1092 Digital System Design Final Project

Pipelined RISC-V Design

Announced at May 20, 2021

TA Information

Baseline/ Extension: 馬咏治

kane@access.ee.ntu.edu.tw

1. Project Description

Table 1. Required Instruction Set

Name	Description
ADD	Addition, overflow detection for signed operand is not required*
ADDI	Addition immediate with sign-extension, without overflow detection*
SUB	Subtract, overflow detection for signed operand is not required*
AND	Boolean logic operation
ANDI	Boolean logic operation with 12bit of immediate
OR	Boolean logic operation
ORI	Boolean logic operation with 12bit of immediate
XOR	Boolean logic operation
XORI	Boolean logic operation with 12bit of immediate
SLLI	Shift left logical (zero padding)
SRAI	Shift right arithmetic (sign-digit padding)
SRLI	Shift right logical (zero padding)
SLT	Set less than, comparison instruction
SLTI	Set less than variable, comparison instruction
BEQ	Branch on equal, conditional branch instruction
BNE	Branch on not equal, conditional branch instruction
JAL	Unconditionally jump and link (Save next PC in $rd)
JALR	Jump and link register(Save next PC in $rd)
LW	Load word from data memory (assign word-aligned)
SW	Store word to data memory (assign word-aligned)
NOP	No operation(addi $r0 $r0 0)

* Different from definition in [1], the exception handler for arithmetic overflow is not required.

In final project, you are asked to design a **pipelined RISC-V processor (synchronous active low reset) with instruction cache and data cache**. This processor should at least support the instruction set defined in Table 1. The instruction set is referenced from Chapter2 (RV32I base integer instruction set) of [1], and we encourage you read it in detail.

The whole module hierarchy is shown in Figure 1. And the processor architecture is in Figure 2. As you see, this is modified from single-cycle architecture of our HW3. Your design should follow this **5-stage pipelined architecture**. You need to modify several parts to fit our specifications. For example, you need to add the path for **J-type instructions**.

Also, you should **solve the hazards** by adding some circuits. There are 3 hazard categories should be properly handled in your pipelined processor:

1) Structure hazard
2) Data hazard
3) Branch hazard

Although all of these hazards can be solved by insert NOP manually or automatically in your test program, we ask you to implement **data forwarding unit** and **pipeline stall unit** to solve these hazards.

