Yueqiao Wang, Work Log

Milestone 1

Dec 21

Group Meeting in L210

- Confirmed 16-bit size.
- Designed registers, memory allocation, Core Instruction Formats
- Finished most instructions.

Jan 2

Time: 1:00 AM - 3:00 AM
Location: Home
Solo Work

- Verified, revised, fixed, optimized, and commented on the algorithm.
- Formatted the document.

Jan 9

Time: 1:45 PM – 5:15 PM
Location: L226
Meeting with Team & Solo Work

- Machine Code Translation
- Developed the Introduction Philosophy section
- Discussed Performance metrics
- Formatted the final document
- Submission process completed

Week 5 Meeting

Thu Jan 11
Time: 9:00 AM – 9:50 AM
Location: Library

- Meeting Plan
- M2 Requirements Discussion

Fri Jan 12
Time: 9:25 AM –

Things need to be fixed:
Philosophy
- Should the decision reason, target. Short

Performance :
- need to have a comparison
  - Ex. Compared to RISC-V

How the UI is processed
- But it should be reset during
- Automat hardware store
- OR() need to be ensured.

More Code Example
Minimum
Recursion
Things with UI

Missed Calling convention

Milestone Progress

Jan 13

Time: 5:10 PM – 6:50 PM
Location: NAB
Done:

Distribute the task.
Edition to Reg
Will Move the PC to specific reg
reason: PC is special and give more space.
No Need for Memory Data Register (MDR) right now
Memory Data Register (MDR) is a register used for holding information that is in the process of being transferred from the memory to the central processor, or vice versa1. It holds data that is being transferred to or from memory2.
Done directly

Jan 14

Individual Work Period:

Time: 12:45PM – 2:30PM (45 min break included)
Location: Home:
Done:

Create team channel for today's meeting notes.
Change OR(UI,imm) to IG (UI, imm), indicates that the Immediate Generator will format the immediate based on the available imm digit in instruction and the UI register.
RTL for lw sw jalr beq blt bne bge.

Question I have:

1. PC store current index of instruction of next instruction, should we increase the PC firstly or lastly
    1. this will influence:

1. jalr Jump And Link Reg 2RI 1010 001 **R[rd] = PC + 2**; PC = R[rs1] + IG(UI,imm)

2. Programable Registers only have one pair of input and output, so it can only process write or read one reg
    1. input_A = Reg[inst[12:10]] input_B = Reg[inst[9:7]] won't work
3. Might need to change the memory to instruction memory and data memory
    1. the component can only be reached once per cycle
4. we can let PC to do the 2*imm because it distributed the work more reasonably.
    1. Immediate Generator give the difference, PC 2*the difference
    2. normal increment, give 1, PC += 2*1;
    3. for jalr: R[rd] = PC + 2; PC = R[rs1] + IG(UI,imm)
        1. let "R[rs1] + IG(UI,imm)" be the index of instruction instead of the address. and let the PC *2
    4. * or have a 2*(A+B) in ALU
        1. then need 2 cycle for branch

Team Meetings:

Time: 4:00 PM – 7:00 PM
Location: Wabash Starbuck & NAB

Done:

1. write the behaviors description
2.

Decision:

PC store the current index of instruction, so we need to update the PC at the end of the instruction.

PC will do the *2 and addition stuffs.

So ALU can do operation for bench.

PC will integrate PC_isbranch[0:0] and PC_set[0:0] like and gate

Added output_branchType[1:0]: to determine which branch it should happen
Sperate Memory because each component can only be accessed once per cycle.

Jan 15

Individual Work Period:
Time: 4:30 PM – 6:00PM
Location: D115

Done:

Format and submit the DesignDocument

Personal Work Log

Question:

The naming conventions is missing.