This is further information for SP a 7

This is for the testbench that applies tests to the ALU version that has input registers and an output bus driver. It also generates flags.

The code for the testbench is in file tbalu3.vhdl

The testbench runs a bus cycle on the input busses. The only way to load values into the ALU input registers is for the testbench to load them. The following shows the timing on the busses.



In the testbench code in tbalu3.vhdl this buy cycle is run through a procedure declared in the declarative region of the process. Then to run a bus cycle the user simply call the procedure buscycle with the appropriate parameters

Procedure buscycle has the following parameters

Procedure buscycle (

 (abusop,bubusop : busoptype

 Busoptype can have a value of

 idle : The bus is highz

 drva : A bus driven from registers

 drvb : B bus driven from registers

 tbdra : testbench drives the A BUS

 tbdrb : testbench drives the B BUS

 Abusval,bbusval : std\_logic\_vector (15 downto 0)

 These are values that are needed when the value of the bus comes

from the testbench

 aluinctl : aluctlintyp

 has values of

 idle : Load no values this bus cycle

 ltca : Load ABUS into A input register

 ltcb : Load BBUS into B input register

 ltcab : Load both A and B input register from respective busses

 oper – the value to assign to control signal op - 4 entry std\_logic\_vector

 abus\_v

 Cin\_v the last 4 are the std\_logic value of the signals

 s\_v

 dr\_v)

This makes it very easy to write the tests. They will look something like this

 buscycle (tbdra,tbdrb,i0,i0,ltcab,opand,'0','0','0','1');

 buscycle (tbdra,tbdrb,i0,i1,ltca,opand,'0','0','0','1');

 buscycle (tbdra,tbdrb,i0,i2,ltcb,opor,'0','0','0','1');

 buscycle (tbdra,tbdrb,i0,i3,idle,opor,'0','0','0','1');



Simply write tests here to do a better job of testing your alu design.