Group 14 -
Jerome Colomb 260349913
Matthew Johnston 260349319

Lab 2 Report

g14_longest_row.vhd

Top-level I/O

Inputs

ASP_r_0 ... ASP_r_7 – ASP_arrays

ASP_dr – 7 bit vector

ASP_empty – 7 bit vector

NM – bit

Outputs

ASP_lrow : 2 bit std_logic_vector

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.

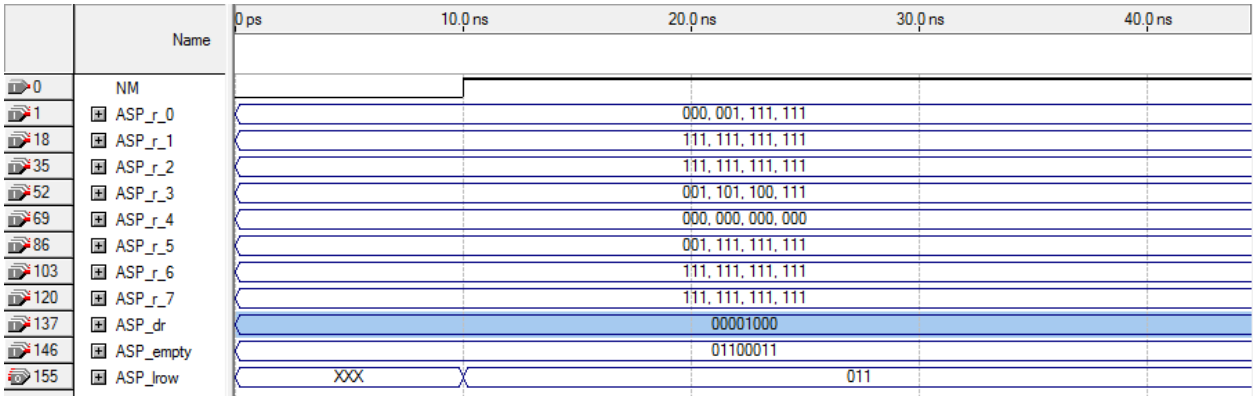
Flow summary and timing analyzer summary could not be completed on this circuit due to the specified inputs and outputs requiring 212 pins, while the device which the circuit was designed for has only 189 I/O pins. Thus, Quartus II could not run the flow analysis necessary to generate these reports.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

Group 14 -
Jerome Colomb 260349913
Matthew Johnston 260349319

Flow Status	Flow Failed - Thu Oct 13 14:28:00 2011
Quartus II Version	9.0 Build 235 06/17/2009 SP 2 SJ Full Version
Revision Name	g14_lab2
Top-level Entity Name	g14_longest_row
Family	FLEX10K
Device	EPF10K70RC240-4
Timing Models	Final
Met timing requirements	N/A
Total logic elements	258 / 3,744 (7 %)
Total pins	212 / 189 (112 %)
Total memory bits	0 / 18,432 (0 %)

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.



Group 14 -
Jerome Colomb 260349913
Matthew Johnston 260349319

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.

Flow Status	Successful - Thu Oct 13 14:25:00 2011
Quartus II Version	9.0 Build 235 06/17/2009 SP 2 SJ Full Version
Revision Name	g14_lab2
Top-level Entity Name	g14_row_config
Family	FLEX10K
Device	EPF10K70RC240-4
Timing Models	Final
Met timing requirements	Yes
Total logic elements	13 / 3,744 (< 1 %)
Total pins	28 / 189 (15 %)
Total memory bits	0 / 18,432 (0 %)

Group 14 -
Jerome Colomb 260349913
Matthew Johnston 260349319

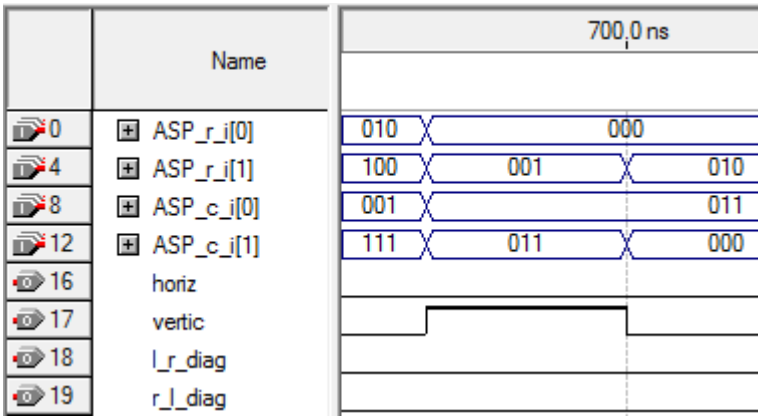
Timing Analyzer Summary

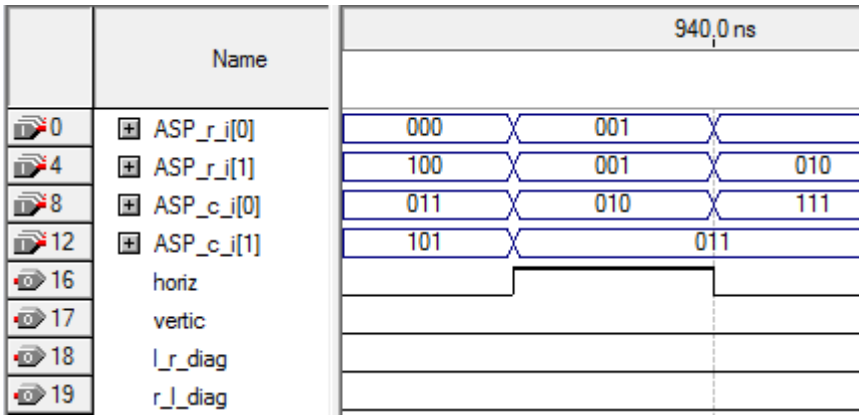
	Type	Slack	Required Time	Actual Time	From	To	From Clock	To Clock	Failed Paths
1	Worst-case tpd	N/A	None	34.300 ns	ASP_c_i[0][0]	horiz	--	--	0
2	Total number of failed paths								0

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]





Group 14 -
Jerome Colomb 260349913
Matthew Johnston 260349319