3 SEQ2 WIN STATE MACHINE.

This section will provide some insight into the first of 4 slave state machines, the “Win State Machine”. The real work of the Tic Tac Toe program is performed with the slave state machines.

The Win State Machine scans through the board looking for an opportunity for the PLC to win by playing a single square. You can think of this as pattern recognition, but I think of it as brute force ladder logic.

3.1 SEQ2 WIN STATE MACHINE DIAGRAM.

Figure 7 Win State Machine Diagram provides is the thing that you need to look at. This state machine is slave the the main state machine and simply scans the game board for an opportunity for the PLC to win the game.

The Win State Machine uses the COUNTER, Seq2, to manage its state. You will notice that the RELAY, RxCxChkWin is used to control 3 transitions:

1. Transition from Seq2:0 to Seq2:1 when the RELAY, RxCxChkWin, is asserted. This RELAY is controlled by the main state machine. This RELAY controls the starting of the state machine.
2. Transition from Seq2:8 to Seq2:0 when the RELAY, RxCxChkWin, is de-asserted. Seq2:8 is reached after all tests to “see” if the PLC can win have failed. While in Seq2:8 a RELAY is asserted that feeds back to the main state machine indicating that no winning move could be found. In response to this RELAY, the main state machine will de-assert RxCxChkWin.
3. Transition from Seq2:31 to Seq2:0 when the RELAY, RxCxChkWin, is de-asserted. Seq2:31 is reached if this state machine found a way to win and while in this state a RELAY is asserted that feeds back to the main state machine indicating that the winning move has been made.

You should, also, notice that the transitions from Seq2:1 through Seq2:9 have no labels to indicate when these transitions are valid. I was being lazy and didn’t clutter the state diagram with text. States Seq2:1 through Seq2:8 test for the possibility of a win for an indivdual game square. If a test did result in a Win, then the state machine transitions to State 31, Seq2:31. If If the test does not result in a Win, then the state machine advances to the next state, say form Seq2:1 to Seq2:2. The PLC code that implements the state transition rules ensures that all of this happens in an orderly fashion and if there isn’t a square that the PLC can take to win the game the state machine will work its way done to Seq2:9

