

Homework #8b

Name:

Section:

submit via gradescope

For 1023 memory locations in a BRAM, build a circuit to read in a 16-bit DATA_IN whenever READY goes HIGH, increment the data by 7, store the value in the current memory location, and then read the data stored at this memory location back out and store it in an output register called DATA_OUT. Then repeat for the next memory location.

You are given hw8b2_tb.vhdl,

You are given partial files for HW8b2_design_template.pptx, hw8b2_cu.vhdl, and hw8b2_dp.vhdl Which you will need to complete.

Here is the mini-C to implement:

```
1.  for(i=0; i<1023; i++) {      // for every memory location
2.      while(READY==0);        // wait for ready (rising edge)
4.      RAM[i] = DATA_IN + 7; // store data + 7 in memory location i
5.      DATA_OUT = RAM[i];     // read data from memory location i
6.      while(READY==1);        // wait for ready (falling edge)
7.  } // end for
```

You must use the mini-C design method taught in lesson 10.

Turn in:

1. Block diagram of your datapath design. (see HW8b2_design_template.pptx and hw8b2_dp.vhdl)
2. State diagram for your control unit design (see HW8b2_design_template.pptx and hw8b2_cu.vhdl)
3. The output control word table for your design. (see HW8b2_design_template.pptx and hw8b2_cu.vhdl)
4. A testbench and partial datapath and partial control unit VHDL code are provided. The datapath has a BRAM instantiation with 1024 16-bit initial data values preloaded. Upload to bitbucket your final version of the control unit and datapath VHDL code.

5. Your testbench output simulation plots showing at least the following signals below - remove all other signals. Your plot should be zoomed in showing all the states of the statemachine executing from one READY to the next READY showing the proper value written to DATA_IN and the proper DATA_OUT saved in the register. See class notes for example.

- clk
- reset
- Status word
- Control word
- Ready
- DATA_IN (from testbench)
- DI (into the BRAM)
- DO (from the BRAM)
- DATA_OUT (from the register)
- i (or address into BRAM, WR_ADDR, RD_ADDR)
- control unit state