Computer Architecture Homework 4

110062219

May 18, 2023

Contents

1 Stuck of Single-Cycle Processor

1.1 `RegWrite`

- `beq`
- `sd`

1.2 `MemRead`

Though `MemRead` of all instruction except `ld` is 0 according to the lecture slide provided by professor, so long as `MemtoReg` and `RegWrite` are correct, `add`, `sub` and `beq` would not be affected. Moreover, if our memory supports read and write simultaneously, then `sd` might still operate correctly.

1.3 $\texttt{ALUOp}_1$

- `sub`

The `ALUOp` of R-type instructions would be `00` and the ALU would perform addition. Therefore, `add` is still correct.

2 Instruction `0xFF4288E3`

The instruction represents the assembly `beq t0,s4,-16`.

2.1 Values of Signals

Table 1: Values of Control Signals of Branch Instruction

Branch	MemRead	MemtoReg	ALUOp	MemWrite	ALUSrc	RegWrite
1	0	Don't Care	01	0	0	0

2.2 Inputs of ALU

Registers `t0` and `s4`, i.e., Reg_5 and Reg_{20}.

3 Shortest Possible Clock Time

The critical path is load instruction: `PC` $\rightarrow$ `IM` $\rightarrow$ `Reg` $\rightarrow$ `MUX` $\rightarrow$ `ALU` $\rightarrow$ `DM` $\rightarrow$ `MUX` $\rightarrow$ `Reg`. Since we're writing back to register, we should delay for setup time. Thus the shortest possible clock time is $30 + 200 + 140 + 30 + 160 + 200 + 30 + 20 = 810$ps.

4 Structural Hazard

4.1 Stalls

Before we fetch an instruction, we first check that whether `MemRead`, `MemWrite` in `EX/MEM` pipeline register is set or not, i.e., the instruction in `MEM` stage is `ld` or `sd`. If so, then we have to stall and not to fetch the instruction.

Notwithstanding, suppose that our memory supports read and write simultaneously, then even though `MemWrite` in `EX/MEM` pipeline register is set, we're still able to fetch the incoming instruction. So in this case, we only have to stop if the instruction in `MEM` stage is `ld`.

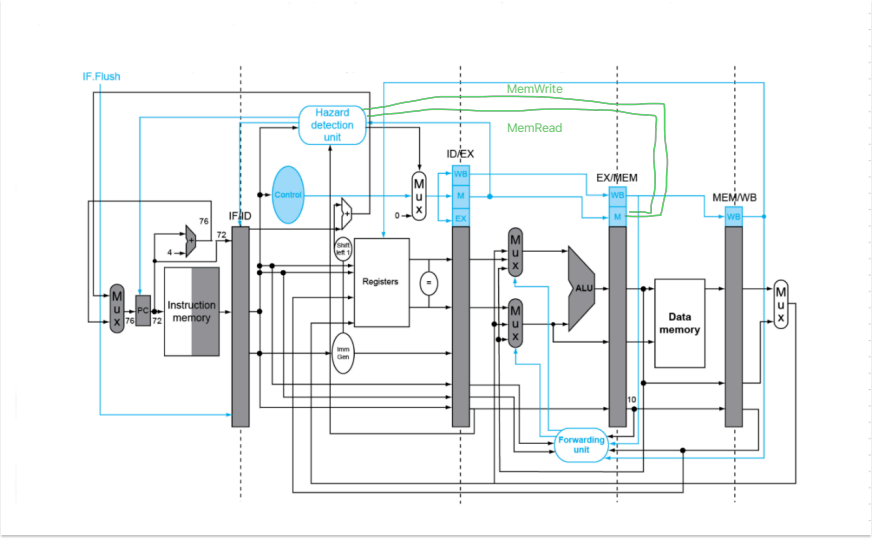

Figure 1

4.2 Tables

Table 2: If memory could not read and write simultaneously

Cycle	1	2	3	4	5	6	7	8	9	10	11	12	13	14
sd	IF	ID	EX	MEM	WB									
ld x31		IF	ID	EX	MEM	WB								
ld x28			IF	ID	EX	MEM	WB							
add				stall	stall	stall	IF	ID	EX	MEM	WB			
beq								IF	ID	EX	MEM	WB		
and									IF	ID	EX	MEM	WB	
or										IF	ID	EX	MEM	WB

