**National Institute of Technology Karnataka Surathkal**
**Department of Information Technology**

# IT 301 Parallel Computing

## Scalar Processors – Control Instructions

**Dr. Geetha V**

*Assistant Professor*

*Dept of Information Technology*

*NITK Surathkal*

# Index

- 1. Five Stage pipeline design
- 2: Control Hazards

# 1: Introduction

**Course Plan: Theory:**

**Part A: Parallel Computer Architectures**

Week 1,2,3: ***Introduction to Parallel Computer Architecture:***

Parallel Computing,

Parallel architecture,

bit level, instruction level , data level and task level parallelism.

**Instruction level parallelisms: pipelining (Data and control instructions),**

**scalar processors** and superscalar processors,

vector processors.

Parallel computers and computation.

# 1. Five Stage Pipeline Design

- Fetch Stage: The address in program counter is sent to Instruction register and instruction is fetched from memory. PC is incremented.
- Decode Stage: The operands are fetched from register file; Sign extension done if necessary.
- Execution Stage: The instruction is executed.
- Memory Stage: The Load and Store instructions execution.
- Write Back Stage: The results are written back to register file.

| Fetch (F) | Decode (D) | Execute (E) | Memory Access (M) | Write Back (W) |
|---|---|---|---|---|

# 1. Five Stage Pipeline Design

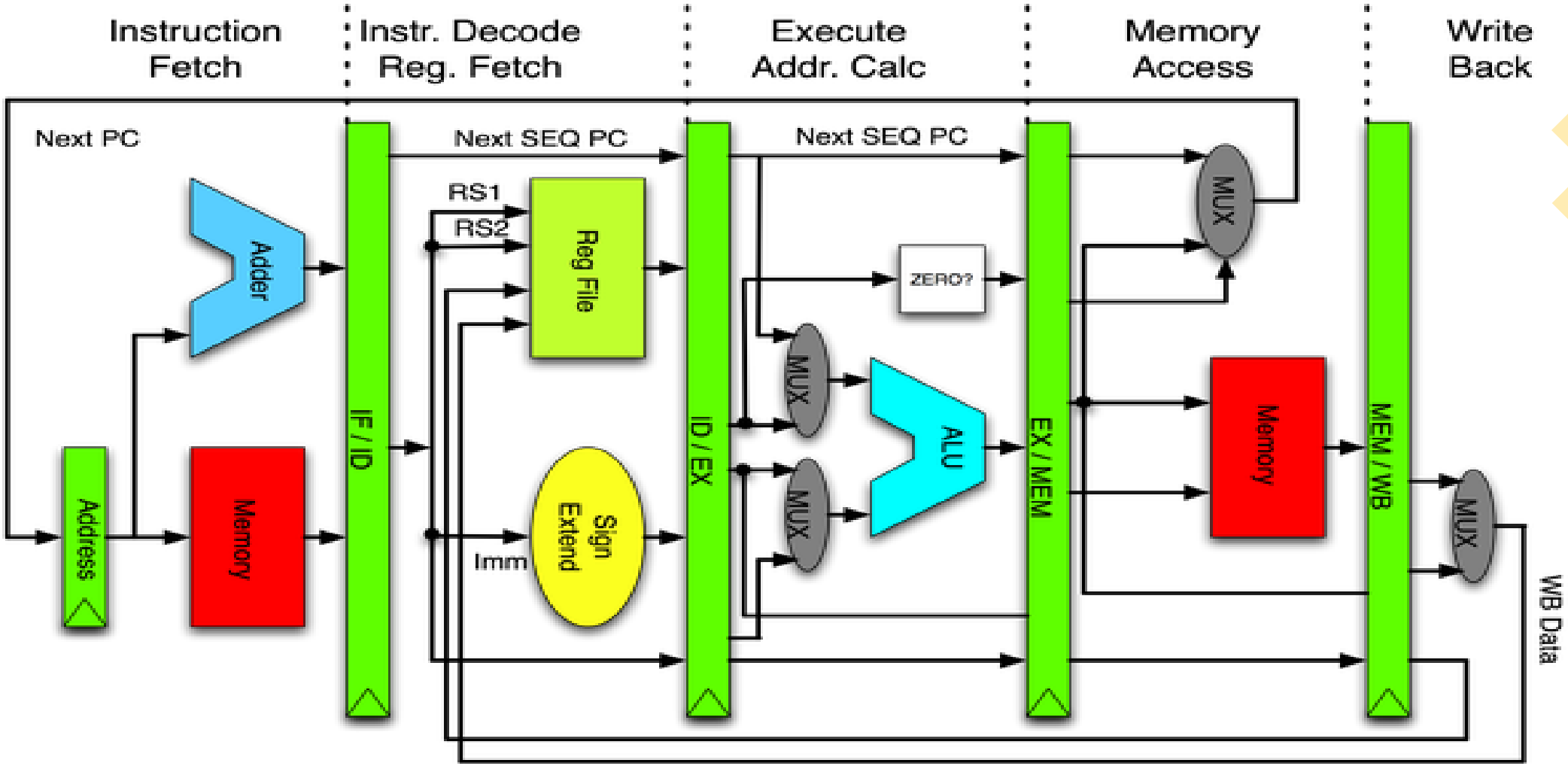


Image from Wikimedia

# 1: Five stage pipeline design

- **Data Dependency (Data Hazard)**

