Laboratory 6

(Due date: **002**: April 15th, **003**: April 16th)

OBJECTIVES

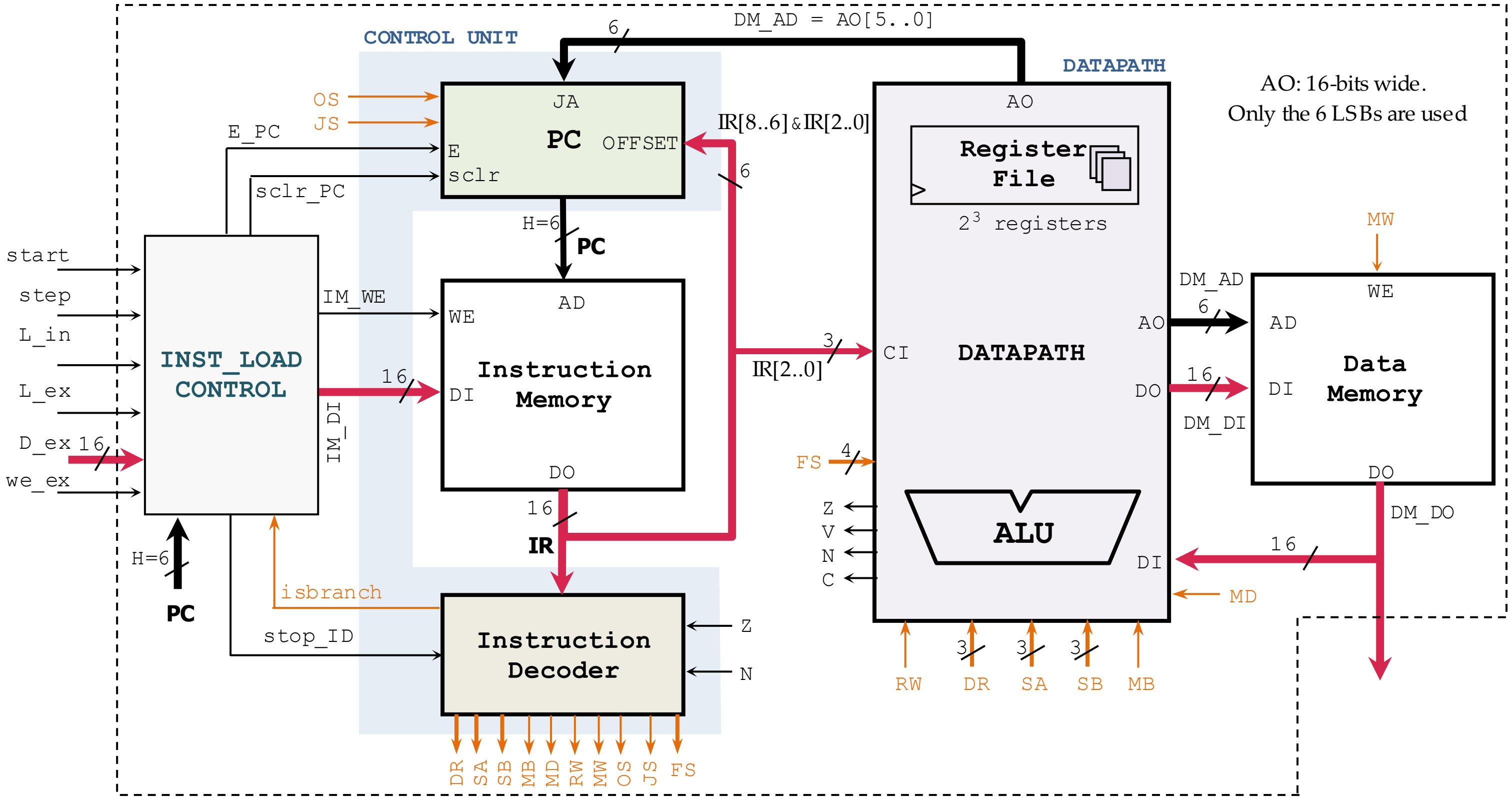
- ✓ Design a 16-bit microprocessor with Single-Cycle Hardwired Control.
- ✓ Implement an Instruction Set.

VHDL CODING

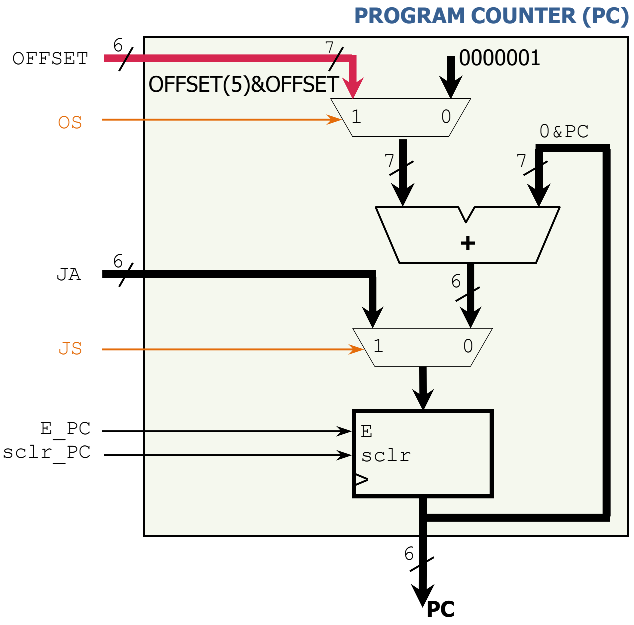
- ✓ Refer to the Tutorial: VHDL for FPGAs for a tutorial and a list of examples.

