Group 54

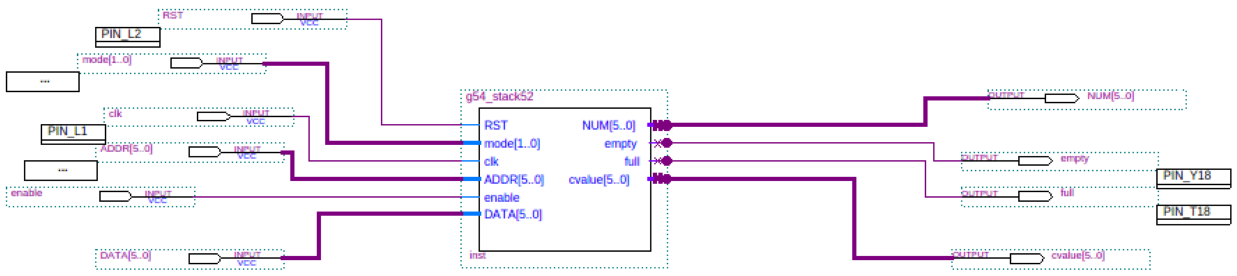
Eva Suska 260618967
Marcel Morin 260605670

g54_stack52 - Lab 3

Description

Input: 1 bit reset, 1 bit enable, 3 bit mode, 6 bit address, 6 bit data, clock

Output: 1 bit empty, 1 bit full, 6 bit number of stack elements, 6 bit card value Input:



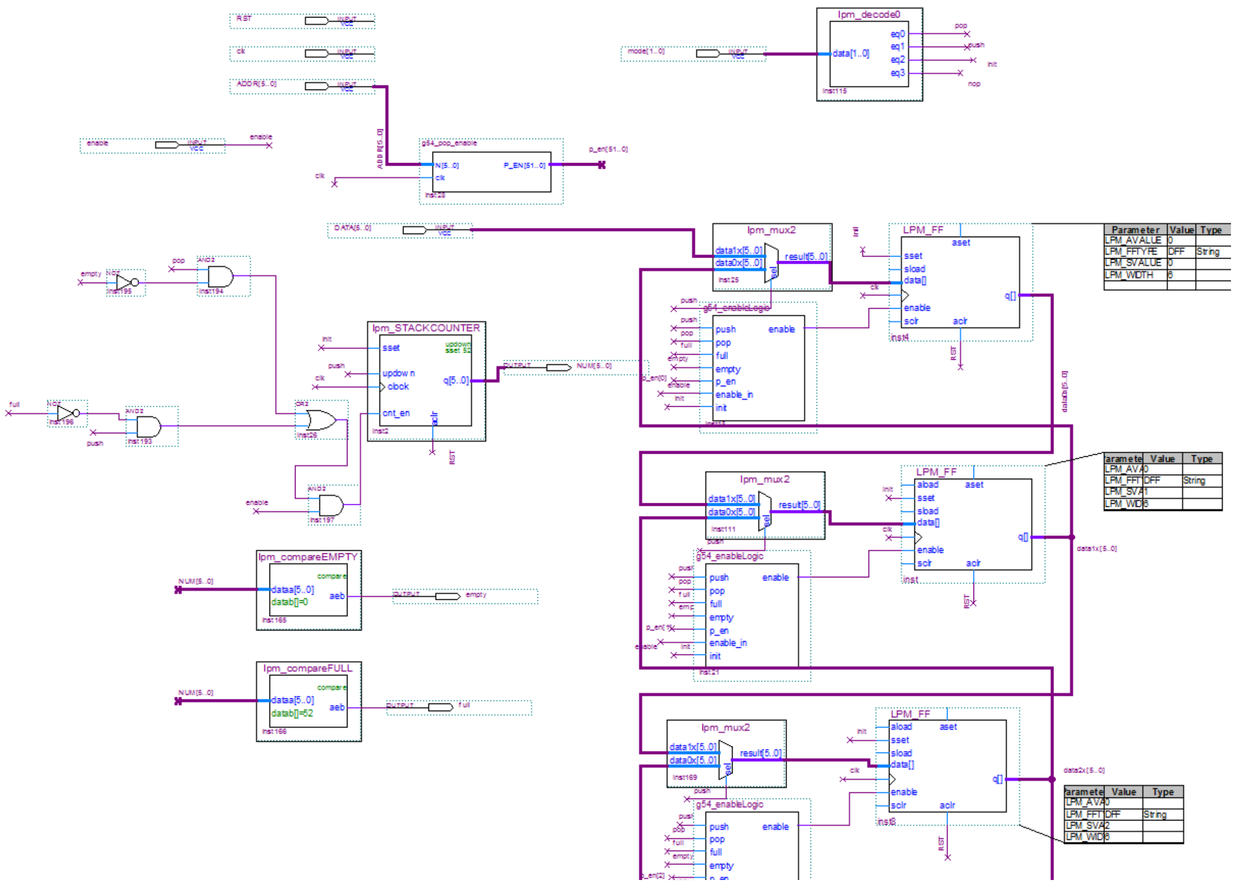
This circuit represents a stack of 52 cards in a deck, each represented by a 6 bit number. This stack can “push” a card to the top of the stack if it is not full, and “pop” a card to remove it from the stack from the top or from the interior stack using a 6-bit input address value.

