Akram_10_17_2021_Grading_Rubric_Add_Sub

Sunday, October 17, 2021 9:06 PM

CSC 343, Fall 2021

Very SIMPLE Add-Sub unit with Register file (3-ported RAM)

Instructor: Professor Izidor Gertner

Screenshots with brief explanations due: OCTOBER 13, 2021, BY 6:15 PM

Report, Video, QAR file_ with README : October 17 TH, 2021, by 10:00 PMN

GRADING

I will grade based on the following:

Please SELF EVALUATE your work based on the following grading RUBRIC and submit to me in a separate file named
mylast_name_my_evaluation_ grade_add_sub

A. 40 points

Shown in the waveforms windows

- Content of Register named Instruction Register -IR specify two operand registers, destination register where to store the result, and the operation code: add or sub.

The fields should be as following:

- Instruction Register Opcode Bits [26:31] = 000000 encode addition operation
- Instruction Register Opcode Bits [26:31] = 000001 encode subtraction operation

- Instruction Register Bits [21:26] = destination register index ranging from {00000} to {11111}
- Instruction Register Bits [16:20] = source register index ranging from {00000} to {11111}
- Instruction Register Bits [11:15] = source register index ranging from {00000} to {11111}
- The remaining bits in IR are set to zero.
- Show IN WAVEFORM WINDOW the corresponding registers in Register File

Show the above for three different sets of numbers.

B. 40 points

For registers RS,RT,RD 5 bit addresses have to be stored in IR register fields, and applied to register ports RA,RB,RW and shown in waveform window.

Demonstrate in waveform window addition subtraction operation from first writing event to the next writing event. Repeat the above for three different set of numbers.

C. 20 points for 2 min video presentation
D. QAR file named last_name lab title + README file that explains HOW TO VERIFY YOUR DESIGN.

NOTE: PLEASE BE HONEST. IF YOUR GRADE IS MUCH HIGHER THAN MY GRADE , I WILL REDUCE THE GRADE.

A. Yes, this is done in the ENTITY of my VHDL project (line 24). The appropriate signals and porting is also done in the ARCHITECTURE. This is shown in the waveform windows (named Akram_IR). IR stands for Instruction Register.

   The two operand registers are taken care of in lines 38 & 39, and also 67 & 68. This is also shown in the waveforms in Akram_Sum_rd (the output).
   Destination register is line 37 and also 66, also shown in waveform (same justification).
   Operation code is explained as a STD_LOGIC since I'm focusing on the 26th bit of the Instruction Register (line 36 and 65). All of this is shown in the waveform through the Akram_IR input.

   40/40

A. Yes, this is also done and explicitly shown in the waveform (and video). Akram_Sum_rd. The IR is 32 bits, and RA, RB, and RW are 5-bits each within that 32-bit structure. The other remaining bits are the operation (add or sub) and 11 0s (specified in rubric). Demo is in video and also screenshots in my report included.

   40/40

C. (2 min presentation included where I explain the waveform and allude to my written report which has even more detail (I left detailed comments for my future self so I remember exactly what I wrote).

   20/20

D. QAR file and README already submitted.