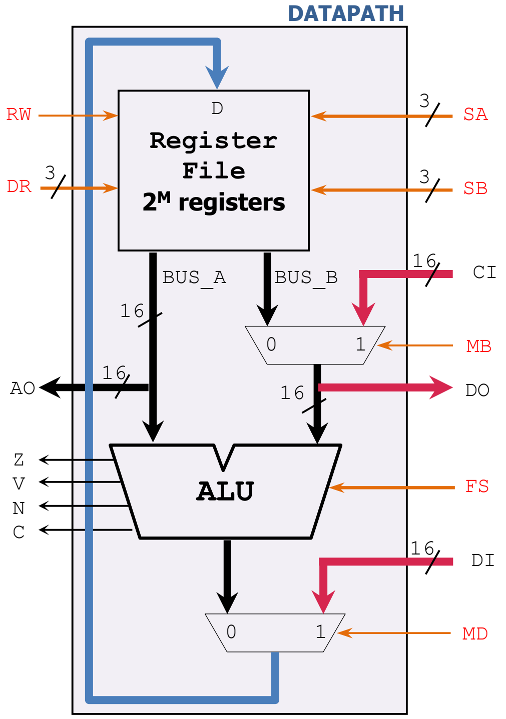
FIRST ACTIVITY: 16-BIT MICROPROCESSOR DESIGN AND SIMULATION (100/100)

- Implement the **Simple Computer** (see Notes – Unit 6): uP with 6-bit IM/DM address, 16-bit instructions, and 16-bit data.



- **Components**:
  - ✓ DM, IM: 64 words, 16 bits per word. Use the files RAM_emul.vhd, my_rege.vhd. (set the proper parameters).
  - ✓ Datapath: (note that CI[2..0] = IR[2..0], CI[15..3]="00…0")
    - ▫ Register File: 8 registers (R0 – R7) are included. See Notes – Unit 6 for an example with 4 registers.
    - ▫ ALU: Use the files: alu.vhd, alu_arith.vhd, alu_logic.vhd, super_addsub.vhd, fulladd.vhd.
  - ✓ PC: Note that OFFSET is a 6-bit signed number. The adder uses 7 bits, from which we only retrieve the 6 LBSs.
  - ✓ Instruction Decoder (ID): This is a large combinational circuit. The outputs depend directly on the inputs.
    - ▫ The outputs are generated based on the instructions on IR (Instruction Register).
    - ▫ Instruction Set: For the list of instructions, refer to Notes – Unit 6. The Instruction Set does not include instructions that read the V and C bits. Thus, the ID does not consider these two bits.
    - ▫ stop_ID: This input signal causes all the ID outputs to be '0' if stop_ID=1.
    - ▫ isbranch: If the instruction in IR is a branch or jump instruction, this signal is set to '1'.
  - ✓ Instruction Load Control: This component is required in order to write instructions on the IM, and then to trigger program execution. Use the file instload_ctrl.vhd (use parameters H=6, N=16) This circuit is a FSM that works as follows:
    - ▫ To store instructions on IM from an external port, assert L_ex and then use the inputs D_ex and we_ex.
    - ▫ To store instructions on IM using pre-stored hardwired data, assert L_in.
    - ▫ Once instructions are written on the IM, program execution is started by asserting start for a clock cycle. The step signal controls whether to enable program execution (step=1) or disable it (step=0).





SIMULATION

- We will execute the following pre-stored program (storing numbers from 43 down to 29 in Data Memory on addresses 0 to 14): (see instload_ctrl.vhd). Note that the number to be stored appears in R6.

Address	Assembly Program	VHDL code snippet
000000	start: LDI R2,5	CD(0) <= “1001100010---101”
000001	LDI R6,7	CD(1) <= “1001100110---111”
000010	ADI R6,R6, 7	CD(2) <= “1000010110110111”
000011	MOVA R4,R6	CD(3) <= “0000000100110---"
000100	ADD R6,R4,R6	CD(4) <= “0000010110100110”
000101	loop: INC R6,R6	CD(5) <= “0000001110110---"
000110	ST R4,R6	CD(6) <= “0100000---100110”
000111	BRZ R4, -7	CD(7) <= “1100000111100001”
001000	DEC R4,R4	CD(8) <= “0000110100100---"
001001	JMP R2	CD(9) <= “1110000---010---"
001010		…
...		

- Tesbench:
  - ✓ Set L_in=1 for a clock cycle. Then wait 70 cycles for the program to be written on the Instruction Memory.
  - ✓ Set start=1 for a clock cycle. Make sure that step = 1 during the execution of the program.
    - ▫ Verification: To see if the instructions are processed in the right order, take a look at PC and IR. Then, observe the R0-R7 values as well as other signals such as the ID outputs. To verify that the right data was stored on DM (Data Memory), you can add the Individual Registers (from 0 to 14) of DM to the waveform.

- Design Flow and verification:
  - ✓ Write the VHDL for the given circuit. Synthesize your circuit. (Run Synthesis).
  - ✓ Perform Functional Simulation (Run Simulation → Run Behavioral Simulation). **Demonstrate this to your TA.**

- Submit (as a .zip file) the generated files: VHDL code and VHDL testbench to Moodle (an assignment will be created). DO NOT submit the whole Vivado Project.

- **You can work in teams of up to two (2) students. Only one Moodle submission per team.**

TA signature: ______________________________ **Date:** ___________________________