More detailed description of the stack operations.

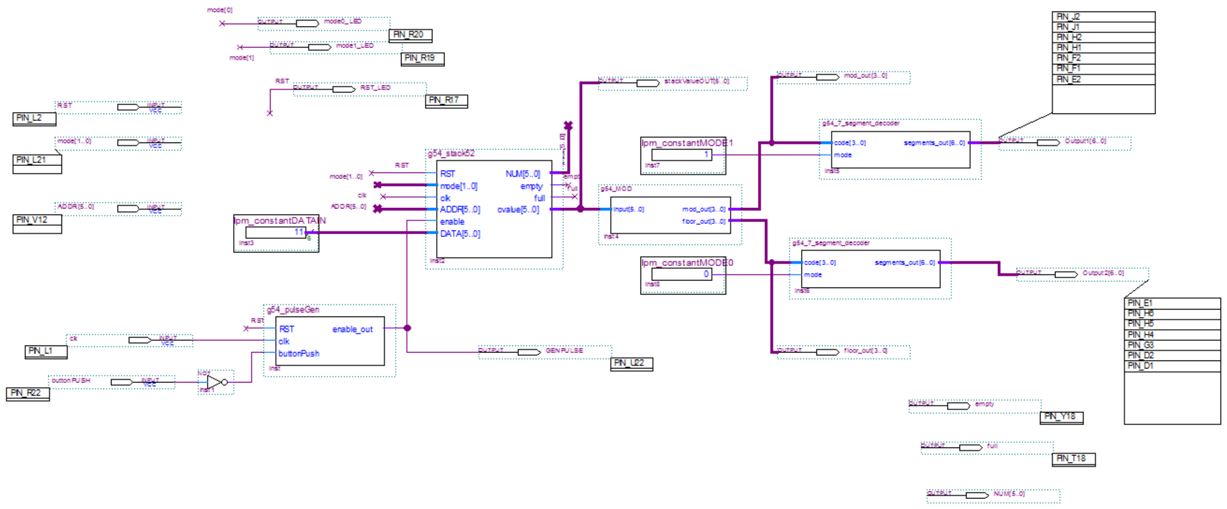
- POP: If NUM=0 (empty=1) then nothing happens. Otherwise, stack slots ADDR+1 through stack slot 52 are shifted up by one place, while stack slots 0 through ADDR are left unchanged. The element in stack slot ADDR is lost (overwritten by the value from stack slot ADDR+1). The value of NUM is decremented by one. Stack slot 52 is the “bottom” of the stack.
- PUSH: If NUM=52 (full=1) then nothing happens. Otherwise, the value of NUM is incremented by one. The value at the DATA input is loaded into stack slot 0, and all of the other stack slots are shifted down (i.e. stack slot 0 gets shifted to stack slot 1, stack slot 1 to stack slot 2, and so on with stack slot 50 being shifted to stack slot 51). The contents of the last slot (slot 51) will be lost.
- INIT: The value of NUM is set to 52, the top entry is set to 0, the next entry is set to 1, the next to 2, and so on, with the final entry (stack bottom) set to 51.
- NOP: (No Operation). Nothing happens in this operation.
- RESET: The value of NUM is set to 0, and the entries in each stack location are all set to zero