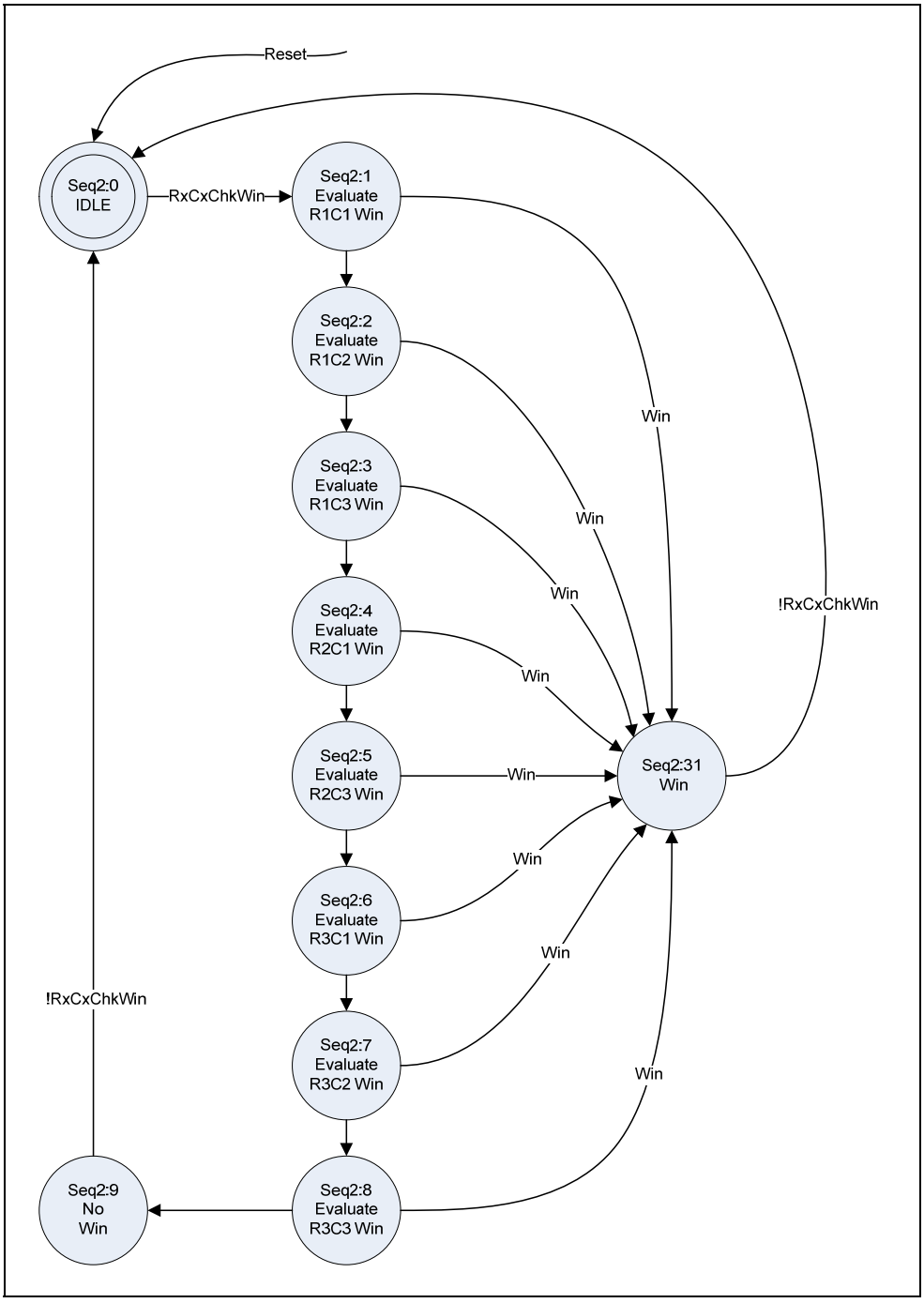

Figure 7 Win State Machine Diagram

3.2 SEQ2 WIN STATE MACHINE TRANSITION RULES

Figure 8- Win State Transition Rules is the ladder logic implementation. You should notice that the ladder logic implementation is much simpler for the Win State Machine than what was done for the Main State Machine. The simpler implementation took advantage of two things:

1. The [AVseq] mechanism was used to simply increment the Seq2 counter to go to the next state when the previous state did not result in a “win”.
2. . The [AVseq] ladder logic requires a minimum of 2 ladder logic scans to detect a rising edge condition. Each state transition from Seq2:1→Seq2:9 must be maintained for one full scan time. The [AVseq] mechanism actually forces the state machine to remain in states 2:1 through 2:9 for 2 full scans. This is just fine. The RELAY, FastClk, ensures that the timing requirements are met for the use of [AVseq].

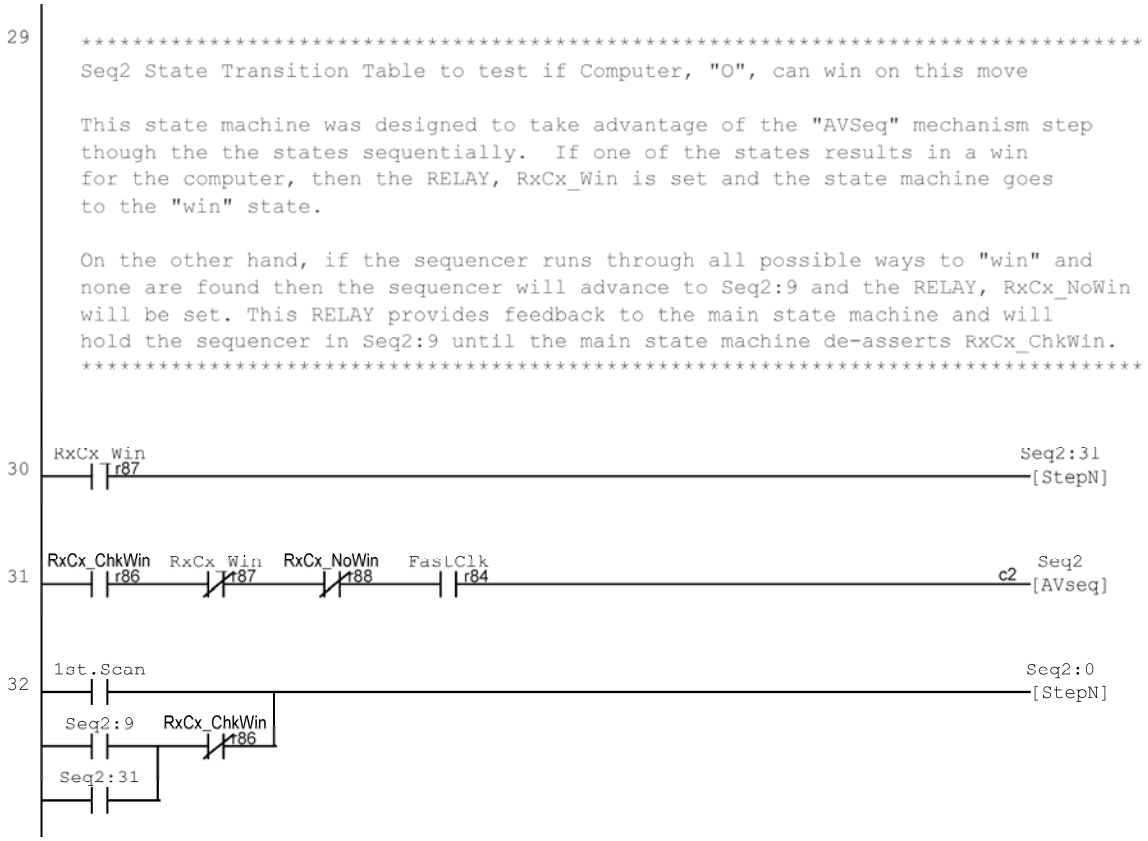

Figure 8- Win State Transition Rules

3.2.1 What happens On Each State?

The state transition rules were handled in the previous section. Now we will look at what happens on a state by state basis. Please refer to to Figure 9 Win State Machine Hardware Actions.