Compsys 701 Assignment 1

Name: Arjun Kumar, UPI : akmu999, Student ID : 432595724

The FSM_tb test bench file contains the description of all the components and their port maps along with the test inputs. I believe this is a similar to a top level file as it has all the components in it port mapped.

Task 1 Multiplier Implementation

The multiplier has been implement with a testbench. The multiplier takes two 8 bit inputs and multiplies them to give a 16 bit number. I later on in the vhdl code add an extra bit to account for overflow that may occur when these numbers are added using the adder. I use the following statement

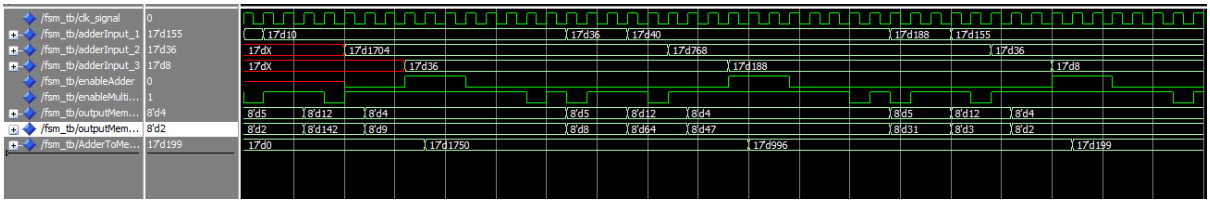
output <= unsigned ('0'& std_logic_vector(multiplicationResult));

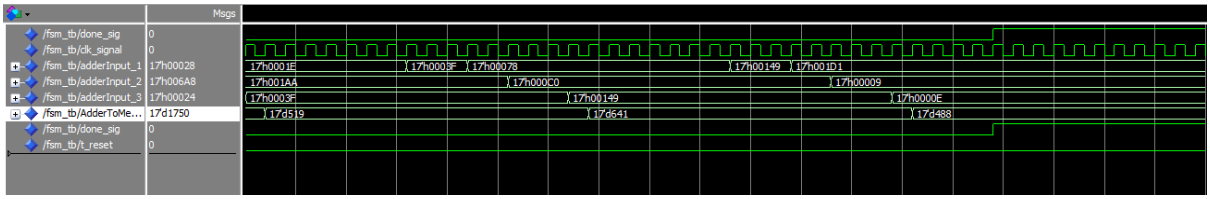
Task 2: Matrix multiplication FSM 1

For Task 2- I implemented the matrix multiplication using 3 registers to store the output from the multiplier. In order to select which register should store the output I used a Mux. Afterwards all the 3 values from the registers were added together using an adder and then stored in the Memory.

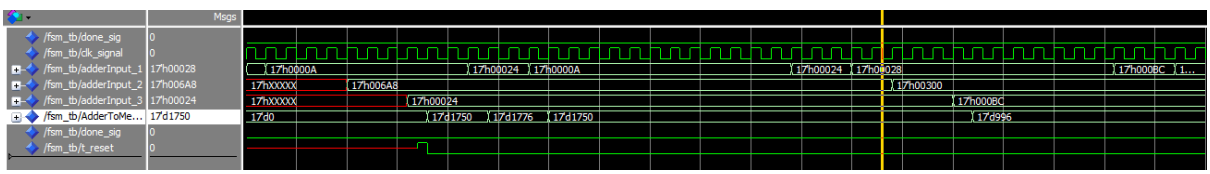
The following Model Sim simulation shows the progressing matrix multiplication
For the Model Sim simulation, select the clk_signal, AdderToMem, enableMultiplier, adderinput_1, aderInput_2, adderInput_3.

I used a 4bit counter to keep track of the cycles. This helped me in implementing the FSM as now I could keep track of which signals should be enabled in a given clock cycle. There is always a delay when the signal is applied and when it actually takes effect. Using the if statements and the counter values I was able to implement the clock signals for one state. Once I had one state working properly I repeated the code for all the other states. This is certainly not the best way but it made it easier for me to debug the FSM.