This stack was implemented using 52 flip flop modules into a bi-directional shift register, as seen below:

Eva Suska 260618967
Marcel Morin 260605670



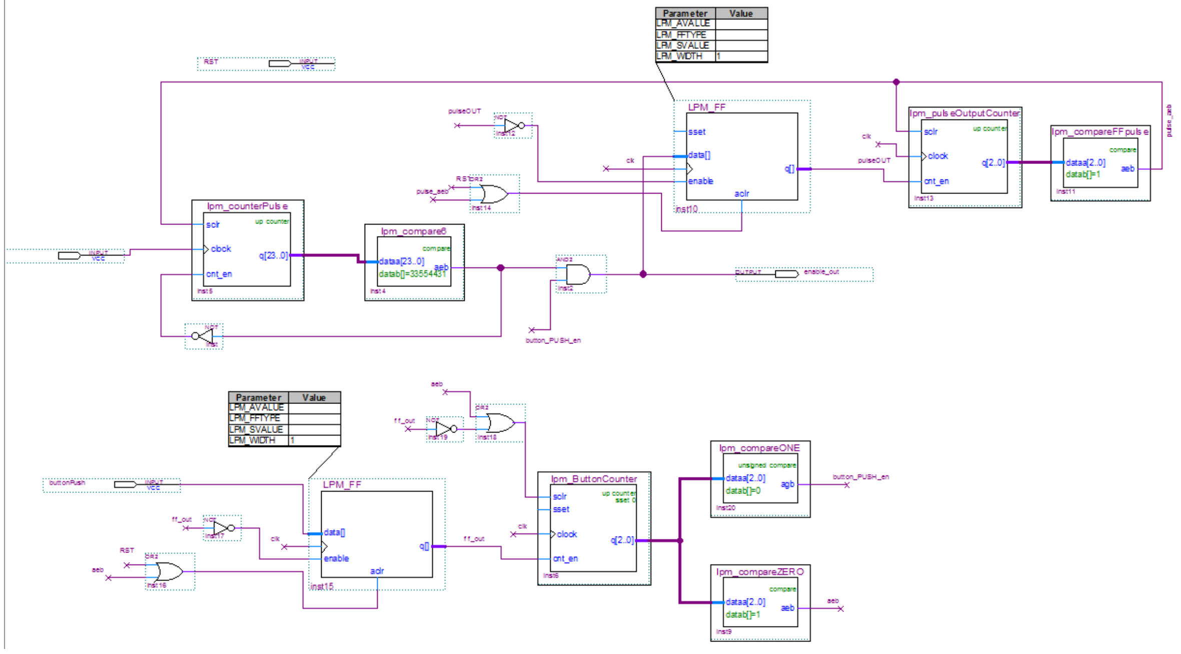
A testbed of the stack was created to present inputs to the stack and displays outputs of the stack to let us evaluate the performance of the stack. This is shown below:



A pulse generator was created for the test bed of the stack to prevent button presses to perform many operations on the stack instead of 1 because of the speed of the clock cycle being much

Eva Suska 260618967
Marcel Morin 260605670

faster than the speed of the button switches. This is shown below:



Parameter	Value
LPM_AVALUE	
LPM_FFTYPE	
LPM_SVALUE	
LPM_WIDTH	1

Parameter	Value
LPM_AVALUE	
LPM_FFTYPE	
LPM_SVALUE	
LPM_WIDTH	1

Along with the pulse generator, a floor and modulus 13 component was needed to obtain the value of the card in the stack. The floor of the value divided by 13 represented the suit of the card, while the modulus 13 represent the face value of the card. This was component was implement in VHDL code with an 6 but input of the card value and 4 bit outputs of the floor and modulus 13. A snippet of this code's architecture is shown below:

