# Computer Architecture II project phase 1

|  |  |  |
| --- | --- | --- |
| Name | BN | Sec |
| محمد حسانين محمد | 16 | 2 |
| بلال الحسينى احمد عبدالفتاح | 19 | 1 |
| احمد عطيه عبدالراضى عبدالعال | 4 | 1 |
| محمد سيد حسين عبدالرازق | 19 | 2 |

# Pipeline hazards and solutions

## Structural Hazard

Stall the pipeline:

If the memory stage uses the address bus, the fetch stage will issue control signals of “NOP” instruction without incrementing the PC.

## Data Hazard

### EX hazard (EX to EX forwarding)

If(EX/MEM.RegWrite AND (EX/MEM.Rd == ID/EX.(Rs/Rd)))

Forward EX/MEM.Rd data

### MEM hazard (MEM to EX forwarding)

#### Case1 (Forward from ith instruction to (i+1)th instruction):

If(MEM/WP.RegWrite AND (MEM/WP.Rd == ID/EX.(Rs/Rd)))

Forward MEM/WP.Rd data to

#### Case2 (Forward from ith instruction to (i+2)th instruction):

If(MEM/WP.RegWrite AND NOT(EX/MEM.RegWrite) AND (EX/MEM.Rd != ID/EX.(Rs/Rd)))

Forward MEM/WP.Rd data

#### Case3 (Load-case)

If(ID/EX.MemRead AND (ID/EX.Rs = IF/ID.(Rs/Rd)))

Stall the pipeline

## Control Hazard

Approach: Not Taken static prediction.

Evaluation in Decode stage.

## Instruction bits details

## Registers

10

4 bits

R0 (0000) to R7 (0111), PC (1000), SP (1001)

## Types

### R-type

|  |  |  |  |
| --- | --- | --- | --- |
| Op-code (8) | Rdst(4) | Rsrc(4) | Don’t care (16) |

All instructions except:

IADD, SHL, SHR, LDM

### I-type

|  |  |  |  |
| --- | --- | --- | --- |
| Op-code (8) | Rdst(4) | Rsrc(4) | IMM(16) |

Instructions:

IADD, SHL, SHR, LDM

PUSH and POP would consist of 4 words and take 2 cycles to complete. It will have the two instructions

PUSH:

IADD SP, -2

STD Rdst, 0(SP)

POP:

LDD Rdst, 0(SP)

IADD SP, 2

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MOV |  |  |  |  |  |  |  | 0 | 0 | 0 | 0 | 0 |
| ADD |  |  |  |  |  |  |  | 0 | 0 | 0 | 0 | 1 |
| SUB |  |  |  |  |  |  |  | 0 | 0 | 0 | 1 | 0 |
| AND |  |  |  |  |  |  |  | 0 | 0 | 0 | 1 | 1 |
| OR |  |  |  |  |  |  |  | 0 | 0 | 1 | 0 | 0 |
| IADD |  |  |  |  |  |  |  | 0 | 0 | 1 | 0 | 1 |
| SHL |  |  |  |  |  |  |  | 0 | 0 | 1 | 1 | 0 |
| SHR |  |  |  |  |  |  |  | 0 | 0 | 1 | 1 | 1 |
| RLC |  |  |  |  |  |  |  | 0 | 1 | 0 | 0 | 0 |
| RRC |  |  |  |  |  |  |  | 0 | 1 | 0 | 0 | 1 |
|  | | | | | | | | | | | | |
| NOP |  |  |  |  |  |  |  | 0 | 1 | 0 | 1 | 0 |
| SETC |  |  |  |  |  |  |  | 0 | 1 | 0 | 1 | 1 |
| CLRC |  |  |  |  |  |  |  | 0 | 1 | 1 | 0 | 0 |
| CLR |  |  |  |  |  |  |  | 0 | 1 | 1 | 0 | 1 |
| NOT |  |  |  |  |  |  |  | 0 | 1 | 1 | 1 | 0 |
| INC |  |  |  |  |  |  |  | 0 | 1 | 1 | 1 | 1 |
| DEC |  |  |  |  |  |  |  | 1 | 0 | 0 | 0 | 0 |
| NEG |  |  |  |  |  |  |  | 1 | 0 | 0 | 0 | 1 |
| OUT |  |  |  |  |  |  |  | 1 | 0 | 0 | 1 | 0 |
| IN |  |  |  |  |  |  |  | 1 | 0 | 0 | 1 | 1 |
|  | | | | | | | | | | | | |
| LDM |  |  |  |  |  |  |  | 1 | 0 | 1 | 0 | 0 |
| LDD |  |  |  |  |  |  |  | 1 | 0 | 1 | 0 | 1 |
| STD |  |  |  |  |  |  |  | 1 | 0 | 1 | 1 | 0 |
|  | | | | | | | | | | | | |
| JZ |  |  |  |  |  |  |  | 1 | 0 | 1 | 1 | 1 |
| JN |  |  |  |  |  |  |  | 1 | 1 | 0 | 0 | 0 |
| JC |  |  |  |  |  |  |  | 1 | 1 | 0 | 0 | 1 |
| JMP |  |  |  |  |  |  |  | 1 | 1 | 0 | 1 | 0 |

