MIPS (MICROPROCESOR) PIPELINE

Studentă: Alina Aurică

Profesor îndrumător: Vlad Miclea

Materie:Arhitectura Calculatoarelor

CUPRINS:

1. INSTRUCȚIUNI
2. TABEL CU REGISTRI
3. TABEL PIPELINE
4. RTL Schematic
5. FUNCȚIONALITATE
6. INSTRUCȚIUNI:

Instrucțiunile pe care eu mi le-am ales ca instrucțiuni suplimentare nu au modificat în niciun fel structura MIPS-ului pipeline așa cum o știm noi.

Cele 4 instrucțiuni sunt:

* De tip R: XOR și NOR
* De tip I: ORI și XORI

1. TABEL CU REGISTRI:

|  |  |  |  |
| --- | --- | --- | --- |
| REG\_IF\_ID(31-0) | REG\_ID\_EX(82-0) | REG\_EX\_MEM(55-0) | REG\_MEM\_WB(36-0) |
| Instr(15-0) | WB(1-0) | WB(1-0) | WB(1-0) |
| PC(PC1)(31-16) | M(3-2) | M(3-2) | rd(17-2) |
|  | EX(8-4) | BAddr(19-4) | ALURez(33-18) |
|  | PC(PC1)(24-9) | Zero(20) | MuxDst(36-34) |
|  | rd1(40-25) | ALURez(36-21) |  |
|  | rd2(56-41) | rd2(52-37) |  |
|  | ExtImm(72-57) | MuxDst(55-53) |  |
|  | func\_sa(76-73) |  |  |
|  | rt(79-77) |  |  |
|  | rd(82-80) |  |  |

1. TABEL PIPELINE:

Programul pe care îl face MIPS-ul:

Add $5 $1 $2

Add $0 $0 $0

Add $0 $0 $0

Sll $3 $5 $0

Add $0 $0 $0

Add $0 $0 $0

Sub $4 $3 $6

Add $0 $0 $0

Add $0 $0 $0

Srl $7 $4 $0

Add $0 $0 $0

Add $0 $0 $0

Addi $1 $7 7

Add $0 $0 $0

Add $0 $0 $0

Sw $5 (0)$1

Add $0 $0 $0

Add $0 $0 $0

Lw $2 (0)$5

Add $0 $0 $0

Add $0 $0 $0

Beq $0 $1 6

Add $0 $0 $0

Add $0 $0 $0

Add $0 $0 $0

And $6 $1 $4

Add $0 $0 $0

Add $0 $0 $0

Or $3 $6 $4

Jump 33

Add $0 $0 $0

Ori $3 $1 7

Add $0 $0 $0

Add $0 $0 $0

Xori $4 $3 2

Xor $4 $3 $2

Add $0 $0 $0

Add $0 $0 $0

Nor $7 $4 $1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instr | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | C10 | C11 | C12 | C13 | C14 | C15 | C16 | C17 | C18 | C19 |
| ADD | IF | ID | EX | MEM | WE($5) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| SLL |  | IF | ID($5) | EX | MEM | WE($3) |  |  |  |  |  |  |  |  |  |  |  |  |  |
| SUB |  |  | IF | ID($3) | EX | MEM | WE($4) |  |  |  |  |  |  |  |  |  |  |  |  |
| SRL |  |  |  | IF | ID($4) | EX | MEM | WE($7) |  |  |  |  |  |  |  |  |  |  |  |
| ADDI |  |  |  |  | IF | ID($7) | EX | MEM | WE($1) |  |  |  |  |  |  |  |  |  |  |
| SW |  |  |  |  |  | IF | ID($1) | EX | MEM | WE($2) |  |  |  |  |  |  |  |  |  |
| LW |  |  |  |  |  |  | IF | ID($2) | EX | MEM | WE |  |  |  |  |  |  |  |  |
| BEQ |  |  |  |  |  |  |  | IF | ID($1) | EX | MEM | WE |  |  |  |  |  |  |  |
| AND |  |  |  |  |  |  |  |  | IF | ID($1) | EX | MEM | WE($6) |  |  |  |  |  |  |
| OR |  |  |  |  |  |  |  |  |  | IF | ID($6) | EX | MEM | WE($3) |  |  |  |  |  |
| JUMP |  |  |  |  |  |  |  |  |  |  | IF | ID | EX | MEM | WE |  |  |  |  |
| ORI |  |  |  |  |  |  |  |  |  |  |  | IF | ID($1) | EX | MEM | WE($3) |  |  |  |
| XORI |  |  |  |  |  |  |  |  |  |  |  |  | IF | ID($3) | EX | MEM | WE($4) |  |  |
| XOR |  |  |  |  |  |  |  |  |  |  |  |  |  | IF | ID($3) | EX | MEM | WE($4) |  |
| NOR |  |  |  |  |  |  |  |  |  |  |  |  |  |  | IF | ID($4) | EX | MEM | WE($7) |

Un al 2-lea tabel cu valori se gasește în excel-ul pipeline atașat.

