ECE 2300
Digital Logic & Computer Organization
Spring 2017

More Pipelined Microprocessor
Announcements

• Prelim 2 tonight, 7:30-9:00pm, PHL 101

• Extra instructor office hour today 3:30-4:30pm
Example: Branch Delay Slot

BEQ R2, R3, X
NOP
OR R4, R1, R3
X: AND R6, R1, R2
ADDI R7, R6, 3
SUB R5, R2, R2

• Can you identify one instruction to fill the branch delay slot? Say NONE otherwise.
Review: Load Instructions and Forwarding

LW R1, 0(R2)

OR R4, R1, R3

SUB R5, R2, R1

AND R6, R1, R2

ADDI R7, R7, 3
Pipeline Control Unit (CU)
Pipeline Control Requirements

• Generate control signals for each stage
  – IF: PCJ
  – EX: MB, F
  – MEM: MW, MD
  – WB: LD

• Detect forwarding conditions and generate MUX control signals

• Detect data hazards and insert pipeline bubbles
  – Assumes no load delay slot defined in ISA

• Assume branch delay slot defined in ISA
Generating Control for Each Stage

PCJ  →  PC  →  Inst RAM  →  Decoder  →  RF  →  Adder  →  CU

IF/ID  →  ID/EX  →  ALU  →  EX/MEM  →  MEM/WB

LD  →  SA  →  SB  →  DR  →  D_in  →  RF

MW  →  MD  →  Data RAM  →  D_IN

= ?  →  sign bit  →  MB, F  →  F_m ... F_0

PCJ  →  PCJ  →  +2  →  MUX  →  PCL

Lecture 19:  7
Forwarding Unit (Partial)
R-type to R-type Forwarding

- **ADD** \( R1, R2, R3 \)
- **OR** \( R4, R1, R3 \)
- **SUB** \( R5, R2, R1 \)
- **AND** \( R6, R1, R2 \)

**MEM→EX**: \( \text{MEM.DR} == (\text{EX.SA} || \text{EX.SB}) \)

**WB→EX**: (1) \( \text{WB.DR} == (\text{EX.SA} || \text{EX.SB}) \) and (2) there is no forwarding from MEM→EX in this cycle

**WB→ID**: \( \text{WB.DR} == (\text{ID.SA} || \text{ID.SB}) \)
**R-type to R-type Forwarding**

- **ADD** $R1, R2, R3$
- **OR** $R1, R1, R3$
- **SUB** $R5, R2, R1$
- **AND** $R6, R1, R2$
- **ADDI** $R7, R1, 3$
R-type to R-type Forwarding

- **MEM→EX**
  - MEM.DR == (EX.SA || EX.SB)

- **WB→EX**
  - WB.DR == (EX.SA || EX.SB) and
  - MEM.DR != (EX.SA || EX.SB)

- **WB→ID**
  - WB.DR == (ID.SA || ID.SB)
  - No need to check other forwarding paths. Why?
R-type to Branch Forwarding

- **ADD** \( R_1, R_2, R_3 \)
- **OR** \( R_4, R_6, R_3 \)
- **BGEZ** \( R_1, X \)
- **AND** \( R_6, R_5, R_2 \)
- **ADDI** \( R_7, R_7, 3 \)

*Other forwarding conditions to handle as well*
Pipeline with Fwding + Branch HW in ID
Data Hazards Requiring Bubbles

• Occur when instructions are too close together for forwarding to work

• Requires adding bubbles in the pipeline

• Data hazard conditions to detect and handle
  – Load followed by R-type
  – Load followed by I-type ALU instruction
  – Load followed by Load
  – Load followed by Store (two cases)
  – Load followed by Branch
  – ALU instruction followed by Branch
Load Followed by R-type Instruction

if (EX.Load && ID.R-type) { // Inst in EX is a Load and inst in ID is R-type
    if (EX.DR == (ID.SA || ID.SB)) { // DR of Load matches SA or SB of inst in ID
        Insert NOP into EX in next cycle // Insert bubble
        Don’t Load IF/ID in next cycle // Hold instruction in ID for a cycle
        Don’t Load PC in next cycle // Hold instruction in IF for a cycle
    }
}
Load Followed by R-type Instruction

\[ \text{LW } \text{R1}, 0(\text{R2}) \]

\[ \text{OR } \text{R4, R1, R3} \]

\[ \text{SUB } \text{R5, R2, R1} \]

\[ \text{AND } \text{R6, R1, R2} \]
Load Followed by R-type Instruction

IM
LW R1,0(R2)
OR R4,R1,R3
SUB R5,R2,R1
AND R6,R1,R2

Reg

ALU

DM

Reg

OR R4,R1,R3
LW R1,0(R2)
Load Followed by R-type Instruction

IM
LW R1,0(R2)
OR R4,R1,R3
SUB R5,R2,R1
AND R6,R1,R2

Reg

ALU

DM

Reg

SUB R5,R2,R1
OR R4,R1,R3
LW R1,0(R2)

Detect hazard
Load Followed by R-type Instruction

- IM
  - LW R1,0(R2)
  - OR R4,R1,R3
  - SUB R5,R2,R1
  - AND R6,R1,R2

- Reg

- ALU
- Sub R5,R2,R1
- OR R4,R1,R3
- AND R6,R1,R2
- NOP

- DM

- Reg

- LW R1,0(R2)

Insert bubble

Hold these instructions
Load Followed by R-type Instruction

Pipeline continues normally
R1 value forwarded from WB to EX and ID
Hazard Detection Unit (Partial)

‘Clear’ puts 0 into ID/EX
Pipeline with Datapath and Control

Lecture 19: 22
Load Followed by Branch Instruction

LW $R1, 0(R2)$

BEQ $R4, R1, X$

Two bubbles are needed

- Also need to handle Load followed by branch two instructions apart
R-type to Store Forwarding

SUB $R3, R1, R2

SW $R3, 4(R1)$

ADD $R4, R3, R2$
Pipeline without Fwding for Store SB
Slight Pipeline Improvement for Stores
Next Time

- H&H 8-8.3

Caches