FINAL Lab Project: Single Cycle CPU-LITE

Instructor: Professor Izidor Gertner
Due date: By May 10, 2021
What to submit: Detailed report, 2 min video presentation, Quartus project files in QAR format, and README file.
Note: All files you submit MUST have your last name as a prefix.

Objective:

The ultimate objective of this final project is to write a program to compute dot product using instructions and cpu you have created.

Design single cycle CPU based on the MIPS instruction set architecture, as it was described in the class and also described in the textbook.
The instructions that you need to implement are MIPS machine instructions listed in the table below

add	addi	addiu	addu	sub	subu	and	andi	nor	ori	sll	srl	sra	sw	lw	beq	bne	j	mult	div

Design and implement in VHDL CPU controller that generates control signals to determine the data path for each instruction.
HOW to TEST:

- **Input machine instructions you want to execute into Instruction Memory block you have designed.**
- **Input data you intend to process into Data Memory block you have designed.**
- **Load the address of the first instruction to PC-Program Counter register.**
- **Start executing the code by fetching the first instruction to instruction register-IR, and then continue step by step.**
- **Demonstrate the correctness of your program execution using waveforms in simulation, and by comparing to MIPS program on MARS simulator.**

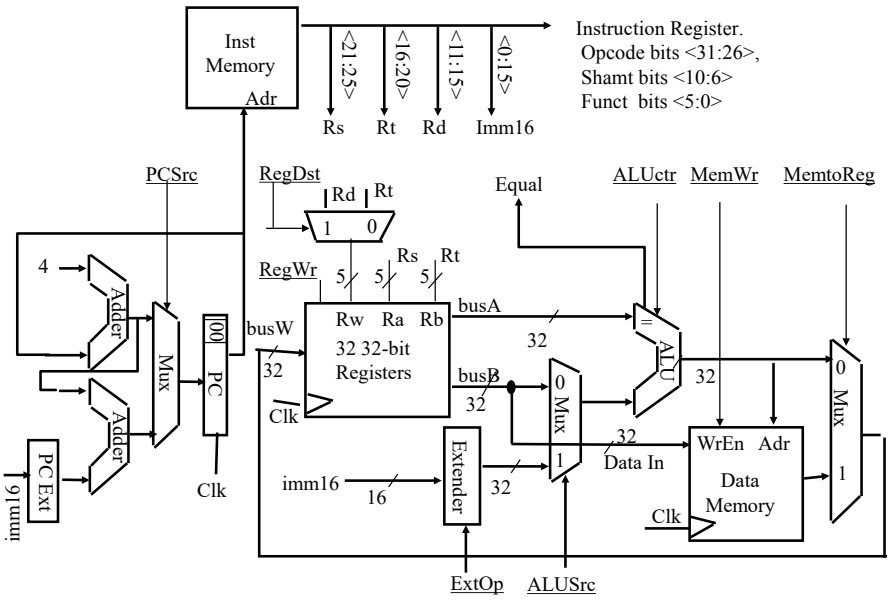
FINAL Lab Project: Single Cycle CPU-LITE

Instructor: Professor Izidor Gertner
Due date: By May 10, 2021
What to submit: Detailed report, 2 min video presentation, Quartus project files in QAR format, and README file.
Note: All files you submit MUST have your last name as a prefix.

1. **Perform the ultimate test of your design by inputting integers x_1, x_2, x_3, x_4, x_5, y_1, y_2, y_3, y_4, y_5, into DATA Memory, and computing the following expression**

$$Z = \sum_{k=1}^{5} X_k Y_k$$

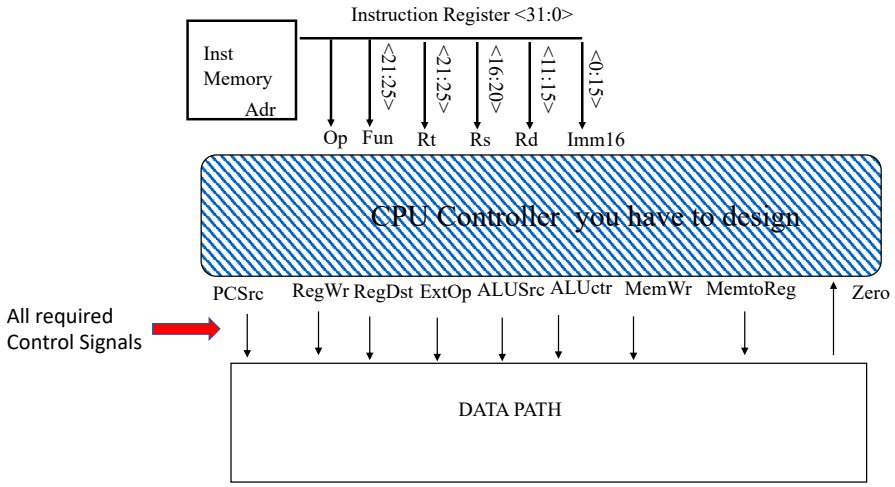
Putting it All Together: Final Project A Single Cycle CPU



FINAL Lab Project: Single Cycle CPU-LITE

Instructor: Professor Izidor Gertner
Due date: By May 10, 2021
What to submit: Detailed report, 2 min video presentation, Quartus project files in QAR format, and README file.
Note: All files you submit MUST have your last name as a prefix.

