED1/LED2!!**

|  |  |  |  |
| --- | --- | --- | --- |
| SW1 | SW2 | Q (LED1 Initial State) | Q+ (LED1 Final State) |
|  |  | 0 | 0 |
|  |  | 0 | 0 |
|  |  | 0 | 1 |
|  |  | 1 | 0 |
|  |  | 1 | 1 |
|  |  | 1 | 1 |

**Q2)** What are the inputs SW1 and SW2 (which is Reset & Set). Are they active high or active low? What is the invalid state in this flip-flop (e.g.: SW1=?, SW=?) and in that invalid state what are the outputs?

**Part 1-B: NAND Based RS Latch**

**NB: See detailed information in the procedure if you don’t understand what the state transition tables are asking, it IS NOT simply the state of LED1/LED2!!**

**Q1)** Complete the state transition table. It may take some experimentation to get from one state to the next. Where two lines are given for the same transition there may be more than one method of getting that transition to occur. The state transition table shows how we move from the state given by Q to the state given by Q+.

|  |  |  |  |
| --- | --- | --- | --- |
| SW1 | SW2 | Q (LED1 Initial State) | Q+ (LED1 Final State) |
|  |  | 0 | 0 |
|  |  | 0 | 0 |
|  |  | 0 | 1 |
|  |  | 1 | 0 |
|  |  | 1 | 1 |
|  |  | 1 | 1 |

**Q2)** What are the inputs SW1 and SW2 (which is Reset & Set). Are they active high or active low? What is the invalid state in this flip-flop (e.g.: SW1=?, SW=?) and in that invalid state what are the outputs.

**Part 1-C: JK Flip-Flop**

**NB: See detailed information in the procedure if you don’t understand what the state transition tables are asking, it IS NOT simply the state of LED1/LED2!!**

**Q1)** Complete the state transition table. It may take some experimentation to get from one state to the next. Where two lines are given for the same transition there may be more than one method of getting that transition to occur. The state transition table shows how we move from the state given by Q to the state given by Q+. Note for the JK flip-flop you must use the CLOCK line to cause the state transition to occur. So the procedure is:

1. The current state of LED1 is the value of Q, thus the current state.
2. Set SW1/SW2 (J/K) to the desired value. Note unlike the Part 1-A or Part 1-B the circuit will do nothing until you pulse SW3, thus it doesn’t actually notice changes in the input switches until you pulse SW3 described below.
3. Pulse SW3 (CLK) to 1 then back to 0, and observe the transition that occurred.

|  |  |  |  |
| --- | --- | --- | --- |
| SW1 (J) | SW2 (K) | Q (LED1 Initial State) | Q+ (LED1 Final State) |
|  |  | 0 | 0 |
|  |  | 0 | 0 |
|  |  | 0 | 1 |
|  |  | 0 | 1 |
|  |  | 1 | 0 |
|  |  | 1 | 0 |
|  |  | 1 | 1 |
|  |  | 1 | 1 |

**Q2)** On which clock edge is this flip-flop changing state? E.g. does the output changes when you move the CLK (SW3) from 0🡪1 (Falling Edge) or 1🡪0 (Rising Edge)?

#

# Part 2 Observations

**Part 2-A**

Q1)Play with the switches SW1 & SW2. What does changing the setting of the switches do to the count sequence you observe?

Q2) Fill out the following table:

|  |  |
| --- | --- |
| Switch Settings | Count Sequence |
| SW1 | SW2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 1 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Part 2-B**

Q1) Set SW1 = 1, SW2 = 0. This should result a count sequence of 0,1,2,3,…,d,e,f. What is the sequence now?

|  |  |
| --- | --- |
| Switch Settings | Count Sequence |
| SW1 | SW2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 1 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Q2) FD4RE has a synchronous reset, that is to say it is only valid on the rising clock edge. FD4CE has an asynchronous clear; that is to say as soon as it goes high, even for an instant, the output is cleared. Why do you think the asynchronous clear is giving us problems in this implementation?