## Opcode of each instruction

# Control Unit Detailed Design

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Inst | | MemWrite | MemRead | ALU | IMM | IO | WBO | RegWrite |
| MOV | | 0 | 0 | 0000 | 0 | 0 | 0 | 1 |
| ADD | | 0 | 0 | 0001 | 0 | 0 | 0 | 1 |
| SUB | | 0 | 0 | 0010 | 0 | 0 | 0 | 1 |
| AND | | 0 | 0 | 0011 | 0 | 0 | 0 | 1 |
| OR | | 0 | 0 | 0100 | 0 | 0 | 0 | 1 |
| IADD | | 0 | 0 | 0001 | 1 | 0 | 0 | 1 |
| SHL | | 0 | 0 | 0101 | 1 | 0 | 0 | 1 |
| SHR | | 0 | 0 | 0110 | 1 | 0 | 0 | 1 |
| RLC | | 0 | 0 | 0111 | 0 | 0 | 0 | 1 |
| RRC | | 0 | 0 | 1000 | 0 | 0 | 0 | 1 |
|  |  | | | | | | | |
| NOP | | 0 | 0 | 0000 | 0 | 0 | 0 | 0 |
| SETC | | 0 | 0 | 1001 | 0 | 0 | 0 | 0 |
| CLRC | | 0 | 0 | 1010 | 0 | 0 | 0 | 0 |
| CLR | | 0 | 0 | 1111 | 0 | 0 | 0 | 1 |
| NOT | | 0 | 0 | 1011 | 0 | 0 | 0 | 1 |
| INC | | 0 | 0 | 1100 | 0 | 0 | 0 | 1 |
| DEC | | 0 | 0 | 1101 | 0 | 0 | 0 | 1 |
| NEG | | 0 | 0 | 1110 | 0 | 0 | 0 | 1 |
| OUT | | 0 | 0 | 0000 | 0 | 1 | 0 | 0 |
| IN | | 0 | 0 | 0000 | 0 | 1 | 1 | 1 |
|  |  | | | | | | | |
| LDM | | 0 | 0 | 0001 | 1 | 0 | 0 | 1 |
| LDD | | 0 | 1 | 0001 | 0 | 0 | 1 | 1 |
| STD | | 1 | 0 | 0001 | 0 | 0 | 0 | 0 |

## ALU Operations

|  |  |
| --- | --- |
| ALU\_Operations | ALU |
| Op1 | 0000 |
| ADD | 0001 |
| SUB | 0010 |
| AND | 0011 |
| OR | 0100 |
| SHL | 0101 |
| SHR | 0110 |
| RLC | 0111 |
| RRC | 1000 |
| SETC | 1001 |
| CLRC | 1010 |
| NOT | 1011 |
| INC | 1100 |
| DEC | 1101 |
| NEG | 1110 |
| CLR | 1111 |

# Pipeline registers design

## IF/ID

* Instruction

## ID/EX

* Control Signals:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| MemWrite | MemRead | ALU | IMM | IO | WBO | RegWrite |

* IMM Value (32 bit)
* Src register (32 bits)
* Dst register (32 bits)
* Rdst (4 bits)

## EX/MEM

* Control signals

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MemWrite | MemRead | IO | WBO | RegWrite |

* ALU\_OUTPUT (32 bits)
* Dst register (32 bits)
* Rdst (4 bits)

## MEM/WB

* Control signals

|  |  |
| --- | --- |
| WBO | RegWrite |

* MEM/IO\_OUTPUT (32 bits)
* ALU\_OUTPUT (32 bits)
* Rdst (4 bits)