**Lab 2 Report**

***g14\_longest\_row.vhd***

**Top-level I/O**

**Inputs Outputs**

ASP\_r\_0 ... ASP\_r\_7 – ASP\_arrays ASP\_lrow : 2 bit std\_logic\_vector

ASP\_dr – 7 bit vector

ASP\_empty – 7 bit vector

NM – bit

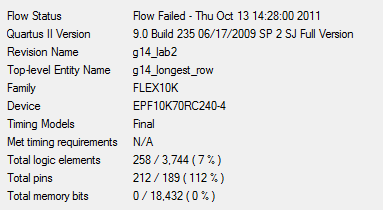
**NB: the ASP\_array data type is a 4 by 3 array of std\_logic\_vector**

**Description of Circuit**

This circuit takes in 8 registers of the type ASP\_array, which contain the current state of the filled positions on the connect-4 board. The 7 bit vector ASP\_dr reflects the states of the registers as to whether they are full or not full at present. The 7 bit vector ASP\_empty reflects the states of the registers as to whether they are empty or not at present. NM is an enable bit.

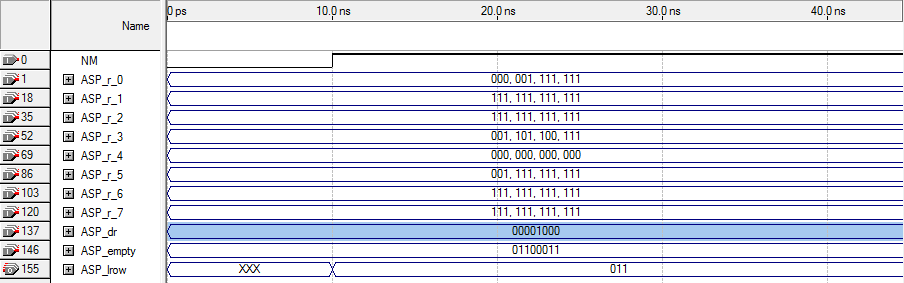
The circuit evaluates the non-empty and non-full registers, searching for the register which contains the longest row of filled cells. Once all the registers have been parsed, the output ASP\_lrow returns a 2 bit vector containing the ID of the register with the longest row.

Flow summary and timing analyzer summary could not be completed on this circuit due to the specified inputs and outputs requiring 212 pins, while the device which the circuit was designed for has only 189 I/O pins. Thus, Quartus II could not run the flow analysis necessary to generate these reports. However, it showed that 258 logic elements were required for the circuit.



**Testing**

The circuit was tested by creating sample situations by assigning the register values arbitrary numbers and then adjusting ASP\_dr and ASP\_empty to correspond to the register values. This was done with various setups of arbitrary register values. As shown in the case included, registers 1, 2, 6, and 7 are empty, register 4 is full, and the longest register counted is register 3. The circuit operates correctly, as “001” is returned for ASP\_lrow, indicating that register 3 is indeed the longest row.



***g14\_row\_config.vhd***

**Top-level I/O**

**Inputs Outputs**

ASP\_r\_i – type ASP\_array horiz -bit

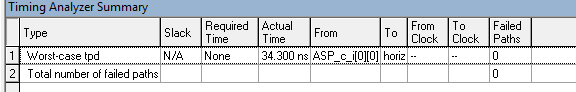
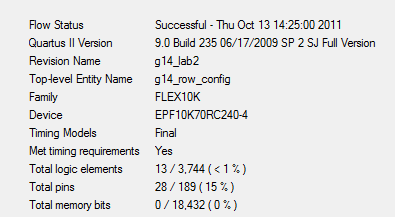
ASP\_c\_i – type ASP\_array vetic -bit

r\_l\_diag – bit

l\_r\_diag – bit

**Description of Circuit**

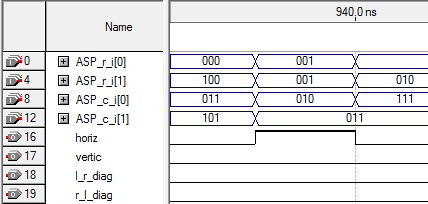
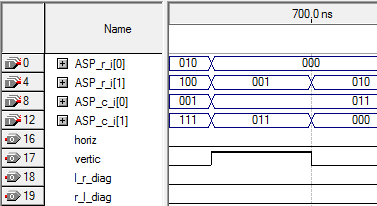
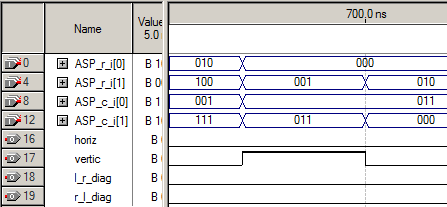
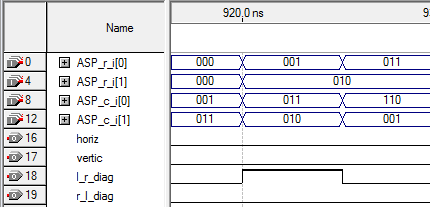
This circuit takes in two registers (ASP\_r\_i and ASP\_c\_i) of type ASP\_array, again 4x3 std\_logic\_vector arrays, which represent the row and column registers for the connect-4 board. The circuit then checks the first two positions of each register (0 and 1) to search for filled cells in horizontal, vertical, and both diagonal configurations. If a configuration is found, its corresponding output becomes true.

Flow analysis was successful for this circuit and showed that 13 logic elements and 28 I/O pins were required for the implementation of this circuit. Timing analyzer showed that the worst-case propogation delay of the circuit was 34.3 ns.  


**Testing**

The circuit was tested by applying random values to the register locations at ASP\_r\_i and ASP\_c\_i at indices 0 and 1, as they were the only ones checked by the circuit. Each possible output case was identified and is discussed as follows.

1. Horizontal is found when ASP\_r\_i[1] = “111”
2. Horizontal is found when r[0] = r[1] and c[0] + 1 = c[1]
3. Vertical is found when r[0] + 1 = r[1] and c[0] = c[1]
4. L\_R\_diag is found when r[0] + 1 = r[1] and c[0] = c[1] + 1
5. R\_L\_diag is found when r[0] + 1 = r[1] and c[0] + 1 = c[1]

** **

