Midterm Exam

ECE 448
Spring 2011

Thursday Section

(15 points)

Instructions
Zip all your deliverables into an archive <last_name>.zip and submit it through Blackboard no later than Thursday, March 10, 10:15 PM EST.

Introduction

Design a circuit that calculates an average of 16 unsigned 4-bit numbers.

The circuit accepts four operands (V, X, Y, Z) in the first clock cycle, and three operands (X, Y, Z) in the following four clock cycles.

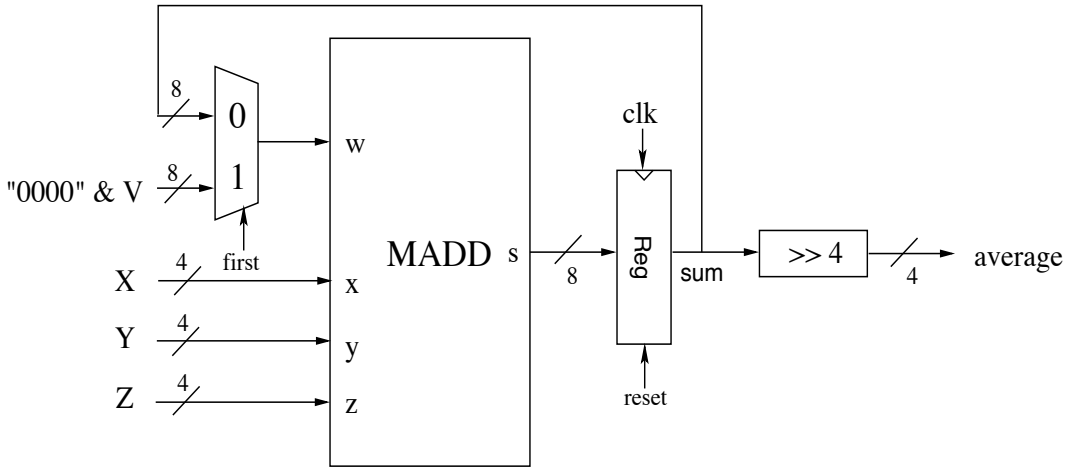
The averaging circuit is specified below using its:

a. Pseudocode,
b. Block Diagram,
c. Interface,
d. Table of input/output ports.

Pseudocode

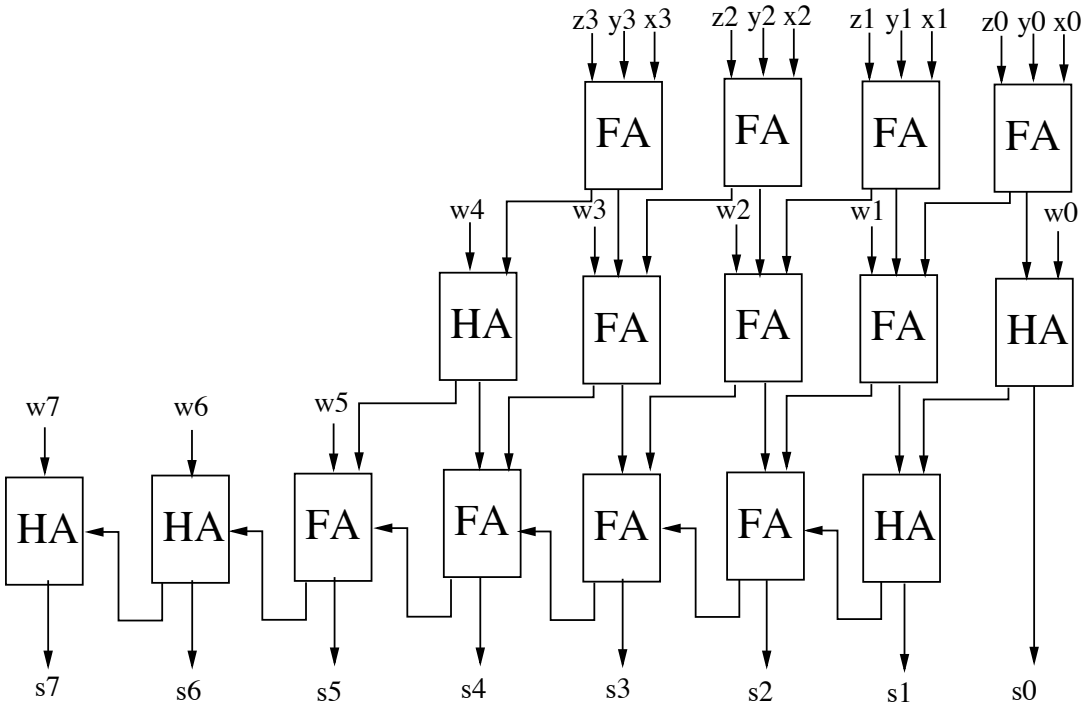
```
sum = 0
for i = 0 to 4 do
      if  i = 0 then
            sum = V + X + Y + Z
       else
            sum = sum + X + Y + Z
      end if
end for
average = sum / 2^4
```

Top-level Block Diagram



Notation:
>> 4 is a logical shift right by 4 positions.

