Work Log

❕ Work log by Yueqiao, WYQWillSleep
Visit
https://wic.monster/worklog/lime

Dec 21
Jan 2
Jan 9
Jan 13
Jan 14
- Individual Work Period:
- Team Meetings:

Jan 15
Jan 17, 2024
- Time: 7:00 PM – 10:00 PM
- Location: NAB J101
- Done:
- Decision:
- Question:

Jan 20, 2024
- Time: 4:00 PM – 7:00 PM
- Location: NAB J101

- Done:

Jan 20, 2024

- Team Meeting
  - Time: 11:00 AM – 1:00 PM
  - Location: NAB J101
  - Done:
  - Todo:
- Personal Work Time
  - Time:
  - Done:
  - Questions:

Jan 23, 2024

- Teamwork Time:
  - Things Comes Up:
- Personal Work time:
  - Things Comes Up:
  - Problems:
  - Notes:
  - Submission Status:

Jan 28, 2024

Jan 30, 2024

Feb 1, 2024

Feb 2 meeting

Feb 4

- Personal Work
- Meeting

Feb 5

- Meeting

Feb 11

Feb 13

Feb 14

Feb 15

Feb 16

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
    a. this will influence:
        i. 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
    a. 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
    a. 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.
    a. Immediate Generator give the difference, PC 2*the difference
    b. normal increment, give 1, PC += 2*1;
    c. for jalr: R[rd] = PC + 2; PC = R[rs1] + IG(UI,imm)
        i. let "R[rs1] + IG(UI,imm)" be the index of instruction instead of the address. and let the PC *2
    d. or have a 2*(A+B) in ALU
        i. 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.

Weekly Meeting 6

Rising Edge

Double check the reading edge

ALU:

Don’t need clk

Clk for reg/data storage only.

Reformat the RTL:

Make the summary to fit the

Get the comment out

IG naming

Name Convention:

how to name things.

RTL Testing Method

One of each type that run edge

M3

Jan 17, 2024

Time: 7:00 PM – 10:00 PM

Location: NAB J101

Done:

1. Draw Datapath for multiple cycle.
    a. So we get component, lists
    b. Control signal

Decision:

Multicycle: seems much better

MISSING SOMETHING

Question:

How to perform PC += 2*imm

1. Two cycle
2. Change PC to a com that have its individual ALU

Jan 20, 2024

Time: 4:00 PM – 7:00 PM

Location: NAB J101

Done:

1. Why Multiple Cycle
    a. I wrote the statement
2. Which components need Clk signal:
    a. Things that store data

MISSING SOMETHING

Jan 20, 2024

Team Meeting

Time: 11:00 AM – 1:00 PM

Location: NAB J101

Done:

RTL for multicycle

Double checked RTL instruction with clock signal

Todo:

Test method, check based on clock signal

Personal Work Time

Time:

1:00 PM – 2:00 PM, 4:00 PM – 6:00 PM

Done:

Quartus project

Write most code

Added Clk signal for Programmable Register File

Reg need clock signal to store?

But do they?

Changed output for Instruction Register, Simple Register

Input/output_ components_Name

Make more sense, should be the naming convention

Questions:

How to represent a variable in Veriolog, “logic is not declared”

Is real Verilog Testing file required for M3?

Jan 23, 2024

Teamwork Time:

Things Comes Up:

Change the simple reg update at falling edge.

More efficiency

Personal Work time:

Revise Datapath

Format Design Document

Revise Name, behaviors,

Multiple Cycle Component Descriptions Summary

Things Comes Up:

Should we have memRed control signal?

This signal might help us to fetch the instruction ready early.

Like before the raising edge of first cycle, so we can process data earlier

While making sure that the store word instruction can be done.

More precise check for time signal edge & cycle delay

Check control unit, it might give control signal after cycle 2.

2	A = Reg[IR[12:10]] B = Reg[IR[9:7]]	Reg(IR[15:13]) = ALUOut PC = ALU(*,2,IG([9:7]))	A = Reg[IR[12:10]] B = Reg[IR[15:13]]	PC = ALU(*,2,IG([12:3]))

Might not work,

Do we have to have clock signal for control unit?

Problems:

Missing Test Cases

Test descriptions for each step in your integration plan. Be specific enough so that each stage will be tested thoroughly.

Be sure you are keeping things you added earlier up to date if you make changes.

Control unit specifications (inputs, outputs, etc) in your selected implementation's components list.

Notes:

Verified the use of reg in Verilog, clarifying that it won't necessarily function as a register.

Submission Status:

Submitted 2 hours late; others have already submitted.

Missed Week 7 Meeting

M4

Jan 28, 2024

Question:

Need to confirm with Dr. W:

An updated design process journal.

Jan 30, 2024

Done:

Explain the Verilog writing to team member and distribute the work

Schedule the upcoming meeting

Feb 1, 2024

Location:

CSLab

Work:

Some Format Stuffs

Decide:

Add the BranchType, Zero, and Negative Flag as input of PC.

PC should be able to process the when doing branch

Added BranchType as output of control:

Tell which branch we are going to run.