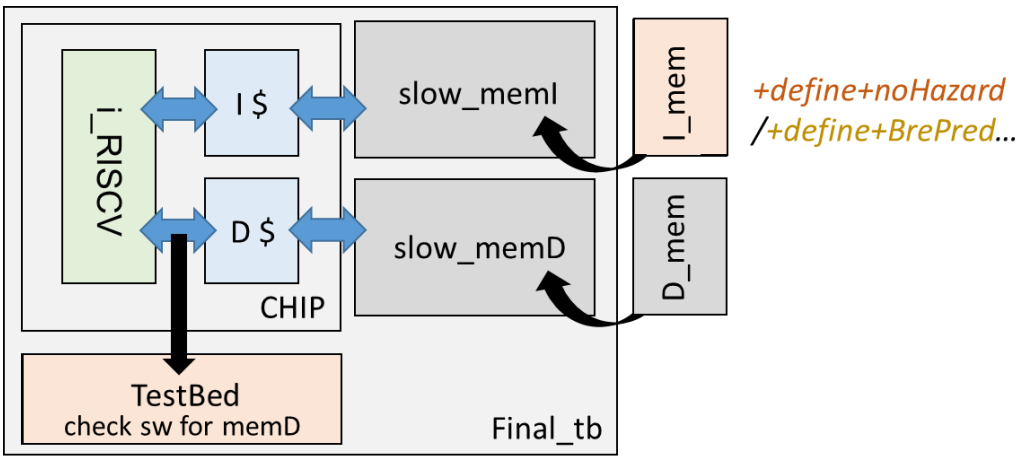

Figure 1. Module Hierarchy

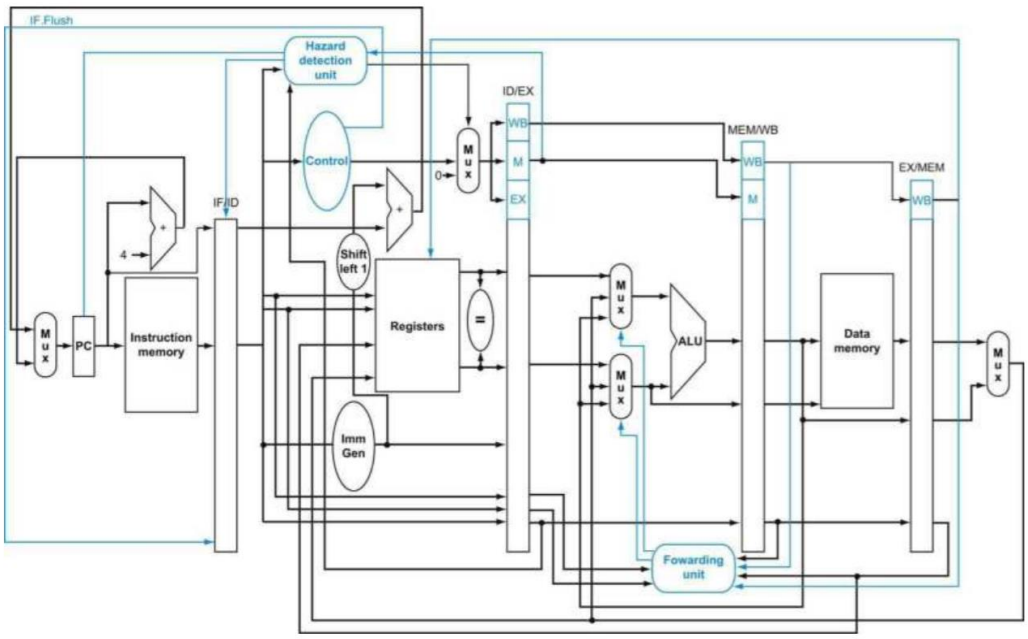

Figure 2. Simplified Pipeline Architecture of RISCV

2. Cache and Memory Interface

The instruction memory and data memory will not be contained in your design. The memory interface is left as module I/O. You have to use the provided slow memory model. **Do not synthesize the slow memory.**

The cache units are suggested to have the same block number (8) and block size (4) as in HW4. Besides, we do not restrict the replacement policy and writing policy of the cache design. You are encouraged to optimize the cache units to fit your RISC-V design.

3. Synthesis Notes

You should synthesize your design using TSMC 0.13 cell library, and the relevant files, e.g. *.synopsys_dc.setup*, can be copied from previous HW or use the attached file. The design constraints are specified in "*CHIP_syn.sdc*". Note that the pipelined RISC-V, instruction cache and data cache are included in the *CHIP.v*. They should be synthesized together.

The post-synthesis simulation is required and all involved Verilog files should be all modeled by gate-level. Note that the maximum clock frequency must be verified by post-synthesis gate-level simulation. And you are recommended to buffer the input signal to avoid timing violation.

4. Grading Policy and Possible Extensions

All grades of this project consist of three equal aspects:

1) Proposal: Project Check Point (5%)
2) Presentation (45%)
3) Final report: Technical features (50%)

The technical features are grading by two parts:

1) Baseline; 2) Extension. The details are as follows.

Baseline (50%)

If your design meets all requirements of above description, and can be synthesized and simulated at gate-level, your design will be qualified to get baseline points. The solid requirements include:

1) Supporting all instructions above
2) With caches
3) Pass all test assembly programs
4) Complete the circuit synthesis. Note that the slack cannot be negative.

Then the performance is evaluated by (the smaller the better):

Area (um^2) * Total simulation time (ns);

And baseline points will be based on your AT ranking among other RISCV teams.

Extension (50%)

There are two topics of extension.

1) Branch prediction mechanism.
2) L2Cache
3) Supporting compressed instructions.

Implement **as much and deep as you can** of the topics of extension.

5. Simulation Example

[Simulation]

```
source /usr/cad/cadence/cshrc
source /usr/spring_soft/CIC/verdi.cshrc
```

RTL

```
ncverilog Final_tb.v CHIP.v slow_memory.v +define+noHazard +access+r
```

Gate level

```
ncverilog Final_tb.v CHIP_syn.v slow_memory.v tsmc13.v
+define+noHazard +define+SDF +access+r
```

6. Submission Requirement

All the files need to be compressed as a single **ZIP file** and **upload it to CEIBA of the team leader**.

Example of filename

DSD_Final_Project_RISCV_G#.zip
DSD_Final_Project_RISCV_G#_v#.zip
e.g. DSD_Final_Project_RISCV_G1_v2.zip

Your submitted file should include the following files:

DSD_Final_Project_RISCV_G#/
- *Src/*
  - *Baseline/*
    - *rtl/*
      - CHIP.v
    - *syn/*
      - CHIP_syn.v
      - CHIP_syn.sdf
      - CHIP_syn.ddc
  - *Extension/*
    - *BrPred/*
      - *rtl/*
      - *syn/*
    - *Compression/*
      - *rtl/*
      - *syn/*
- DSD_Final_Project_Scores_RISCV.pdf
- Presentation.pptx
- Report.pdf
- **Readme.txt**

The homework will be graded **ONLY IF** the filename of your submission is correct!

7. Schedule and Necessary Submissions

Date	Submission/Event
5/20	Final project announcement
5/19	Find your teammates and choose MIPS or RISCV (Fill out the Google form)
6/03	A. Checkpoint. Each team should prepare a proposal (*4-6 pages Powerpoint (about 5 minutes)*) to confirm your current results and future plan. You should upload the team proposal to the CEIBA by the day. B. Extension topics plan should be included in the proposal C. Please attach work assignment chart at last page
6/17	Final presentation. Each team should prepare a full talk (*within 15 minutes*, about 10-20 slides) to demonstrate your fantastic work! Detail presentation plan will be announced in the ceiba.
6/24	Final submission, including a detailed report (*8-16 pages*), the presentation slides, and all the source codes (including all the RTL code and synthesis related files: *.v, *.sdf, *.ddc and a Readme.txt). You should upload the final submission to the CEIBA by the day.

8. Reference

[1] https://riscv.org/specifications/