Dependency of instructions due to operand values unavailability

Solution :

(a) Data forwarding

(b) Stall the pipeline untill the data is available.

- **Control Dependency (Control Hazard)**

Dependency due to unresolved decision on control instructions.

# 2. Control hazards

- The branches – conditional branches – cause pipeline hazard
- The outcome of a conditional branch is not known until the end of the EX stage, but is required at IF to load another instruction and keep the pipeline full.

```
        JGE Next
        Add CX, 02
        Dec BL
        JMP NEXT
NEXT:
```

# 2. Control Hazards

- **Data Dependency (Data Hazard)**

Dependency of instructions due to operand values unavailability

Solution :

(a) Data forwarding

(b) Stall the pipeline untill the data is available.

- **Control Dependency (Control Hazard)**

Dependency due to unresolved decision on control instructions.

(a) Insert NOP (No Operation)

(b) NOP after execution of instruction

(c) Branch Prediction

# 2. Control hazards

| Memory | Instruction | Operation |
|---|---|---|
| 400 | MOV SI, 500 | SI <- 500 |
| 403 | MOV DI, 600 | DI <- 600 |
| 406 | MOV AX, 0000 | AX = 0000 |
| 409 | MOV CL, [SI] | CL <- [SI] |
| 40B | MOV BL, CL | BL <- CL |
| 40D | INC SI | SI = SI + 1 |
| 40E | NXT: ADD AL, [SI] | AL = AL + [SI] |
| 410 | ADC AH, 00 | AH = AH + 00 + cy |
| 412 | INC SI | SI = SI + 1 |
| 413 | DEC CL | CL = CL – 1 |
| 415 | JNZ NXT (40E) | JUMP if ZF = 0 |
| 417 | DIV BL | AX = AX / BL |
| 419 | MOV [DI], AX | [DI] <- AX |
| 41B | HLT | Stop |