Block Diagram of the Multioperand Adder, MADD



Notation:
x3..x0, y3..y0, z3..z0, w7..w0 are individual bits of inputs x, y, z, w.
s7..s0 are individual bits of the output s.
FA is Full Adder. HA is Half Adder.
The left output of FA is Cout (Carry Out), and the right output is S (Sum).
The left output of HA is C (Carry), and the right output is S (Sum).

Hint:
If you are uncertain about the definition of Full Adder or Half Adder, please read the corresponding articles in Wikipedia at:
http://en.wikipedia.org/wiki/Adder_(electronics)#Half_adder
http://en.wikipedia.org/wiki/Adder_(electronics)#Full_adder

Interface

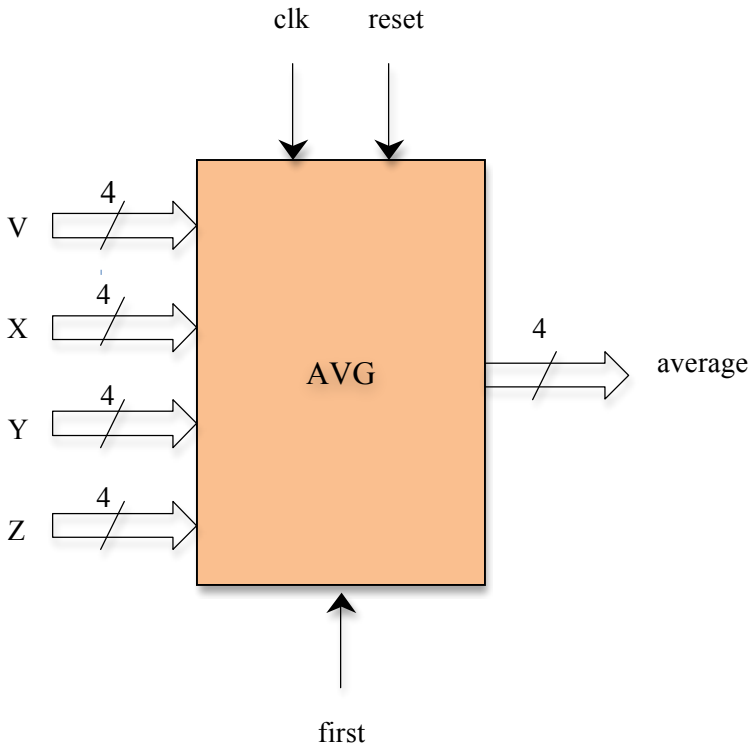


Table of input/output ports

Port	Mode	Width	Function
clk	Input	1	System clock
reset	Input	1	System reset
first	Input	1	Control signal equal to 1 during the first clock cycle of the computations, and 0 otherwise.
V	Input	4	Input V
X	Input	4	Input X
Y	Input	4	Input Y
Z	Input	4	Input Z
average	Output	4	Average of 16 numbers

Example Input and Output

V	X	Y	Z	sum	average
10	**15**	**0**	**1**	**26**	–
–	**7**	**8**	**2**	**43**	–
–	**13**	**10**	**9**	**75**	–
–	**0**	**11**	**0**	**86**	–
–	**5**	**3**	**2**	**96**	**6**

'– ' Stands for don't care. All numbers are given in the decimal notation.

Design Requirements

The combinational portion of the circuit should be described using dataflow VHDL code, and the sequential portion of the circuit should be described using the synthesizable behavioral code. **In your description of MADD, please use a for-generate loop to describe any portions of the circuit with regular structure.** Your code should infer a circuit that requires a minimum amount of FPGA resources. The target clock frequency should be 100 MHz.

Tasks

Perform the following tasks:
1. Write a synthesizable VHDL code representing the described above circuit.
2. Write a testbench verifying the operation of your circuit for the different values.
3. Perform functional simulation of your circuit and use it to debug your VHDL code. Save the obtained timing waveforms.
4. Synthesize your circuit.
5. Implement your circuit using
   FPGA family: Spartan3E,
   Device: 3s100cp132
   Speed Grade: -4.
6. Run the static timing analysis of your circuit.
7. Based on the circuit block diagram and the report from the static timing analysis, determine the most critical path in your circuit and the circuit maximum clock frequency.
8. Based on the implementation reports, determine the number of CLB slices, Logic Cells, LUTs, D flip-flops, and pins used by the circuit.
9. Perform timing simulation of your circuit at the **maximum clock frequency** returned by the static timing analysis. Take a screen shot and include that in the report.

Deliverables

1. VHDL code of your entire circuit fulfilling the requirements specified in the Design Requirements section above.
2. VHDL code of your testbench.
3. Timing waveforms from the functional and timing simulations demonstrating the correct operation of your circuit.
4. Description of the critical path of your circuit.
5. FPGA resource utilization (as defined in Task 8 above)
6. Minimum clock period and maximum clock frequency of your circuit.