NYCU-DCS-2024

Final Project

Design: Optimized CPU Design

Data Preparation

1. Extract files from TA's directory:
   % tar xvf ~DCSTA01/FP.tar
2. The extracted LAB directory contains:
   a. **00_TESTBED**
   b. **01_RTL**
   c. **02_SYN**
   d. **03_GATE**
   e. **09_SUBMIT**

Design Description

About central processing unit (CPU), it is referred in the HW04.

In this design, the single cycle CPU you designed in the HW4 need to be optimized. All the method you can come up with is allowed, such as multi cycle, pipeline, or others.

A 32-bits CPU with a data memory and an instruction memory is designed with your optimized method. Each instruction carries out different operations based on MIPS architecture. Finally, the result of your operations is stored in a certain memory address.

Functional Description

The CPU you designed needs to have at least seven functions: add, addi, beq, j, lw, slt, sw.

About MIPS architecture, please look up the MIPS reference sheet by yourselves:

1. https://inst.eecs.berkeley.edu/~cs61c/resources/MIPS_Green_Sheet.pdf
2. https://uweb.engr.arizona.edu/~ece369/Resources/spim/MIPSReference.pdf
3. https://www.kth.se/social/files/563c63c9f276547044e8695f/mips-ref-sheet.pdf

Input

The input signals for Verilog are as follows:

Input Signal	Bit Width	Description
clk	1	Clock signal
rst_n	1	Asynchronous active-low reset
data_read	32	The data which is read from the data memory
instruction	32	The instruction which is read the instruction memory

1. When the **data_wen** is low, the **data_read** is the data from the data memory based on the **data_addr**.
2. When the **inst_addr** is given, the **instruction** is read from the instruction memory which is read only.

Output

The output signals for Verilog are as follows:

Input Signal	Bit Width	Description
data_wen	1	Be high when the data is written into the data memory; be low when the data is read from the data memory.
data_addr	32	The address of data memory for writing or reading the data
inst_addr	32	The address of instruction memory for reading the instruction
data_write	32	The data which is written into the data memory

1. If the **data_read** wants to read the specific data from a certain location in the data memory, the **data_wen** must be low and the certain **data_addr** is given.
2. If the **data_write** wants to write the specific data into a certain location in the data memory, the **data_wen** must be high and the certain **data_addr** is given.
3. The **instruction** is read from the instruction memory based on the **inst_addr**.

SPEC

1. Verilog
   i. Top module name: CPU (File name: CPU.v)
   ii. It is **positive edge trigger** architecture.
   iii. It is **asynchronous reset** and **active-low** architecture. If you use synchronous reset (considering reset after clock starting), you may fail to reset signals.
   iv. The reset signal (**rst_n**) would be given only once at the beginning of simulation.

v. The execution latency is limited in **100000 cycles**. The latency is the clock cycles between the rising edge of the **rst_n** and the ending of the simulation.

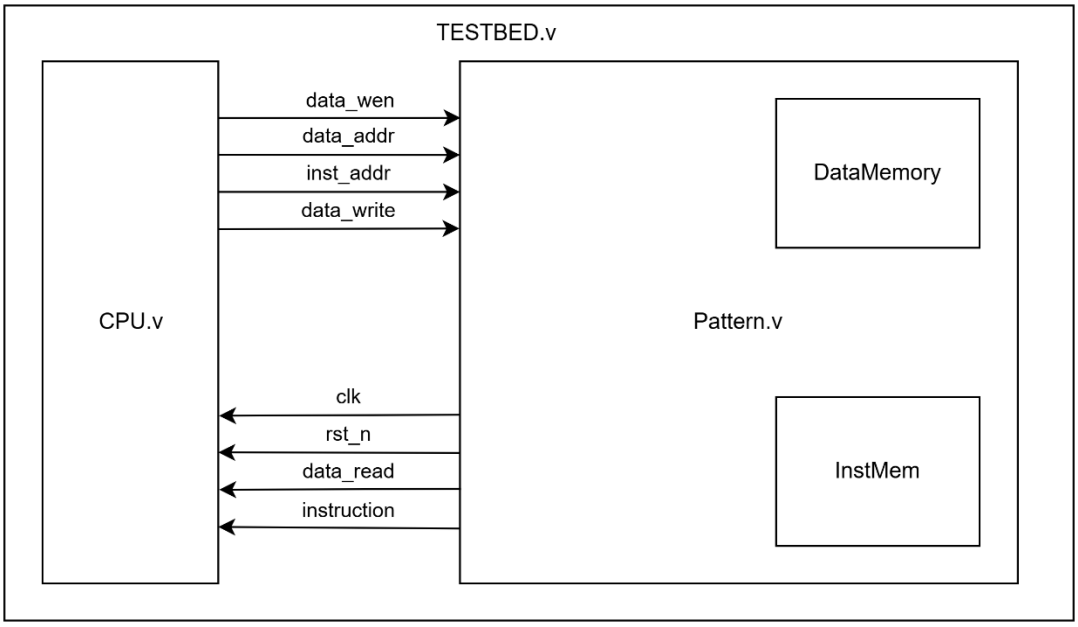
vi. The synthesis result of data type **cannot** include any **latches** (using ctrl+F to find the term of "Latch" or using the command **./08_check** in 02_SYN/)**.**

