MIPS PIPELINE

- Raport -

Nume student: Matiș Oana-Antonia

Grupa: 30228

* + - 1. Configurare registre MIPS16 Pipeline…………………………………………….3
      2. Identificarea hazardurilor…………………………………………………………...3
      3. Schema MIPS Pipeline….……………………....................................................6
      4. Concluzii………………………………………......................................................7

1. Configurare registre MIPS16 Pipeline

Se introduc pe coloane semnalele de date și control mapate la registre, de sus în jos, începând de la biții cei mai semnificativi ai registrului către cei mai puțin semnificativi. Se introduc în paranteză biții din registru alocați pentru fiecare semnal în parte. În dreptul numelui registrelor din primul rând se introduce în paranteză poziția bitului cel mai semnificativ () din care reiese dimensiunea totală alocată registrului.

|  |  |  |  |
| --- | --- | --- | --- |
| **REG\_IF\_ID(31 – 0)** | REG\_ID\_EX(82 – 0) | REG\_EX\_MEM(55 – 0) | REG\_MEM\_WB(36 – 0) |
| Instruction(31 – 16) | rd(82 – 80) | rWA(55 – 53) | rWA(36 – 34) |
| PC + 1(15 – 0) | rt(79 – 77) | RD2(52 – 37) | ALUResOut(33 – 18) |
|  | sa(76) | ALURes(36 – 21) | MemData(17 – 2) |
|  | func(75 – 73) | Zero(20) | RegWrite(1) |
|  | Ext\_Imm(72 – 57) | branchAddr(19 – 4) | MemToReg(0) |
|  | RD2(56 – 41) | Branch(3) |  |
|  | RD1(40 – 25) | MemWrite(2) |  |
|  | PC + 1(24 – 9) | RegWrite(1) |  |
|  | RegDst(8) | MemToReg(0) |  |
|  | ALUSrc(7) |  |  |
|  | ALUOp(6 – 4) |  |  |
|  | Branch(3) |  |  |
|  | MemWrite(2) |  |  |
|  | RegWrite(1) |  |  |
|  | MemToReg(0) |  |  |

1. Identificarea hazardurilor
2. Programul inițial:

add $1, $0, $0

addi $4, $0, 10

addi $6, $0, 3

add $2, $0, $0

add $5, $0, $0

beq $1, $4, 13

lw $3, 10($2)

bne $3, $6, 4

add $5, $5, $3

addi $2, $2, 1

addi $1, $1, 1

j 5

sw $5, 20($0)

* Hazard de control:
  + apare la salturi (branch si jump)
  + rezolvare: dupa fiecare instructiune de branch am adaugat 3 NOOP-uri si dupa fiecare instructiune de jump am adaugat 1 NOOP

1. Programul modificat:

add $1, $0, $0

addi $4, $0, 10

addi $6, $0, 3

add $2, $0, $0

add $5, $0, $0

beq $1, $4, 13

NoOP

NoOP

NoOp

lw $3, 10($2)

bne $3, $6, 4

NoOP

NoOP

NoOp

add $5, $5, $3

addi $2, $2, 1

addi $1, $1, 1

j 5

NoOp

sw $5, 20($0)

