**MIPS Pipeline**

Giesswein Alexia

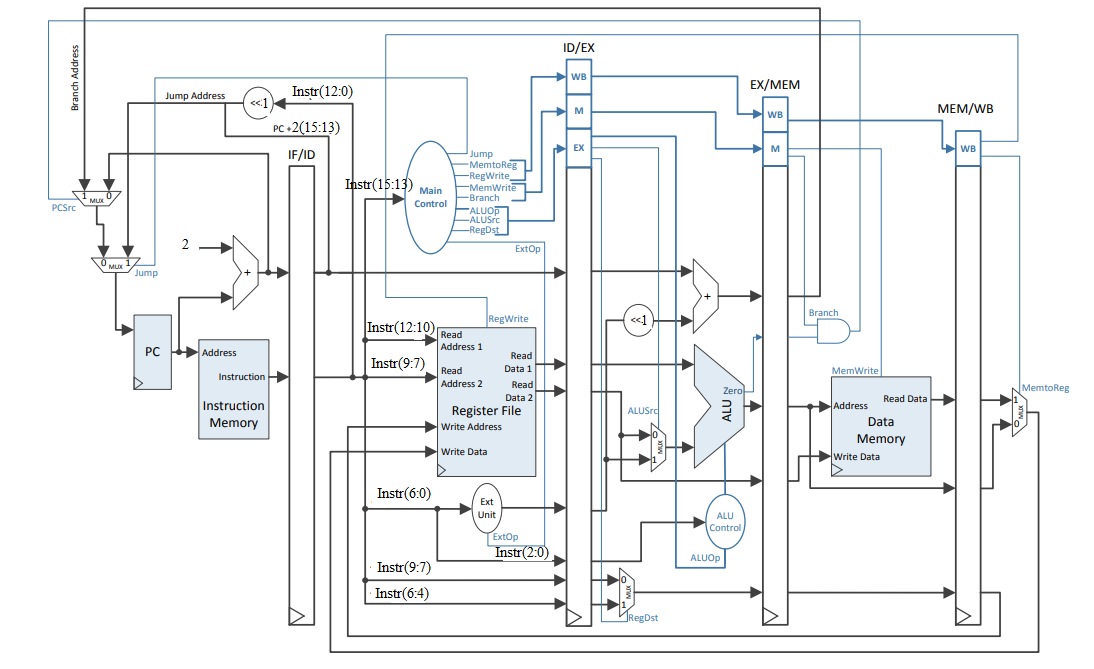
Grupa 30225

Prof. Coord. Mircea Paul Muresan

**Tabelul cu descrierea registrilor de pipeline**

|  |  |  |  |
| --- | --- | --- | --- |
| **IF/ID** | **ID/EX** | **EX/MEM** | **MEM/WB** |
| IF\_ID\_pc(16) | ID\_EX\_memtoreg(1) | EX\_MEM\_memtoreg(1) | MEM\_WB\_memtoreg  (1) |
| IF\_ID\_instruction(16) | ID\_EX\_regwrite(1) | EX\_MEM\_regwrite(1) | MEM\_WB\_regwrite(1) |
|  | ID\_EX\_memwrite(1) | EX\_MEM\_memwrite(1) | MEM\_WB\_memdata  (16) |
|  | ID\_EX\_branch(1) | EX\_MEM\_branch(1) | MEM\_WB\_alures(16) |
|  | ID\_EX\_regdst(1) | EX\_MEM\_aluzero(1) | MEM\_WB\_writeaddress(3) |
|  | ID\_EX\_alusrc(1) | EX\_MEM\_branchaddress  (16) |  |
|  | ID\_EX\_aluop(2) | EX\_MEM\_alures(16) |  |
|  | ID\_EX\_pc(16) | EX\_MEM\_rd2(16) |  |
|  | ID\_EX\_rd1(16) | EX\_MEM\_writeaddress  (3) |  |
|  | ID\_EX\_rd2(16) |  |  |
|  | ID\_EX\_extimm(16) |  |  |
|  | ID\_EX\_rt(3) |  |  |
|  | ID\_EX\_rd(3) |  |  |
|  | ID\_EX\_rs(3) |  |  |
|  | ID\_EX\_func(3) |  |  |

