The assignment will structurally connect up several of the component building block that have been created in assignments MB Comp 1 through 7 to implement a multifunction ALU capable of any 8-bit logic function on two operand inputs A and B or 8-bit binary addition and subtraction.

The general architecture of the unit is shown in Figure 1 below.



Figure 1. MicroBaby ALU

The entity interface will have all the signals that cross the outer box, the signals that interface with the other components of the datapath. They are:

The Inputs:

A, B The data inputs, each an 8-bit BIT\_VECTOR(7 downto 0)

Cin A data input, a single Bit

Csel An input control signal that selects the value into the Add/Subtract component

When 00 the Cin input supplied with the Cin flag from the status register

When 01 the Cin input supplied with the complement of the Cin flag

When 10 the Cin input set to ‘0’

When 11 the Cin input set to ‘1’

AddSub A control signal input the indicate if the operation if addition or substraction

Arlo A control signal input that indicates if the operation it an arithmetic of logic operation

F A 4-bit input that indicates the logic function to be performed

The Outputs:

R An 8-bit BIT\_VECTOR output that is the result of the current operation

Cout The Carry output of the add/subtract function unit

N The negative flag – the most significant bit of the result

Z The zero flag – indicates if the result is all 0s – generated by an all zeros component

The components used in the multifunction ALU:

8-bit Add/Subtract Unit – this is an 8-bit binary adder with A(7..0), B(7..0), and Cin as inputs

Produces Sum(7..0) and Cout as outputs

8-bit 2-to-1 mux with 2 8-bit data inputs, a single select, producing an 8-bit output.

Used twice – once to select B or NOT B for input to the Add/Sub unit

once to select if the output comes from the Add/Sub unit ot the logic unit

All0s detector – used to generate the Z output input R(7..0) output Z

Logic Unit – An 8-bit 4-to-1 mux unit where the data input of all units are connected to the

F(3..0) inputs, the 8 s0 inputs are connected to A(7..0) and s1 connected to B(7..0)

A 4-to-1 mux to select the proper Cin input to the add/sub unit. The select inputs are connect to

Csel(1..0). The 0 data input is connected to the Cin flag, 1 data input is supplied with NOT Cin, 2 data input is connected to fixed ‘0’, the 3 data input is connected to ‘1’

THE ASSIGNMENT: (Part I)

Create the ENTITY for the mfalu. The ports of the entity are described above.

Frist write the ENTIRY.

Then create a structural architecture. You will need the components described above. There are a few internal signal that will need to be declared in the architecture to connect the components.

Compile your ENTITY and ARCHITECTURE in ModelSim.

The next step is to write a testbench. A shell of a testbench with a process to apply values to the signals is posted on the webpage. Note the structure of this process.

Start a report on the VHDL portion of this project, including code and the results of simulation. It may take several waveform figures to show the waveforms in sufficient detail. Provide a write the show you examined the waveform results. This project is verification through visual inspection.

**Part II ( a separate project grade)**

Take you VHDL code and begin a QUARTIS project. You will need to create a directory for the Quartis project. In that directory, place a copy of the files containing the ENTITY and ARCHITECTURE of all the units used in the project. For the add/sub unit you will need not only the add/sub unit but any subunit that is included in it. The same for the logic unit which uses the 4-to-1 mux but here you already need that.

Synthesize the project in Quartis. Not that the synthesis process may take just a bit of time. After synthesis is complete add information on the synthesis to the report. These include, but is not limited to, the synthesized design at the top level. Also drop down into the add/sub unit and logic unit. Include in the report how many LUTs were needed. As this is a combinational logic unit there should be no registers. What is the longest timing path?



Compile your VHDL code. Modify the testbench of the ha and fa to test your code and simulate so that all cases are simulated. Prepare a report showing your VHDL results.

Start up Quartis and perform a synthesis on your design. In your report be sure to show the circuit synthesized by Quartis ala the sample report.

Submit it to the dropbox.