vii. The clock time for single cycle CPU **cannot be larger than 50 ns**.

viii. After synthesis, you can check CPU.area and CPU.timing. The timing report should be **non-negative (MET)**.

ix. After gate-level simulation, the file of vcs.log cannot include any **timing violation**.

Block Diagram



Homework Upload

1. Please use the commands we provided to tar whole file under 09_SUBMIT/ and to submit your homework **before 23:59:00 on 6/23 (Sun.)**.
   A. If homework isn't handed in **after 23:59:00 on 6/23 (Sun.)**, the score is **zero point**.
   B. Noticed that the time **is only based on** the demo result you submit with 09_SUBMIT. It is not allowed that the homework handed in late, even if the time is 23:59:01 on 6/23 (Sun.).
2. Please check the homework file again after you upload it. If you upload the wrong file, you will **fail** this homework.
3. Please submit the report file to the new E3. The name of your report file is **report_DCSXXX.pdf (XXX is your account ID)**. If the file violates the naming

rule and the file format, **10 points will be deducted.**

Grading Policy

1. Verilog Function Validity: total 30%
   - RTL, SYN and GATE: 30%
     - The score **is only based on** the demo result you submit with 09_SUBMIT. Please ensure that you successfully upload the latest version.
     - Please check whether there is any wire/reg/submodule being named as "error", "fail", "pass", "congratulations", "latch". All letters used in the formats of uppercase or lowercase is prohibited. If there is, you will **fail** this homework.
2. Report: total 70%
   - Your report must contain:
     1. The optimized design for CPU (50%)
        I. The method in your design
        II. The problem encountered in this project and how the problem was solved
     2. Please make a comparison between the CPU in the HW04 and the one in this final project. (20%)
3. Platform Architecture (Bonus): total 15%
   - Please submit the screenshot of Platform Architecture (PA) **in your report** to the new E3.
     - The screenshot of result includes the simulation of Platform Architect (PA) and your account information

Command List

- Verilog RTL simulation (01_RTL/):
  - **./01_run_vcs_rtl**
- Synthesis (02_SYN/):
  - **./01_run_dc_shell**
  - **./08_check**
- Gate level simulation (03_GATE/):
  - **./01_run_vcs_gate**
- Submit your files (09_SUBMIT/):
  - **./00_tar XX**
    - XX is cycle time in this design.
  - **./01_submit**

- **y**
  - You must type '**y**' when you ensure you are ready to hand in your homework.
- **./02_check**

❖ Waveform for debug:

- **nWave &**
- find *.fsdb
- shift+L for reloading the *.fsdb file after you simulate your design again.

Note

1. You can modify TESTBED.v and PATTERN.v, but we will demo your design through TA's TESTBED.v and PATTERN.v.
2. Methods to avoid latches: (1) Avoid multiple driven. (2) Ensure all if-else statements and case with full case. (3) Avoid combinational loops.
3. You can add any instructions that you need in the instruction.dat for this design if necessary.
4. You can modify the cycle time in your design. However, its value cannot be larger than 50ns.
5. In this Final Project, the folder for Platform Architecture isn't prepared, you must set up the environment (including any design files) by yourselves. Please don't privately ask TA's anything about Platform Architecture with any way. If you do this, **20 points will be deducted.**