Visualization of CPU Controller Operation



Components you will need:

1. Data Memory: Memory size 64 bytes, data word size 32 bits.
2. Instruction Memory: Memory size 128 bytes, instruction size 32 bits.
3. Register file dual ported for two reads and one additional write: 32 registers, register size 32 bits.
4. PC Register: 32 bit register to store instruction address.
5. IR Register; 32 bit register to store instruction during execution.
6. ALU, 32 bit operands
7. 2 adders for Next Address Logic unit
8. 2: 1 Multiplexers, with 32 bit input, and output wires
9. 16 to 32 bit extender

Note: Please use your drawings in the report.

FINAL Lab Project: Single Cycle CPU-LITE

Instructor: Professor Izidor Gertner
Due date: By May 10, 2021
What to submit: Detailed report, 2 min video presentation, Quartus project files in QAR format, and README file.
Note: All files you submit MUST have your last name as a prefix.

APPENDIX

Not REQUIRED

Example of VHDL code for 3 ported Register File

YOU HAVE DESIGNED 3 PORTED REGISTER FILE USING LPM MODULES.

YOU CAN LOOK INTO EXAMPLE VHDL **CODE for**

3 PORTED register file as an example (Note: this code is from the WEB and may not work!!):

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
 LIBRARY altera_mf;
 USE altera_mf.all;

 ENTITY ram3port IS
     PORT
     (
         clock       : IN STD_LOGIC ;
         data        : IN STD_LOGIC_VECTOR (31 DOWNTO 0);
         rdaddress_a   : IN STD_LOGIC_VECTOR (4 DOWNTO 0);
         rdaddress_b   : IN STD_LOGIC_VECTOR (4 DOWNTO 0);
         wraddress     : IN STD_LOGIC_VECTOR (4 DOWNTO 0);
         wren        : IN STD_LOGIC  := '1';
         qa      : OUT STD_LOGIC_VECTOR (31 DOWNTO 0);
         qb      : OUT STD_LOGIC_VECTOR (31 DOWNTO 0)

     );

 END ram3port;

 ARCHITECTURE SYN OF ram3port IS
```

FINAL Lab Project: Single Cycle CPU-LITE

Instructor: Professor Izidor Gertner
Due date: By May 10, 2021
What to submit: Detailed report, 2 min video presentation, Quartus project files in QAR format, and README file.
Note: All files you submit MUST have your last name as a prefix.

```
    SIGNAL sub_wire0    : STD_LOGIC_VECTOR (31 DOWNTO 0);
    SIGNAL sub_wire1    : STD_LOGIC_VECTOR (31 DOWNTO 0);
    COMPONENT alt3pram
    GENERIC (
        indata_aclr     : STRING;
        indata_reg      : STRING;
        intended_device_family      : STRING;
        lpm_type        : STRING;
        outdata_aclr_a      : STRING;
        outdata_aclr_b      : STRING;
        outdata_reg_a       : STRING;
        outdata_reg_b       : STRING;
        rdaddress_aclr_a        : STRING;
        rdaddress_aclr_b        : STRING;
        rdaddress_reg_a     : STRING;
        rdaddress_reg_b     : STRING;
        rdcontrol_aclr_a        : STRING;
        rdcontrol_aclr_b        :
        STRING; rdcontrol_reg_a     :
        STRING; rdcontrol_reg_b     :
        STRING; width       : NATURAL;
        widthad     : NATURAL;
        write_aclr      : STRING;
        write_reg       : STRING
);
PORT (
qa   : OUT STD_LOGIC_VECTOR (31 DOWNTO 0);
outclock    : IN STD_LOGIC ;
qb   : OUT STD_LOGIC_VECTOR (31 DOWNTO 0); wren    : IN STD_LOGIC ;
inclock : IN STD_LOGIC ;


            : IN STD_LOGIC_VECTOR (31 DOWNTO 0);
        rdaddress_a : IN STD_LOGIC_VECTOR (4 DOWNTO 0); wraddress
        : IN STD_LOGIC_VECTOR (4 DOWNTO 0);
        rdaddress_b : IN STD_LOGIC_VECTOR (4 DOWNTO 0)

);
END COMPONENT;
 BEGIN
    qa    <= sub_wire0(31 DOWNTO
    0); qb    <= sub_wire1(31
    DOWNTO 0);
alt3pram_component : alt3pram
```

FINAL Lab Project: Single Cycle CPU-LITE

Instructor: Professor Izidor Gertner
Due date: By May 10, 2021
What to submit: Detailed report, 2 min video presentation, Quartus project files in QAR format, and README file.
Note: All files you submit MUST have your last name as a prefix.

GENERIC MAP (

