EENG483/EENG661
Introduction to VLSI

Project

Due by May 14, 2024

PART I: Design of the Core

Your goal in this project is to design a CPU, which can execute every instruction in the instruction set. Each instruction is 24 bits long, and should take **two clock cycles to execute**. The instruction set consists of 15 instructions (and 3 reserved opcodes) as given below:

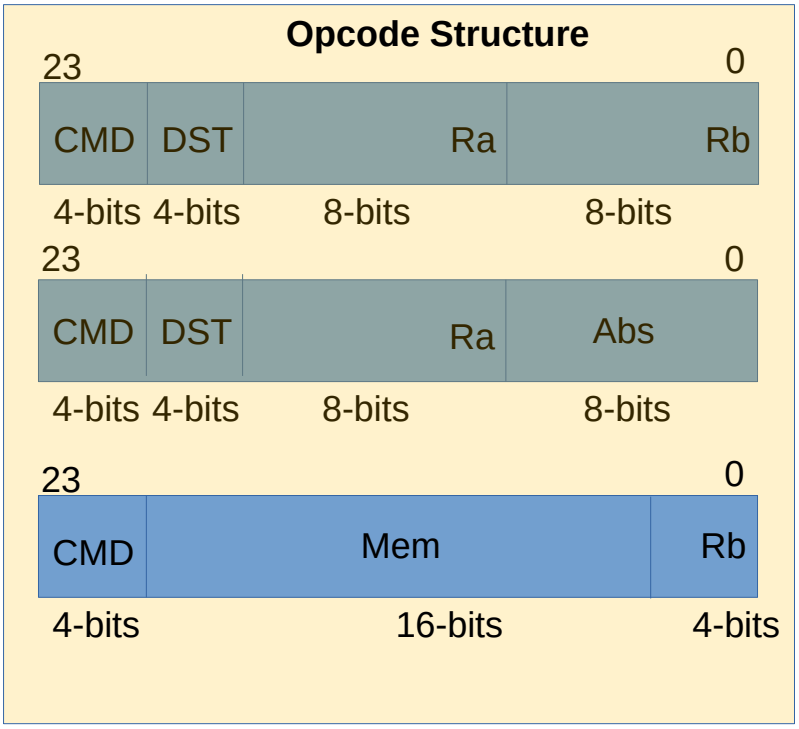

Figure 1: Opcode Structure.

Instruction[1]	Opcode	Description
ADDRR Ra,Rb,Ry	0000RyRaRb	Ry ← Ra + Rb
SUBRR Ra,Rb,Ry	0001RyRaRb	Ry ← Ra - Rb
MULRR Ra,Rb,Ry	0010RyRaRb	Ry ← Ra × Rb
XOR Ra, Rb, Ry	0011RyRaRb	Ry ← Ra$\hat{}Rb$
INV Ra, Ry	0100RyRaXX	Ry ← ∼ Ra
AND Ra, Rb, Ry	0101RyRaRb	Ry ← Ra & Rb
OR Ra, Rb, Ry	0110RyRaRb	Ry ← Ra | Rb
JMP	0111XXRaXX	PC <- Ra
NOP	1000XXXXXX	No Operation (Idle)
RESERVED	1001XXXXXX	N/A
LD R, Mem	1010MMMMRb	Rb ← [Mem]
ST Rb, Mem	1011MMMMRb	[Mem] ← R
ADDRA Ra,#Imm,Ry	1100RyRaIM	Ry ← Ra + #Imm
MULRA Ra,#Imm,Ry	1101RyRaIM	Ry ← Ra × #Imm
RESERVED	1110XXXXXX	N/A
RESERVED	1111XXXXXX	N/A

The pinout of the CPU is as follows:

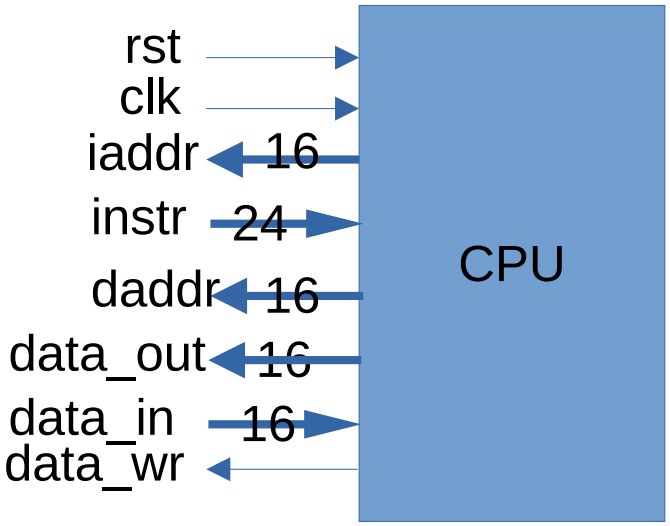

Figure 2: Pinout

The CPU should have 16 24-bit registers (R0,...R16). Your CPU should have the following components (can be in the same Verilog module or instantiated as separate modules):

- Instruction Fetch (IF) for instructions from the memory.

- Instruction Decode (ID) for converting the instruction to usable internal signals.
- Arithmetic Logic Unit (ALU) for doing the arithmetic.
- Write Back (WB) for writing the result of an operation back to the registers.
- Memory (Mem) for writing to memory.

PART II: Memory

Implement the Instruction Memory and Data Memory using Block RAMs. The Instruction Memory should be able to store 32K instructions, and the Data Memory should be able to store 16K 24-bit words.

PART III: Memory-Mapped Peripherals

Implement the following memory-mapped peripherals:

- A 24-bit Timer that counts down from a given value to zero.
- A 16-bit integer divider. The result is stored in two registers (one for the quotient and one for the remainder).

PART IV: Testing

Write testbenches to test the CPU, the memory, and the peripherals. The testbenches should include tests for all the instructions in the instruction set, as well as tests for the peripherals. Your code should execute from the instruction memory and should be able to read and write to the data memory.

Deliverables

- **Submit your homework to Canvas as a single zip file, which has all the verilog code and a pdf report detailing your design, testbenches, and should include screenshots of all the relevant simulation results.**