SP a 9: Semester Project assignment 9

In this step, the register set and the alu will be instantiated in a new high level architecture to be created, the datapath.

The high level diagram for the datapath unit is given below.



Inside of the datapath will be the instantiation of the register set and the alu.

No “glue” logic should be needed to hook things up.

The high level of the register set looks like this. Inside the register set are 3 decoders, and 16 register lines plus some glue logic to generate the individual register line enables.



And the ALU has the following look to it.



One place to start is to create the dp entity, instantiate it into the testbench, and look at a simulation of the signals interior that you will be hooking up to.

Then add the register set to the datapath and simulate, making sure that you can load values into the registers and drive values onto the A and B busses.

Finally, add the ALU and check the overall operation.

The timing that goes on in a bus cycles is depicted in the following.



Testbench vesions –

tbd.vhdl – runs the bus cycle but is not complete – only has signals to run the register set.

tbd1.vhdl – has the register set integrated into the datapath that it simulates.

The bus cycle here is run by procedure buscycle with many parameters which are explained here:

PROCEDURE buscycle (

abusop,bbusop,rbusop - can have values of

idle : the bus is high Z this bus cycle

drva : the a bus is being driven by the register set. Reg(aregno)->Abus

drvb : the b bus is being driven by the register set. Reg(bregno)->Bbus

drvab : both the a and b busses are driven using aregno and bregno

drvr : the ALU output is driving the R bus this bus cycle

tbdra : the testbench is driving a value onto the A bus

tbdrb : the testbench is driving a value onto the B bus

tbdrvab : the testbench is driving a value onto both A and B bus

tbdrr : the testbench is driving a value onto the R bus and asserting

register load signal using the aregno field

tblda not used yet

tbldb not used yet

abusval,bbusval,rbusval – std\_logic\_vector value for the busses when tbdrx

aregno,bregno – the register numbers (an integer field)

aluoper – the operation to be performed by the alu this bus cycle

idle : no alu operation is asserted so the alu load and drive are not asserted

aluopand, etc : the operation to be performed

the input registers are loaded and the control signals

asserted

The report on the datapath is in two parts. (they can be incorporated into the same document)

Part 1 – The HDL code (can be incorporated into one doc file or individual files (no archives)), simulation results that show it working for the operations that the testbench generates.

Part 2 – A report on the results from synthesis showing the top level schematic and descending down at least 2 levels. Also the data on CLBs, registers, and any other resources used.