Având în vedere că dorim să implementăm un MIPS pipeline, trebuie să reparăm hazardele care pot apărea ca urmare a împărțirii pe registri. Cele 3 tipuri de hazarde sunt:

* Hazardul structural – apare atunci când 2 instrucțiuni accesează același registru simultan. Vom scăpa de acest hazard prin scrierea pe front descendent în registrul de memorie.

Spre exemplu hazardul dintre ADDI și BQE (registrul $1).

* Hazardul de date – apare atunci când o instrucțiune are nevoie de datele pe care instrucțiunea precedentă nu a apucat să o scrie în registru

Avem astfel de hazarde între instrucțiunile:

Add și Sll (registrul $5)

Sll și Sub (registrul $3)

Sub și Srl (registrul $4)

Srl și Addi (registrul $7)

Addi și Sw (registrul $1)

Sw și Lw (registrul $2)

And și Or (registrul $6)

Ori și Xori (registrul $3)

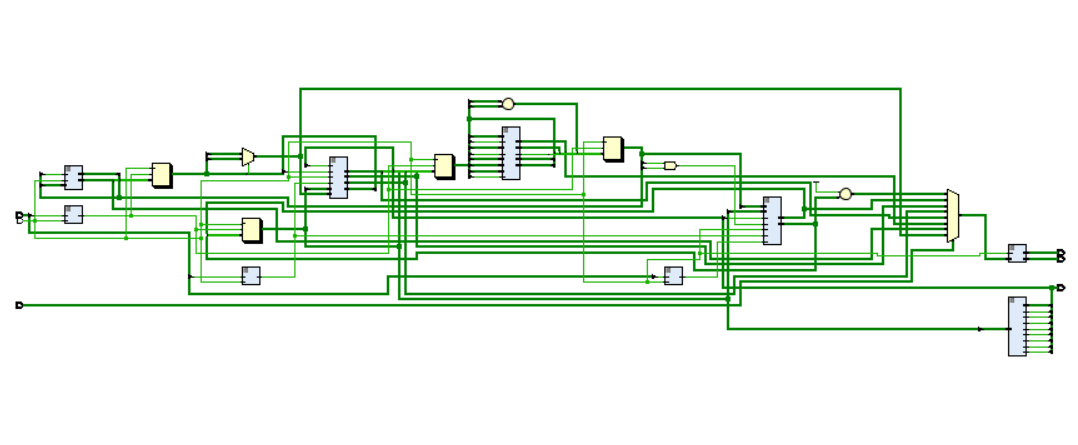
Xor și Nor (registrul $4)

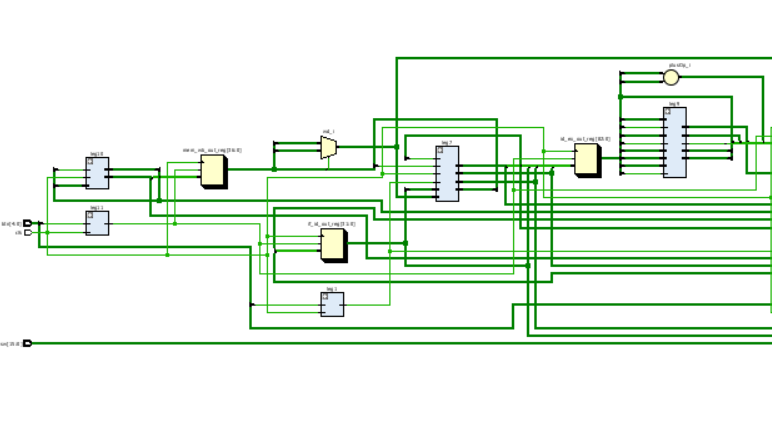
* Hazard de control – apare pe instrucțiunile de salt

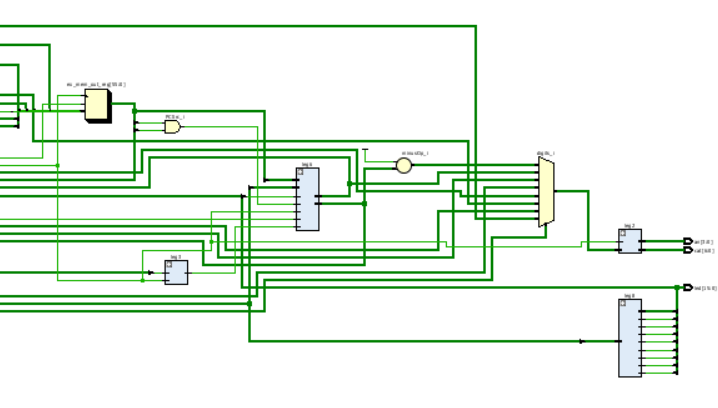
La instrucțiunea Jump adăugăm o instrucțiune NoOp, urmată de modificarea adresei de salt.

La instrucțiunea BQE adăugăm 2 instrucțiuni NoOp, urmată de modificarea adresei de salt.

1. RTL Schematic:







1. FUNCȚIONALITATE:

Mips-ul pipeline a fost testat pe plăcuță, ajunge pana la primele 3 instructiuni.