MIPS

Microprocessor without Interlocked Pipeline Stages

Caratteristiche generali

Istruzioni

- Dimensione: **32 bit** (4 byte)
- Operazioni sui dati: da **registro a registro**
- Operazione sulla memoria: da **registro a memoria** (SW – Store Word) e da **memoria a registro** (LW – Load Word)

Registri

- 32 registri di dimensione 32 bit
- Indicazione dei registri: $0, $1, $2, $3, …
- Il registro $0 ha sempre valore 0

Formato istruzioni

Formato R

opcode	rs	rt	rd	shamt	funct

Formato I

opcode	rs	rt	address/const

Formato J

opcode	target address

Caratteristiche della pipeline

- Una istruzione è divisa in 5 fasi (**IF, ID, EX, MEM, WB**)
- Ogni fase ha la durata di un ciclo di clock
- Quando la pipeline è a regime, ad ogni ciclo di clock:
  - Termina un'istruzione
  - Terminano cinque fasi di cinque istruzioni diverse

Fasi di un'istruzione in pipeline

1. IF – Instruction Fetch

- **IR** ← **MEM**[**PC**]
- **NPC** ← **PC** + 4

2. ID – Instruction Decode

Formato istruzione: R

- **A** ← **Regs**[**rs**]
- **B** ← **Regs**[**rt**]

Formato istruzione: I

- **A** ← **Regs**[**rs**]
- **B** ← **Regs**[**rt**]
- **Imm** ← Campo **Imm** di **IR** (***Imm** è di 16 bit => si estende il segno per ottenere **Imm** di 32 bit)

Formato istruzione: J

- **Imm** ← Campo **Imm** di **IR** (*)

3. EX – Execute Instruction

Tipo istruzione: Aritmetico-logica, Registro-Registro

- **ALUOutput** ← **A** *funct* **B**

Tipo istruzione: Aritmetico-logica, Registro-Immediato

- **ALUOutput** ← **A** *opcode* **Imm**

Tipo istruzione: Riferimento a memoria

- **ALUOutput** ← **A** + **Imm**

Tipo istruzione: Salto condizionato (BEQ)

- **Cond** ← (**A** – **B** == **0**)
- **Target** ← **NPC** + (**Imm** << **2**) (**di Imm viene effettuato lo shifting a sinistra di 2 posizioni per moltiplicarne il valore per 4, essendo Imm il numero di parole (e non la dimensione) da sommare a NPC per ottenere l'indirizzo di destinazione)

Tipo istruzione: Salto incondizionato

- **Target** ← **NPC** + (**Imm** << **2**) (**)

4. MEM – Memory Access / Branch Completition

- **PC** ← **NPC**

Tipo istruzione: Riferimento a memoria

- **LMD** ← **MEM**[**ALUOutput**] (per l'istruzione LW)
- **MEM**[**ALUOutput**] ← **B** (per l'istruzione SW)

Tipo istruzione: Salto condizionato (BEQ)

- **if** (**Cond**) **PC** ← **Target**

Tipo istruzione: Salto incondizionato

- **PC** ← **Target**

5. WB – Write Back

Tipo istruzione: Riferimento a memoria

- **Regs**[**rt**] ← **LMD** (per l'istruzione LW)

Tipo istruzione: Aritmetico-logica, Registro-Registro

- **Regs**[**rd**] ← **ALUOutput**

Tipo istruzione: Aritmetico-logica, Registro-Immediato

- **Regs**[**rt**] ← **ALUOutput**

Pipeline Registers / Pipeline Latches

- Memorizzano l'istruzione in lavorazione, i relativi dati e segnali di controllo
- Ad ogni istante (ciclo di clock) questi registri contengono i dati di un'istruzione diversa
- Sono in realtà **banchi di registri**, e sono i seguenti: **IF/ID**, **ID/EX**, **EX/MEM**, **MEM/WB**

In dettaglio: fase IF

- **IF/ID.IR** ← **MEM**[**PC**]
- **IF/ID.NPC** ← **PC+4** oppure **Target** di salto condizionato/incondizionato

In dettaglio: fase ID

- **ID/EX.IR** ← **IF/ID.IR**
- **ID/EX.A** ← **Regs**[**IF/ID.IR**[**rs**]]
- **ID/EX.B** ← **Regs**[**IF/ID.IR**[**rt**]]
- **ID/EX.Imm** ← **IF/IF.IR**[**Imm**] (*)
- **ID/EX.NPC** ← **IF/ID.NPC**

In dettaglio: fase EX

- **EX/MEM.IR** ← **ID/EX.IR**

Tipo istruzione: Aritmetico-logica, Registro-Registro (oppure Registro-Immediato)

- **EX/MEM.ALUOutput** ← **ID/EX.A** *funct* (oppure *opcode*) **ID/EX.B** (oppure **ID/EX.Imm**)

Tipo istruzione: Riferimento a memoria

- **EX/MEM.ALUOutput** ← **ID/EX.A** + **ID/EX.Imm**
- **EX/MEM.B** ← **ID/EX.B** (per l'istruzione SW)

Tipo istruzione: Salto condizionato/incondizionato

- **EX/MEM.Target** ← **ID/EX.NPC** + (**Imm** << **2**)
- **EX/MEM.Cond** ← **Zero** (**ID/EX.A** – **ID/EX.B** == **0**) (per l'istruzione BEQ)

In dettaglio: fase MEM

- **MEM/WB.IR ← EX/MEM.IR**

Tipo istruzione: Aritmetico-logica, Registro-Registro (oppure Registro-Immediato)

- **MEM/WB.ALUOutput ← EX/MEM.ALUOutput**

Tipo istruzione: Riferimento a memoria

- **MEM/WB.LMD ← MEM**[**EX/MEM.ALUOutput**] (per l'istruzione LW)
- **MEM**[**EX/MEM.ALUOutput**] ← **EX/MEM.B** (per l'istruzione SW)

In dettaglio: fase WB

Tipo istruzione: Aritmetico-logica, Registro-Registro (oppure Registro-Immediato)

- **Regs**[**MEM/WB.IR**[**rd**]] ← **MEM/WB.ALUOutput**

Tipo istruzione: Riferimento a memoria

- **Regs**[**MEM/WB.IR**[**rt**]] ← **MEM/WB.LMD** (per l'istruzione LW)

Segnali di controllo

- **ALUSrc**: origine del secondo operando (B oppure Imm)
- **ALUop**: operazione che deve svolgere la ALU
- **MemWrite**: attivo se l'istruzione è una SW
- **MemRead**: attivo se l'istruzione è una LW
- **PCSrc**: PC + 4 oppure Target, in base a Zero
- **RegWrite**: attivo se va aggiornato il valore di un registro
- **MemtoReg**: attivo se il dato da scrivere proviene dalla memoria (altrimenti proviene dalla ALU)
- **RegDst**: "nome" del registro in cui scrivere il dato in output dall'istruzione
- **Zero**
- **Branch**

Fonti

- PDF disponibili su Moodle del corso di Architettura degli Elaboratori 2017/2018 delle lezioni del:
  - 11/12/2017 (file: *lezione_11_12_17.pdf*)
  - 13/12/2017 (file: *lezione_13_12_17.pdf*)
  - 14/12/2017 (file: *lezione_14_12_17.pdf*)