Lecture 9: Control hazards and exceptions

Monday, February 5, 2018   4:07 PM

Outline
- More examples of hazards
- Branch prediction
- Exceptions

Reminder: 3 types of hazards
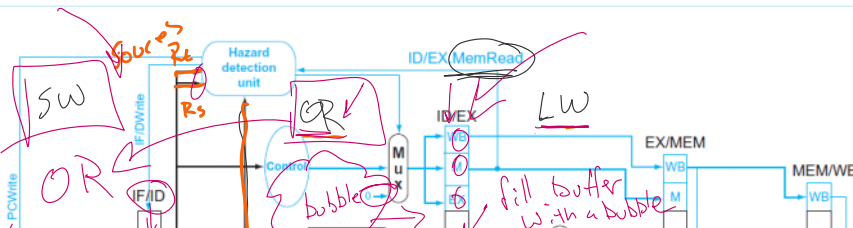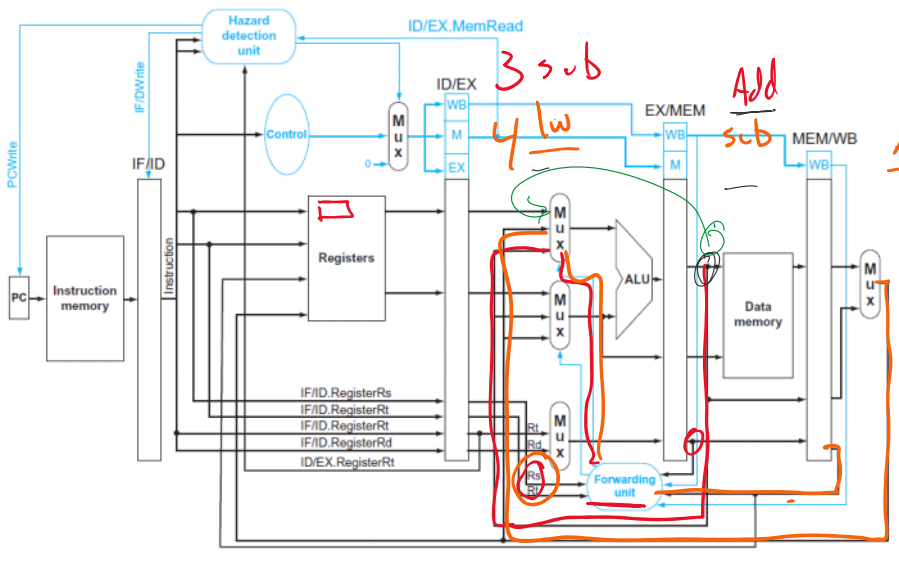
Control → Predict branches

Data → RAW (read after write) data not ready
↳ solve forward or bypass

Structural → add more structures
↳ increase cost
→

Hazard example

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
ADD $1, $2, $3	IF	ID	EX	MEM	WB															
SUB $4, $1, $4		IF	ID	EX	MEM	WB														
LW $2, 100($1)			IF	ID	EX	MEM	WB													
OR $5, $2, $3 not				IF	ID		EX	MEM	WB											
SW $5, 100($1)					IF		ED	EX	MEM	WB										
ADDI $4, $5, 4							IF	ID	EX	MEM	WB									
BEQ $4, $1, -4								IF	ID	EX	MEM	WB								
ADDI $4, $5, 4										IF	ID	EX	MEM	WB						
SW $4, 104($1)											IF	ID	EX	MEM	WB					

Control

Hazard detection and forwarding hardware

3 sub
4 lw
Add
sub
Add

[diagram continued]

sources Rt
SW
OR
Rs
OR
LW
fill buffer with a bubble
bubble

for bubble
insert a nop
into the pipeline

The diagram shows: OR, PCWrite, IF/ID, +4, PC, Instruction memory, Instruction, keep same/no write, Control, Mux, bubble, fill buffer with a bubble, Registers, Mux, Mux, ALU, Data memory, Mux, EX/MEM, WB, M, MEM/WB, WB, Forwarding unit, dest, insert a nop into the pipeline

IF/ID.RegisterRs
IF/ID.RegisterRt
IF/ID.RegisterRt
IF/ID.RegisterRd
ID/EX.RegisterRt

Rt, Rd, Rs, Rt

Reducing control hazards: Branch prediction

```
100: beq        4        ↳ predict if branch is taken or not to reduce stalls for control
   ¦ add
   ¦ sub                 What do we need?
   ¦ OR                  → ✗ something in case wrong prediction
   ¦ sw                          ↳ flush →
120: ¦ add               → remember branch outcomes (maybe)
                         → to know where to go
```

```
             1   2   3   4   5
 → BEQ      IF  ID  EX  MEM  WB
   ADD          IF  ID  EX
   SUB              IF  ID
   OR                   IF
 → 120: add               IF
```

Simple BP scheme
- always taken }   - remember last direction & more
- always not-taken) - count of taken/not

↳ if backwards
      ↳ taken   } easy to impl.
→ if forwards   one bit in inst (sign bit)
      ↳ not taken

branch predictor

PC	prediction
100	taken
200	not taken

insert into table when branch resolved
← 1024 entries
  ↳ 10 bits to index

Problem: aliasing
         two PCs use same predictor

not entire PC
low-order bits of PC

Other indexes into BP table
↳ Global Branch history
[0 1 0 1 1 0 1 0]
         history   pred

for predictor: Saturating counter

taken taken
(10) (11) ← taken
(01) (00) ← not taken
taken

for predictor: saturating county

taken → 11 10

01 00 ← not taken

taken