b"000\_000\_000\_001\_0\_000 " 0010

b"001\_000\_100\_0001010" 220A

b"001\_000\_010\_0000011" 2103

b"000\_000\_000\_010\_0\_000" 0020

b"000\_000\_000\_101\_0\_000" 0050

b"100\_001\_100\_0001011" 860D

b"000\_000\_000\_000\_0\_000",

b"000\_000\_000\_000\_0\_000",

b"000\_000\_000\_000\_0\_000",

b"010\_010\_011\_0001010", 498A

b"110\_011\_110\_0000100", CF04

b"000\_000\_000\_000\_0\_000",

b"000\_000\_000\_000\_0\_000",

b"000\_000\_000\_000\_0\_000",

b"000\_101\_011\_101\_0\_000", 15D0

b"001\_010\_010\_0000001", 2901

b"001\_001\_001\_0000001" 2481

b"111\_0000000000101" E005

b"000\_000\_000\_000\_0\_000",

b"011\_000\_101\_0010100" 6294

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instr/Clk | CC1 | CC2 | CC3 | CC4 | CC5 | CC6 | CC7 | CC8 | CC9 |
| add $1, $0, $0 | IF | ID | EX | MEM | WB |  |  |  |  |
| addi $4, $0, 10 |  | IF | ID | EX | MEM | WB |  |  |  |
| addi $6, $0, 3 |  |  | IF | ID | EX | MEM | WB |  |  |
| add $2, $0, $0 |  |  |  | IF | ID | EX | MEM | WB |  |
| add $5, $0, $0 |  |  |  |  | IF | ID | EX | MEM | WB |

