HW4

author: 111062272 蕭登鴻

1

a

ans	reason
sd	need simm12
ld	need simm12
beq	need simm13 for PC

b

ans: add, beq

c

ans: Branch, RegWrite

signal	value	consequences if set to 1
Branch	0	if (simm12 + rs1 == 0) => PC will be faultily modified
MemRead	don't care / 0	-
MemtoReg	don't care / 0	-
MemWrite	1	-
ALUSrc	1	-
RegWrite	0	will store wrong data in register file

2

a

0x01EEA523: sw x30, 10(x29)

signal	value
Branch	0
MemRead	don't care / 0

signal	value
MemtoReg	don't care / 0
ALUOp	op for "Add"
MemWrite	1
ALUSrc	1
RegWrite	0

b

ans: Reg[29] & 10

3

a

least time-consuming instruction: add, with 600ps

Whose longest datapath is: PC read -> I-Mem -> Register File -> mux -> ALU -> mux -> register setup

b

The minimum clock period is equal to the execution time of most time-consuming instruction.

Therefore, ans: ld, with 835ps

Whose longest datapath is: PC read -> I-Mem -> Register file -> mux -> ALU -> D-Mem -> mux -> register setup

4

notations:

t_1 : time needed to execute an instruction under 4GHz.

t_2 : time needed to execute an instruction under 2GHz.

t_s : time needed for 1 stage under 2GHz.

To achieve maximum speedup, assume that time needed for each stage is equally distributed, i.e, $t_2 = 5 \cdot t_s$

$$t_1 = \frac{4}{4 \cdot 10^9} = 10^{-9}, t_s = \frac{1}{2 \cdot 10^9}$$

Speedup = $\frac{1200 \cdot t_1}{t_2 + (1200-1)t_s} \approx 1.993$

5

a

instruction	dependencies
ld x28, 4(x5)	x28, x5
add x29, x6, x7	x6, x7
sub x30, x28, x29	x28, x29
sd x30, 0(x11)	x30, x11

More specifically, x28 is the dependency between `ld & sub`, x29 is the dependency between `add & sub`, and x30 is the dependency between `sub & sd`.

b

There are none.

Explanation: Since all RAW hazards can be resolved by forwarding, and the only load-use hazard (between instruction 1 & 3) is separated by 1 instruction; so it can also be resolved by forwarding.

c

ans: 4NOPs

Explanation: Between instruction 2&3, 1 NOP is inserted to solve RAW hazard, and 1 is used to solve load-use hazard. Between instruction 3&4, 2 NOPs is inserted to solve RAW hazard.

6

abbreviation:

- T: Taken
- NT: Not Taken

a

always-NT: $\frac{5}{8}$

always-T: $\frac{3}{8}$

b

Ground truth	NT	T	T	NT	NT	NT	NT	T
State	T	NT	T	T	NT	NT	NT	NT
Decision	T	NT	T	T	NT	NT	NT	NT
Correctness	F	F	T	F	T	T	T	F

accuracy: 4/8

c

prefix:

- s: strongly
- w: weakly

Ground truth	NT	T	T	NT	NT	NT	NT	T
State	sT	wT	sT	sT	wT	wNT	sNT	sNT
Decision	T	T	T	T	T	NT	NT	NT
Correctness	F	T	T	F	F	T	T	F

accuracy: 4/8

7

a

5 + 2 (NOP) instructions; 5 stages => 5 + (7 - 1) = 11

2 NOPs is added between `ld` and `add` to resolve structural hazards raised by conflict between `IF` stage of `add & sub` and `MEM` stage of `sd & ld`.

ans: This instruction sequence has an execution time of 11 stages. Furthermore, structural hazards can be resolved by adding 5 NOPs between each instruction such that every instruction won't overlap in pipeline phases, which resulted in meaning lost for pipelining.

b

reordered instruction sequence:

```
beq x11, x12, Label
add x11, x6, x12
sub x11, x13, x12
sd x15, 0(x23)
ld x15, 0(x24)
```

Since `add & sub` won't need `MEM` stage, `IF` stage of `sd & ld` won't conflict with them.

=> Minimum number of cycles = 5 + (5 - 1) = 9

c

if determined in `MEM` stage: [beq, NOP, NOP, NOP, sd, ld, add, sub]: 8

if determined in `ID` stage: [beq, NOP, sd, ld, add, sub]: 6

Speedup = $\frac{5+(8-1)}{5+(6-1)} = 1.2$

d

notations:

t_s: time needed per stage

s: remain stage count

S: total execution time for all stages

$$t_s = \max(210, 160, 220, 180, 100) = 220,\ t'_s = t_s = 220$$
$$s = 5 - 1 = 4,\ s' = s = 4$$
$$S = 210 + 160 + 220 + 180 + 100 = 870,\ S' = 210 + 160 + 220 + 100 = 690$$

Speedup = $\frac{S+t_s \cdot s}{S'+t'_s \cdot s'} \approx 1.11465$

e

$$t_s = \max(210, 160, 220, 180, 100) = 220,\ t'_s = t_s + 25 = 245$$
$$s = 5 - 1 = 4,\ s' = s = 4$$
$$S = 210 + 160 + 220 + 180 + 100 = 870,\ S' = 210 + 160 + 245 + 100 = 715$$

Speedup = $\frac{S+t_s \cdot s}{S'+t'_s \cdot s'} \approx 1.03245$

8

Flush `IF`, `ID` happens at C4.

	IF	ID	EX	MEM	WB
C1	add	-	-	-	-
C2	sub	add	-	-	-
C3	beq	sub	add	-	-
C4	sd	**beq**	sub	add	-

	IF	ID	EX	MEM	WB
C5	EH	NOP	NOP	sub	add