Table 3: If memory could read and write simultaneously

Cycle	1	2	3	4	5	6	7	8	9	10	11	12	13
sd	IF	ID	EX	MEM	WB								
ld x31		IF	ID	EX	MEM	WB							
ld x28			IF	ID	EX	MEM	WB						
add				IF	ID	EX	MEM	WB					
beq					stall	stall	IF	ID	EX	MEM	WB		
and								IF	ID	EX	MEM	WB	
or									IF	ID	EX	MEM	WB

5 Data Hazard

5.1 Neither Forwarding Unit Nor Hazards Detection

We should insert 2 bubbles between the second `ld` and the first `add`, and another 2 bubbles between the first and the second `add`.

5.2 Only Forwarding Unit But No Hazards Detection

We should insert a bubble between the second `ld` and the first `add`.

5.3 Both Forwarding Unit And Hazards Detection

5.3.1 Stall on Load-Use Hazard

We would detect the hazard and stall when the second `ld` is in the `EX` stage, which occurred in cycle 4.

5.3.2 `EX/MEM` Forward

We would forward from when `EX/MEM` pipeline register to one of the input of ALU when the second `add` is in the `EX` stage, which occurred in cycle 7.

5.3.3 `MEM/WB` Forward

We would forward from when `MEM/WB` pipeline register to one of the input of ALU when the first `add` is in the `EX` stage, which occurred in cycle 6.

5.4 Both Forwarding Unit And Hazards Detection (cont.)

There are 4 instructions and we have to insert a `nop`, so in total it takes $5 + (4 + 1) - 1 = 9$ cycles.

6 Pipeline Performance

The cycle time is $\frac{10^9}{300 \times 10^6} = \frac{10}{3}$ns.

6.1 7-stage Pipeline

$S = 30 \times \frac{3}{10} - 7 + 1 = 3$, thus

$$30 + 4 \times 3 \times \frac{10}{3} = 70$$

6.2 N-stage Pipeline

$$\begin{cases} N+ \quad (S-1) = \quad 90 \times \frac{3}{10} = 27 \\ N+ \quad (6S-1) = \quad 290 \times \frac{3}{10} = 87 \end{cases}$$

, so we have $S = 12, N = 16$.

7 Branch Prediction

7.1 Always-Taken & Alway-Not-Taken

The accuracy rate:

Always-Taken $\frac{3}{8} = 37.5\%$

Always-Not-Taken $\frac{5}{8} = 62.5\%$

7.2 1-bit Dynamic

Table 4: 1-bit Dynamic Predictor

Predict	Outcome	Correct
T	T	✓
T	NT	
NT	NT	✓
NT	NT	✓
NT	NT	✓
NT	T	
T	NT	
NT	T	

The accuracy rate is $\frac{4}{8} = 50\%$.

7.3 2-bit Dynamic

Table 5: 2-bit Dynamic Predictor

State	Outcome	Correct
Strongly Not Taken	T	
Weakly Not Taken	NT	✓
Strongly Not Taken	NT	✓
Strongly Not Taken	NT	✓
Strongly Not Taken	NT	✓
Strongly Not Taken	T	
Weakly Not Taken	NT	✓
Strongly Not Taken	T	

The accuracy rate is $\frac{5}{8} = 62.5\%$.

8 Exception with Pipeline

Table 6

Cycle	IF	ID	EX	MEM	WB
1	`ld`				
2	`sub`	`ld`			
3	`beq`	`sub`	`ld`		
4	`beq`	`sub`	bubble	`ld`	
5	`sd`	`beq`	`sub`	bubble	`ld`
6	1^{st} instruction of exception handler	bubble	bubble	`sub`	bubble