**Schema procesorului mips pipeline**

****

**Hazardurile**

Programul original în assembly mips 16:

add $1, $0, $0 ; i=0 contorul

addi $2, $0, 10 ; n=10 (nr de iteratii)

add $3, $0, $0 ; initializam indexul locatiei de memorie

add $4, $0, $0 ; suma=0

addi $6, $0, 1 ; in registrul $6 avem salvat 1

beq $1, $2, 7 ; i<10

lw $5, 11($3) ; in $5 avem elementul curent

and $7, $6, $5 ; verificam daca numarul curent e par sau impar

beq $6, $7, 1 ; daca numarul e impar sarim

add $4, $4, $5 ; daca numarul e par il adaugam la suma

addi $3, $3, 2 ; indexul urmatorului element din sir

addi $1, $1, 2 ; i=i+2

j 5 ; sarim inapoi la bucla

sw $4, suma($0) ; salvam suma in memorie la adresa suma

Hazardurile:

* Dupa instructiunea beq $1, $2, 7 este hazard de control.
* Intre instructiunea lw $5, 11($3) si and $7, $6, $5 este hazard de date dupa $5.
* Intre instructiunea and $7, $6, $5 si beq $6, $7, 1 este hazard de date dupa $7.
* Dupa instructiunea beq $6, $7, 1 este hazard de control.
* Dupa instructiunea j 5 este hazard de control.

Rescrierea programului fara hazarde:

add $1, $0, $0

addi $2, $0, 10

add $3, $0, $0

add $4, $0, $0

addi $6, $0, 1

beq $1, $2, 20

NoOp

NoOp

NoOp

lw $5, 11($3)

NoOp

NoOp

NoOp

and $7, $6, $5

NoOp

NoOp

NoOp

beq $6, $7, 4

NoOp

NoOp

NoOp

$4, $4, $5

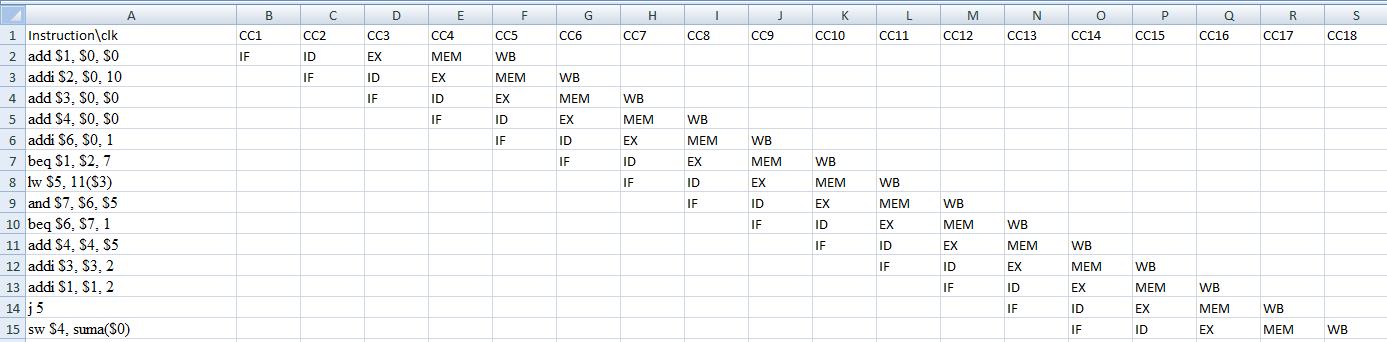
addi $3, $3, 2

addi $1, $1, 2

j 5

NoOp

sw $4, 51($0)

****

Am implementat toate activitățile din laboratoarele 9-10, am testat codul pe plăcuța Basys3 și funcționează corect. Nu există erori de sintaxă, se generează bitstreamul, iar programul a fost testat pe placă și este funcțional complet.