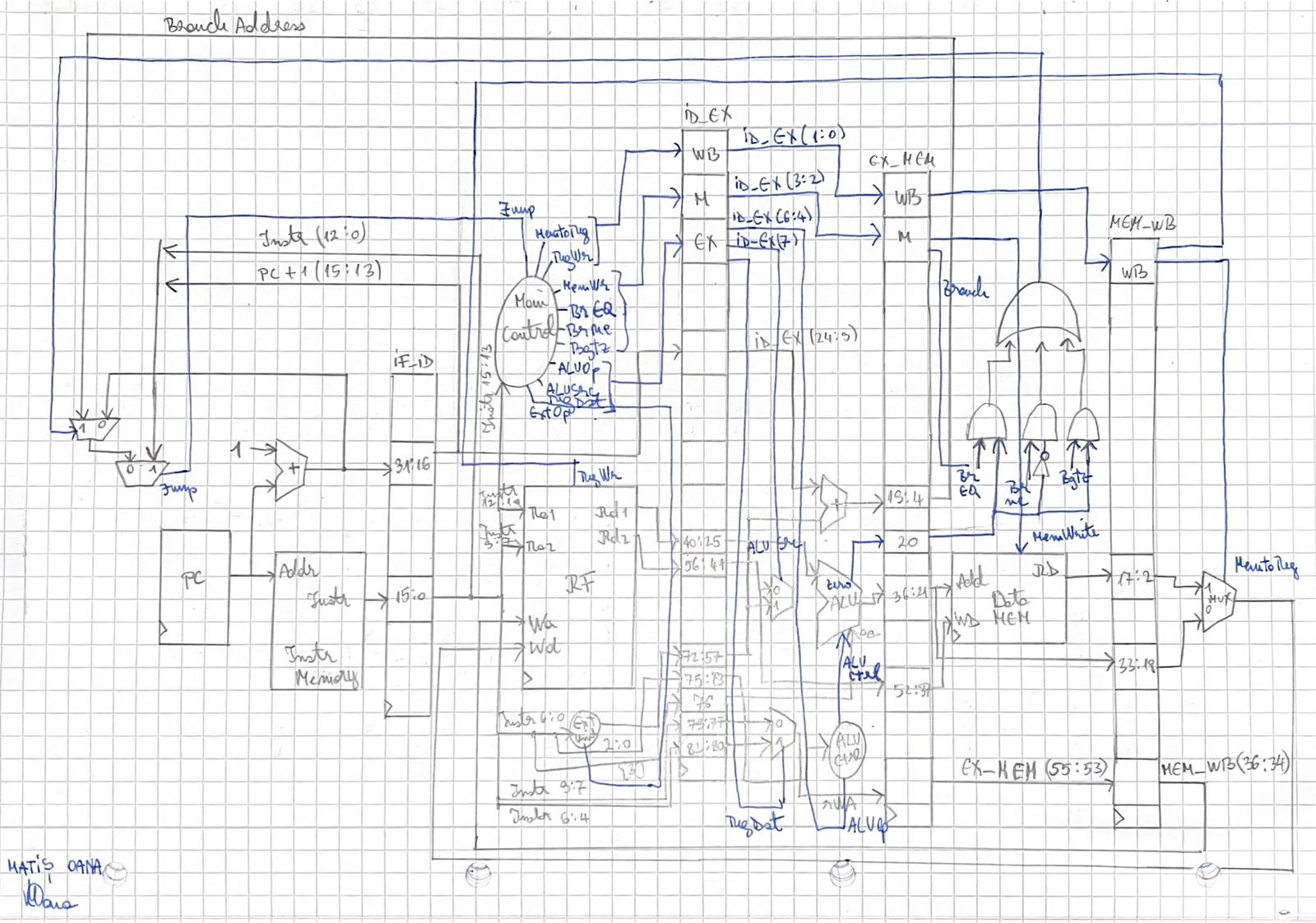
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instr/Clk | CC4 | CC5 | CC6 | CC7 | CC8 | CC9 | CC10 | CC11 | CC12 |
| add $2, $0, $0 | IF | ID | EX | MEM | WB( |  |  |  |  |
| add $5, $0, $0 |  | IF | ID | EX | MEM | WB |  |  |  |
| beq $1, $4, 13 |  |  | IF | ID | EX | MEM | WB |  |  |
| NoOP |  |  |  | IF | ID | EX | MEM | WB |  |
| NoOP |  |  |  |  | IF | ID | EX | MEM | WB |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instr/Clk | CC7 | CC8 | CC9 | CC10 | CC11 | CC12 | CC13 | CC14 | CC15 |
| NoOP | IF | ID | EX | MEM | WB |  |  |  |  |
| NoOP |  | IF | ID | EX | MEM | WB |  |  |  |
| NoOp |  |  | IF | ID | EX | MEM | WB |  |  |
| lw $3, 10($2) |  |  |  | IF | ID | EX | MEM | WB |  |
| bne $3, $6, 4 |  |  |  |  | IF | ID | EX | MEM | WB |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instr/Clk | CC10 | CC11 | CC12 | CC13 | CC14 | CC15 | CC16 | CC17 | CC18 |
| lw $3, 10($2) | IF | ID | EX | MEM | WB |  |  |  |  |
| bne $3, $6, 4 |  | IF | ID | EX | MEM | WB |  |  |  |
| NoOP |  |  | IF | ID | EX | MEM | WB |  |  |
| NoOP |  |  |  | IF | ID | EX | MEM | WB |  |
| NoOp |  |  |  |  | IF | ID | EX | MEM | WB |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instr/Clk | CC13 | CC14 | CC15 | CC16 | CC17 | CC18 | CC19 | CC20 | CC21 |
| NoOP | IF | ID | EX | MEM | WB |  |  |  |  |
| NoOp |  | IF | ID | EX | MEM | WB |  |  |  |
| add $5, $5, $3 |  |  | IF | ID | EX | MEM | WB |  |  |
| addi $2, $2, 1 |  |  |  | IF | ID | EX | MEM | WB |  |
| addi $1, $1, 1 |  |  |  |  | IF | ID | EX | MEM | WB |
|  |  |  |  |  |  |  |  |  |  |
| Instr/Clk | CC16 | CC17 | CC18 | CC19 | CC20 | CC21 | CC22 | CC23 | CC24 |
| addi $2, $2, 1 | IF | ID | EX | MEM | WB |  |  |  |  |
| addi $1, $1, 1 |  | IF | ID | EX | MEM | WB |  |  |  |
| j 5 |  |  | IF | ID | EX | MEM | WB |  |  |
| NoOp |  |  |  | IF | ID | EX | MEM | WB |  |
| sw $5, 20($0) |  |  |  |  | IF | ID | EX | MEM | WB |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

1. Schema MIPS PIPELINE



1. Concluzii

Concluziile acestui raport sunt:

* toate activitatile din laboratoarele 9 – 10 sunt complete
* nu exista erori cu privire la codul scris in VHDL
* procesorul a fost testat pe placa, dar nu este functional