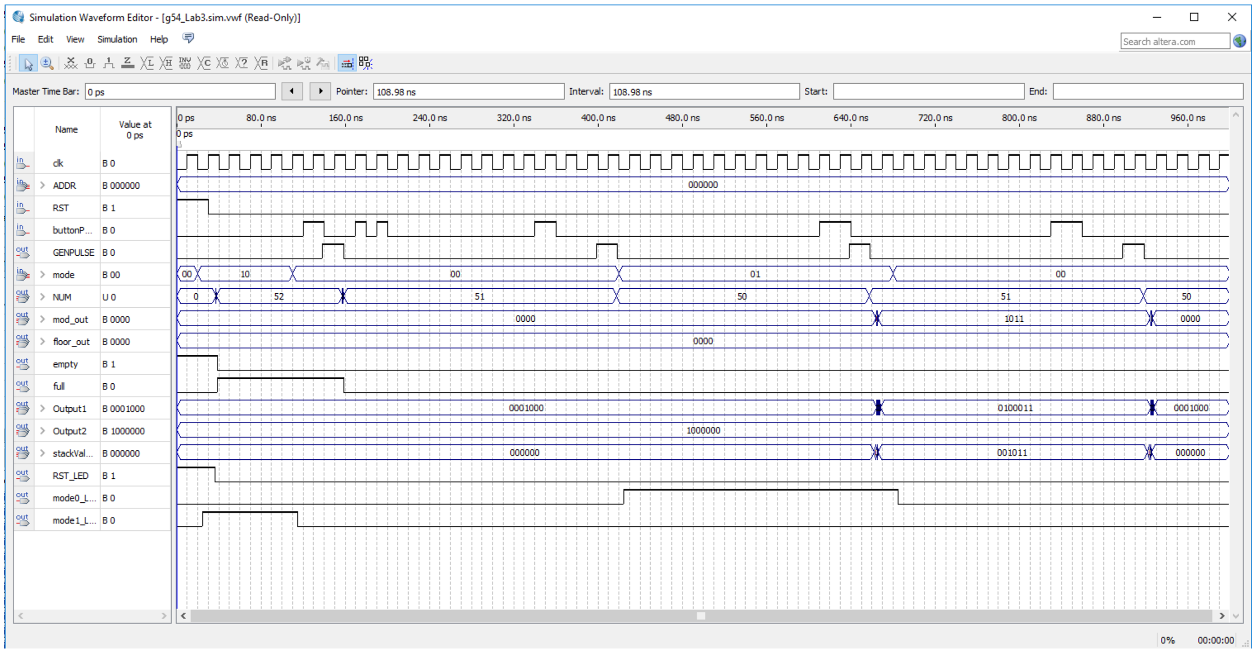
```
BEGIN

temp1 <= (input & "00")+input;
temp2 <= ("000000" & temp1(7 downto 6));
floor_out <= temp2(3 downto 0);

temp3 <= (temp2 & "000") + (temp2 & "00") + temp2;
mod_out <= (input(3 downto 0) - temp3(3 downto 0));
```

Testing

Our testing of the stack was done with the testbed all together which can be seen bellow. We had to modify our pulse generator to a shorter period so that we could simulate and test enough actions given the short simulation window Quartus gives us.



Simulation Waveform Editor - [g54_Lab3.sim.vwf (Read-Only)]

Master Time Bar: 0 ps Pointer: 108.98 ns Interval: 108.98 ns Start: End:

Name	Value at 0 ps
clk	B 0
ADDR	B 000000
RST	B 1
buttonP...	B 0
GENPULSE	B 0
mode	B 00
NUM	U 0
mod_out	B 0000
floor_out	B 0000
empty	B 1
full	B 0
Output1	B 0001000
Output2	B 1000000
stackVal...	B 000000
RST_LED	B 1
mode0_L...	B 0
mode1_L...	B 0

With the testbed setup, we then ran it on the Simulator and tested the modes. As you can see, the stack changes value when you either pop or push to it. The RST and init commands also work at the beginning as our stack goes from empty to initialized. Our button push was even tested in the case where simultaneous presses were picked up successively. In our design the subsequent presses are ignored and only one enable command is executed.