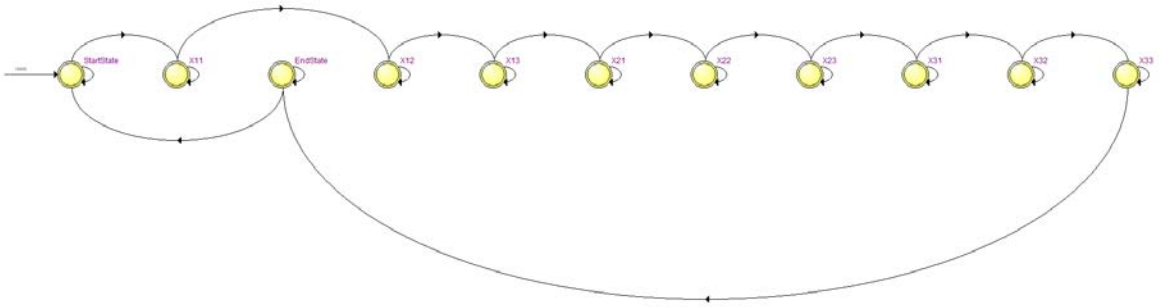


The second last signal in the above screen shot shows the done signal when the last calculation has been completed



The above screenshot show a reset signal restarting the process. The last signal is the restart signal. The same values for the adder reappear after the reset signal has been switched off.

FSM Diagram



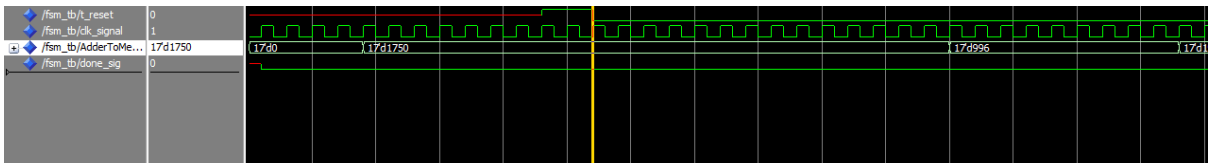
I have used 9 states for the calculation of 9 different elements along with the start and End state.

Synthesis Results for First FSM

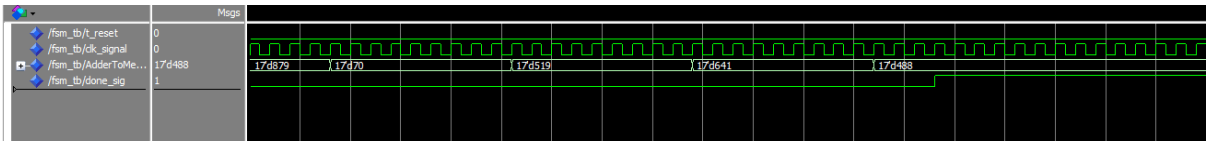
Flow Summary	
Flow Status	Successful - Fri Mar 16 19:24:29 2018
Quartus II 64-Bit Version	13.0.0 Build 156 04/24/2013 SJ Full Version
Revision Name	MatrixMultiplication
Top-level Entity Name	FSM
Family	Cyclone V
Device	5CGXFC7C7F23C8
Timing Models	Preliminary
Logic utilization (in ALMs)	39 / 56,480 (< 1 %)
Total registers	11
Total pins	40 / 268 (15 %)
Total virtual pins	0
Total block memory bits	0 / 7,024,640 (0 %)
Total DSP Blocks	0 / 156 (0 %)
Total HSSI RX PCSs	0 / 6 (0 %)
Total HSSI PMA RX Deserializers	0 / 6 (0 %)
Total HSSI PMA RX ATT Deserializers	0
Total HSSI TX PCSs	0 / 6 (0 %)
Total HSSI PMA TX Serializers	0 / 6 (0 %)
Total HSSI PMA TX ATT Serializers	0 / 6 (0 %)
Total PLLs	0 / 13 (0 %)
Total DLLs	0 / 4 (0 %)

Task 3: Matrix multiplication FSM 2

The task 3 implementation involved writing an entity to split the 24 bit number into 3 values. These values are then passed to the 3 multipliers. The 3 multipliers then compute the numbers. These 3 values are directly added using an adder and then written into the Memory.



The reset signal being applied for Task 3. The reset signal is the top signal on the waveform.



The done signal activated at the end of the completion of the calculation.