FAET630004: Computer & AI Architecture **(Due: 04/14/19)**

Homework Assignment #2

Instructor: Chixiao Chen *Name:* , *FudanID:*

- This HW counts 15% of your final score, please treat it carefully.
- Please submit the electronic copy via mail: cxchen@fudan.edu.cn before 04/14/2019 11:59pm.
- It is encouraged to use LaTeX to edit it, the source code of the assignment is available via: https://www.overleaf.com/read/xwppxvvsnwgw
- You can also open it by Office Word, and save it as a .doc file for easy editing. Also, you can print it out, complete it and scan it by your cellphone.
- Problem 2 needs verilog/SV simulation. If you do not have a local verilog simulator, please use an online tool: https://www.edaplayground.com/, you need register for save.
- You can answer the assignment either in Chinese or English

Problem 1: Hazards in Pipelined Architecture

(15+15=30 points)

Given the five-stage pipeline RV32I architecture, the following assemble codes are going to be executed.

```
addi s0, s0, 1
addi t0, t0, 4
lw   t1, 0(t0)
add  t2, t1, x0
```

(a) How many hazards will occur if no forward logic ? Please point out the hazard type. Among these, is there any hazard can not be fixed by forwarding?

(b) As a *smart* compiler guy, can you re-order the instruction to avoid the stall inserted?

Solution:

Problem 2: Verilog/System Verilog for a Single Cycle Processor (15+5+30+20=70 points)

Assuming you are asked to design a specific processor to implementing matrix-vector multiplication based on the basic RV32 architecture. In other words, perform the following computing in a single cycle processor,

$$\underbrace{\begin{bmatrix}1 & 2 & 3\end{bmatrix}}_{1\times 3}\cdot\underbrace{\begin{bmatrix}2 & 1 & 3\\ 3 & 3 & 2\\ 4 & 1 & 2\end{bmatrix}}_{3\times 3}=\underbrace{\begin{bmatrix}1\cdot 2+2\cdot 3+3\cdot 4\\ 1\cdot 1+2\cdot 3+3\cdot 1\\ 1\cdot 3+2\cdot 2+3\cdot 2\end{bmatrix}=\begin{bmatrix}20\\ 10\\ 13\end{bmatrix}}_{1\times 3}$$

Figure 1: A matrix-vector product example

Register `s0` is the base pointer to save the following sequence $\{1, 2, 3, 2, 3, 4, 1, 3, 1, 3, 2, 2\}$ in words (32bit). Register `t0` is the base pointer to store the output results. For the multiplication, please refer to the RV32M extension in appendix I. For simplicity, we only use the unsigned multiplication instruction `MUL` here.

(a) Write assembly codes to implement the matrix-vector product in Fig. 1 based on RV32I/RV32M instruction set. (hint: please use branch to minimize the code length.)

(b) Tailor a specific instruction set which only needs to support the above code. Use a table to illustrate all the instructions and their opcode/operand fields.

(c) Write an RTL code to realize **all** the instructions used in the above table. (hint: no need for pipeline here, and reuse the code in HW#1.)

(d) Write a test bench to perform the matrix-vector product in Fig. 1.

Appendix I:

RV32M Standard Extension

0000001	rs2	rs1	000	rd	0110011	MUL
0000001	rs2	rs1	001	rd	0110011	MULH
0000001	rs2	rs1	010	rd	0110011	MULHSU
0000001	rs2	rs1	011	rd	0110011	MULHU
0000001	rs2	rs1	100	rd	0110011	DIV
0000001	rs2	rs1	101	rd	0110011	DIVU
0000001	rs2	rs1	110	rd	0110011	REM
0000001	rs2	rs1	111	rd	0110011	REMU

Figure 2: RV32M Instruction Set Extension