Feb 2 meeting

Add reset,

Update RTL

Divided

Feb 4

Personal Work

Catching up the control unit design

Done:

Added output_control_mem2Reg[0:0] to the control, used missed

Revised the Component Descriptions

Revised the stated machine

Meeting

Location: J101

Time: 11:00 AM – 13:00 PM

Topic:

Divide the Datapath to multiple group

how to do Verilog Module Instantiation

Feb 5

Meeting

Location J101

Time: 6PM – 10PM

Helped the team with all the Verilog stuffs:

Make sure the logic and syntx is right

Personally working on verify the status diagram

Feb 11

Finished PC, PCtb (used to be nazil

Added mux before the Programmable registers since the branch type need reg[rd]

Changed simple reg, now it will keep the data until new data is in.

Feb 13

Working on Control Unit:

Find problem for ALU:

Missing 2* operation, added

```
two_star = {1'b0, add_result}; // Multiply sum by 2
```

> ❕ Check is required

During the process:

ALU shift are not checked at all.

ALU shift do:

```
3'b010: // Bitwise Shift
    output_ALU = input_A << input_B;
3'b011: // Arithmetic Bitwise Shift
    output_ALU = (input_B[2]) ? (input_A >> input_B) : (input_A << in
```

```
Bitwise Shift (3'b010):
This operation shifts the bits of input_A left by the number of posit
It performs a logical shift, meaning it shifts in zeros from the righ
For example, if input_A is 1010 (decimal 10) and input_B is 2, the re
Arithmetic Bitwise Shift (3'b011):
This operation performs an arithmetic right shift if the most signifi
An arithmetic right shift fills the vacated MSB positions with the va
For example, if input_A is 1101 (decimal -3) and input_B is 2, the re
If input_A is 1010 (decimal 10) and input_B is 2, the result will be
```

Feb 14

changed All 3R, 2RI, RI type’s func 4 for better ALUOp prepare

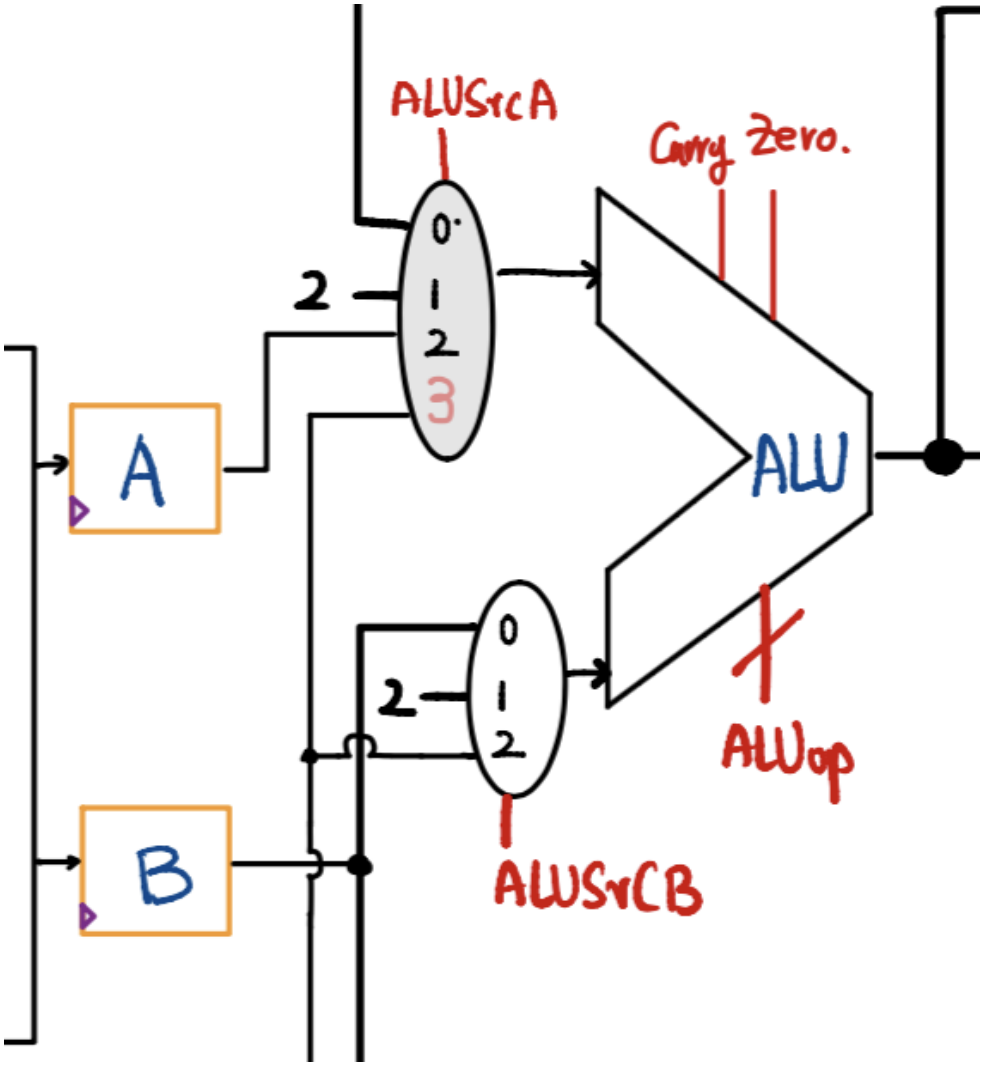
control unit

have to change RTL, Datapath, Control

PC=ALU(*,2,IG([9:7])) cannot be achieved

don’t have times, so have to work on shift

so 2 << IG don’t work



Feb 15

done control unit, complied

writing control tb

problem exist when dealing with lw

Feb 16

8:40am writing control tb mentioned the problem to teams

8:50am working on doc