- Program to find average of array elements

(a) Inserting NOP

1. Insert NOP (No operation) when Branch instruction

| F | D | E | M | W |
|---|---|---|---|---|
| NOP | NOP | JNZ 40E | DEC CL | INC SI |

| F | D | E | M | W |
|---|---|---|---|---|
| ADD AL, [SI] | NOP | NOP | JNZ 40E | DEC CL |

# 2. Control hazards

| Memory | Instruction | Operation |
|---|---|---|
| 400 | MOV SI, 500 | SI <- 500 |
| 403 | MOV DI, 600 | DI <- 600 |
| 406 | MOV AX, 0000 | AX = 0000 |
| 409 | MOV CL, [SI] | CL <- [SI] |
| 40B | MOV BL, CL | BL <- CL |
| 40D | INC SI | SI = SI + 1 |
| 40E | NXT: ADD AL, [SI] | AL = AL + [SI] |
| 410 | ADC AH, 00 | AH = AH + 00 + cy |
| 412 | INC SI | SI = SI + 1 |
| 413 | DEC CL | CL = CL – 1 |
| 415 | JNZ NXT (40E) | JUMP if ZF = 0 |
| 417 | DIV BL | AX = AX / BL |
| 419 | MOV [DI], AX | [DI] <- AX |
| 41B | HLT | Stop |

- Program to find average of array elements

(b) NOP after execution of instruction

2.Insert NOP (No operation) when Branch is taken (after execution of branch)

| F | D | E | M | W |
|---|---|---|---|---|
| MOV [DI], AX | DIV BL | JNZ 40E | DEC CL | INC SI |

| F | D | E | M | W |
|---|---|---|---|---|
| ~~MOV [DI], AX~~ NOP | ~~DIV BL~~ NOP | JNZ 40E | DEC CL | INC SI |

| F | D | E | M | W |
|---|---|---|---|---|
| ADD AL, [SI] | ~~MOV [DI], AX~~ NOP | ~~DIV BL~~ NOP | JNZ 40E | DEC CL |

# 2. Control hazards

| Memory | Instruction | Operation |
|---|---|---|
| 400 | MOV SI, 500 | SI <- 500 |
| 403 | MOV DI, 600 | DI <- 600 |
| 406 | MOV AX, 0000 | AX = 0000 |
| 409 | MOV CL, [SI] | CL <- [SI] |
| 40B | MOV BL, CL | BL <- CL |
| 40D | INC SI | SI = SI + 1 |
| 40E | NXT: ADD AL, [SI] | AL = AL + [SI] |
| 410 | ADC AH, 00 | AH = AH + 00 + cy |
| 412 | INC SI | SI = SI + 1 |
| 413 | DEC CL | CL = CL – 1 |
| 415 | JNZ NXT (40E) | JUMP if ZF = 0 |
| 417 | DIV BL | AX = AX / BL |
| 419 | MOV [DI], AX | [DI] <- AX |
| 41B | HLT | Stop |

- Program to find average of array elements

(c) Branch Prediction : Branch Taken

3. Branch Prediction: Branch Taken

| F | D | E | M | W |
|---|---|---|---|---|
| ADC AH, 00 | ADD AL, [SI] | JNZ 40E | DEC CL | INC SI |

Branch not taken

| F | D | E | M | W |
|---|---|---|---|---|
| ~~ADC AH, 00~~ NOP | ~~ADD AL, [SI]~~ NOP | JNZ 40E | DEC CL | INC SI |

| F | D | E | M | W |
|---|---|---|---|---|
| DIV BL | ~~ADC AH, 00~~ NOP | ~~ADD AL, [SI]~~ NOP | JNZ 40E | DEC CL |

Branch Taken

| F | D | E | M | W |
|---|---|---|---|---|
| INC SI | ADC AH, 00 | ADD AL, [SI] | JNZ 40E | DEC CL |

# Thank You

# Reference

**Textbooks and/or Reference Books:**


1. Professional CUDA C Programming – John Cheng, Max Grossman, Ty McKercher, 2014
2. Wilkinson, M. Allen, ”Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers”, Pearson Education, 1999
3. I. Foster, ”Designing and building parallel programs”, 2003
4. Parallel Programming in C using OpenMP and MPI – Micheal J Quinn, 2004
5. Introduction to Parallel Programming – Peter S Pacheco, Morgan Kaufmann Publishers, 2011
6. Advanced Computer Architectures: A design approach, Dezso Sima, Terence Fountain, Peter Kacsuk, 2002
7. Parallel Computer Architecture : A hardware/Software Approach, David E Culler, Jaswinder Pal Singh Anoop Gupta, 2011 8. Introduction to Parallel Computing, Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, Pearson, 2011