```
    indata_aclr => "OFF",
    indata_reg => "INCLOCK",
    intended_device_family => "Stratix II", lpm_type
    => "alt3pram",
    outdata_aclr_a => "OFF", outdata_aclr_b
    => "OFF", outdata_reg_a => "OUTCLOCK",
    outdata_reg_b => "OUTCLOCK",
    rdaddress_aclr_a => "OFF",
    rdaddress_aclr_b => "OFF",
    rdaddress_reg_a => "INCLOCK",
    rdaddress_reg_b => "INCLOCK",
    rdcontrol_aclr_a => "OFF",
    rdcontrol_aclr_b => "OFF",
    rdcontrol_reg_a => "UNREGISTERED",
    rdcontrol_reg_b => "UNREGISTERED",
    width => 32,
    widthad => 5, write_aclr
    => "OFF", write_reg =>
    "INCLOCK"
```

)

```
PORT MAP (
    outclock => clock,
    wren => wren, inclock
    => clock, data =>
    data,
    rdaddress_a => rdaddress_a,
    wraddress => wraddress,
    rdaddress_b => rdaddress_b, qa
    => sub_wire0,
    qb => sub_wire1
```

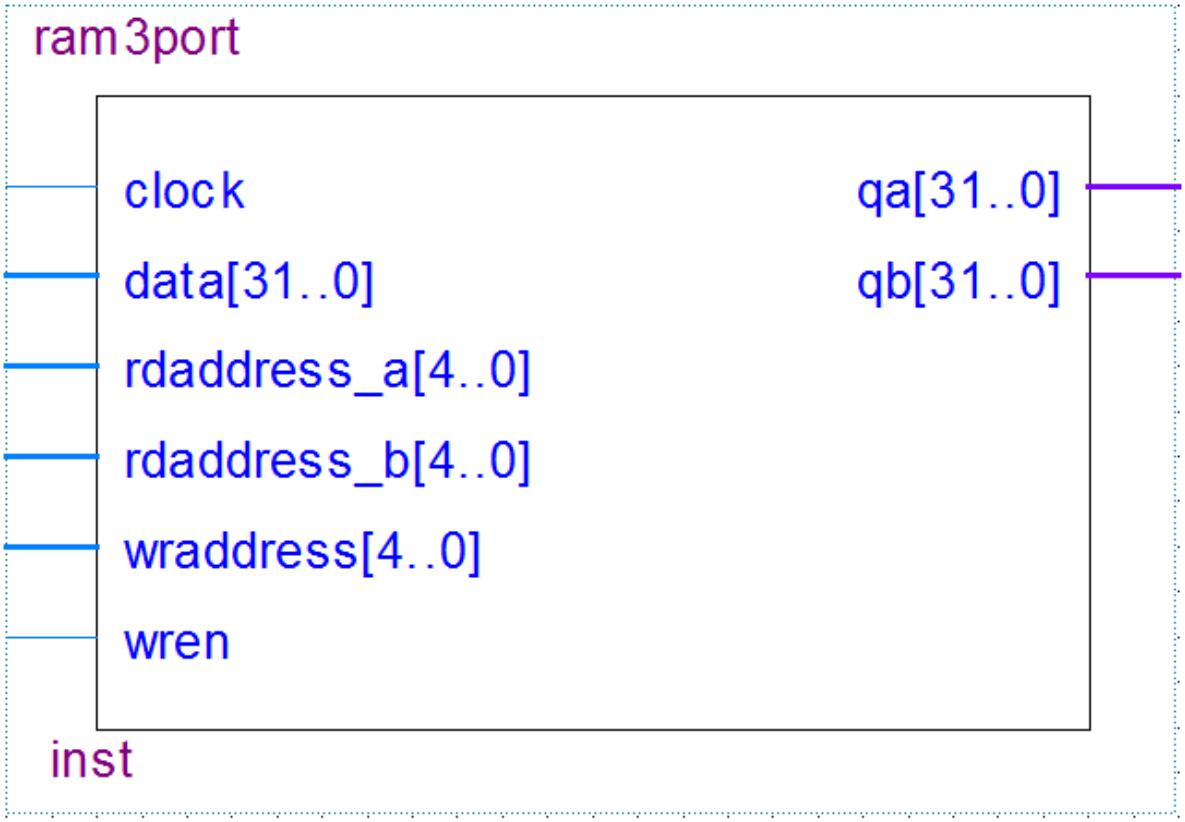
);

```
END SYN;
```

REMARK: MAKE SURE that the file name is ***ram3port (the same as entity).***
Create a symbol for your use.

FINAL Lab Project: Single Cycle CPU-LITE

Instructor: Professor Izidor Gertner
Due date: By May 10, 2021
What to submit: Detailed report, 2 min video presentation, Quartus project files in QAR format, and README file.
Note: All files you submit MUST have your last name as a prefix.



END REGISTER FILE EXAMPLE