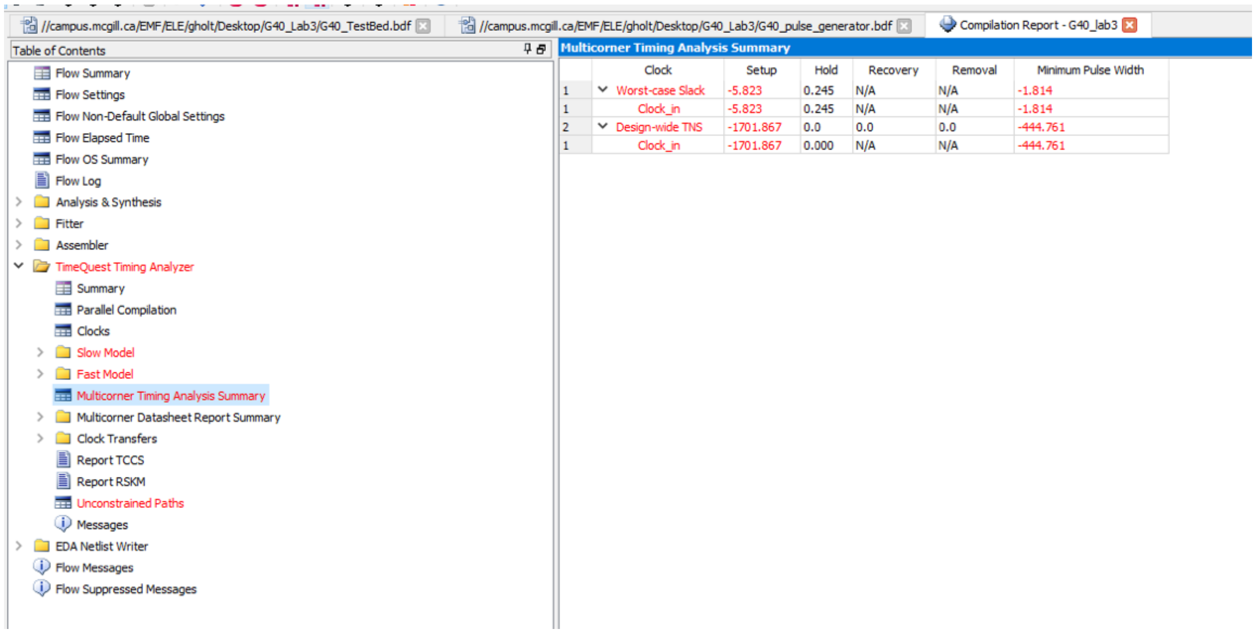
We also set up and uploaded the testbed to run on the Altera board. We set it up to display the value of card to display as 2 of 7 segment displays. One to represent the suit and the other to represent the card's face value. Along with connecting the other input pins to switches to preform operations and reset etc. We tested these operations and the display on the Altera board to make sure they were functioning properly.

We then went ahead with on chip testing using the SignalTap II Logic Analyzer, to more precisely measure the signals inside the chip. We specified the circuit nodes to be analyzed during the simulation: VALUE, EMPTY, FULL, and NUM output. We then uploaded the SignalTap simulation on to the Altera board and then tested the operations using the board to see if the signals were responding to how they were supposed to.

Timing Performance

Our timing performance is shown below and our high timing delay is due mostly to the flip flops in our pulse generator and stack that cause this overall system delay.

Eva Suska 260618967
Marcel Morin 260605670

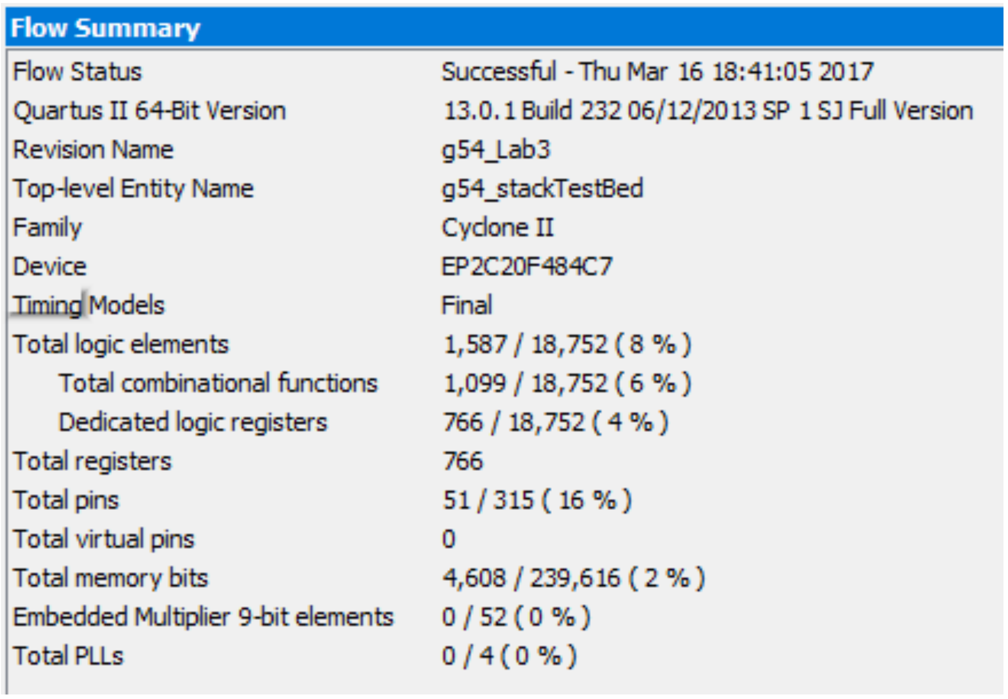

//campus.mcgill.ca/EMF/ELE/gholt/Desktop/G40_Lab3/G40_TestBed.bdf
//campus.mcgill.ca/EMF/ELE/gholt/Desktop/G40_Lab3/G40_pulse_generator.bdf
Compilation Report - G40_lab3
Table of Contents
Flow Summary
Flow Settings
Flow Non-Default Global Settings
Flow Elapsed Time
Flow OS Summary
Flow Log
Analysis & Synthesis
Fitter
Assembler
TimeQuest Timing Analyzer
Summary
Parallel Compilation
Clocks
Slow Model
Fast Model
Multicorner Timing Analysis Summary
Multicorner Datasheet Report Summary
Clock Transfers
Report TCCS
Report RSKM
Unconstrained Paths
Messages
EDA Netlist Writer
Flow Messages
Flow Suppressed Messages

Multicorner Timing Analysis Summary

	Clock	Setup	Hold	Recovery	Removal	Minimum Pulse Width
1	Worst-case Slack	-5.823	0.245	N/A	N/A	-1.814
1	Clock_in	-5.823	0.245	N/A	N/A	-1.814
2	Design-wide TNS	-1701.867	0.0	0.0	0.0	-444.761
1	Clock_in	-1701.867	0.000	N/A	N/A	-444.761

FPGA Resource Utilization

Our stack_52 FPGA resource utilization is shown below. As you can see our stack did not use up a lot of the total resources available, only 8% of the logic elements and only 2% of the total memory bits.



Flow Summary

Flow Status	Successful - Thu Mar 16 18:41:05 2017
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Full Version
Revision Name	g54_Lab3
Top-level Entity Name	g54_stackTestBed
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Total logic elements	1,587 / 18,752 (8 %)
Total combinational functions	1,099 / 18,752 (6 %)
Dedicated logic registers	766 / 18,752 (4 %)
Total registers	766
Total pins	51 / 315 (16 %)
Total virtual pins	0
Total memory bits	4,608 / 239,616 (2 %)
Embedded Multiplier 9-bit elements	0 / 52 (0 %)
Total PLLs	0 / 4 (0 %)