|  |  |
| --- | --- |
| Instr.Nr. | Program |
| 1 | addi $1,$0,10 |
| 2 | addi $4,$0,8 |
| 3 | addi $2,$0,0 |
| 4 | addi $3,$0,0 |
| 5 | add $1,$2,$2 |
| 6 | addi $3,$3,1 |
| 7 | addi $1,$1,2 |
| 8 | beq $3,$4,1 |
| 9 | j 4 |

Raport

Mips pipeline:

1. Pentru mipsul pipeline, in cazul instructiunilor suplimentare s-au facut modificari in execution Unit pentru a include noile operatii.

|  |  |  |  |
| --- | --- | --- | --- |
| Tip R | Addition | add | 010 |
| Substraction | sub | 001 |
| Shift Left Logical | sll | 011 |
| Shift Right Logical | srl | 100 |
| Logical And | and | 101 |
| Logical OR | or | 110 |
| Logical XOR | xor | 000 |
| Set on Less than | slt | 111 |
| Tip I | Add Immediate | addi | 101 |
| Load Word | lw | 111 |
| Store Word | sw | 010 |
| Branch On Equal | beq | 011 |
| Load Immediate | li | 001 |
| Branch if not Equal | bne | 110 |
| Tip J | Jump | j | 100 |

Codul assembly trecut in biti este mai jos:

101\_000\_001\_0001010 --addi $1,$0,10

101\_000\_100\_0001000 --addi $4,$0,8

101\_000\_010\_0000000 --addi $2,$0,0

101\_000\_011\_0000000 --addi $3,$0,0

000\_000\_000\_000\_0\_010 --NoOP

000\_010\_010\_001\_0\_010 --add $1,$2,$2

000\_000\_000\_000\_0\_010 --NoOp

000\_000\_000\_000\_0\_010 --NoOp

101\_011\_011\_0000001 --addi $3,$3,1

101\_001\_001\_0000010 --addi $1,$1,2

011\_100\_011\_0000010 --beq $3,$4,2

100\_0000000000100 --jmp 4

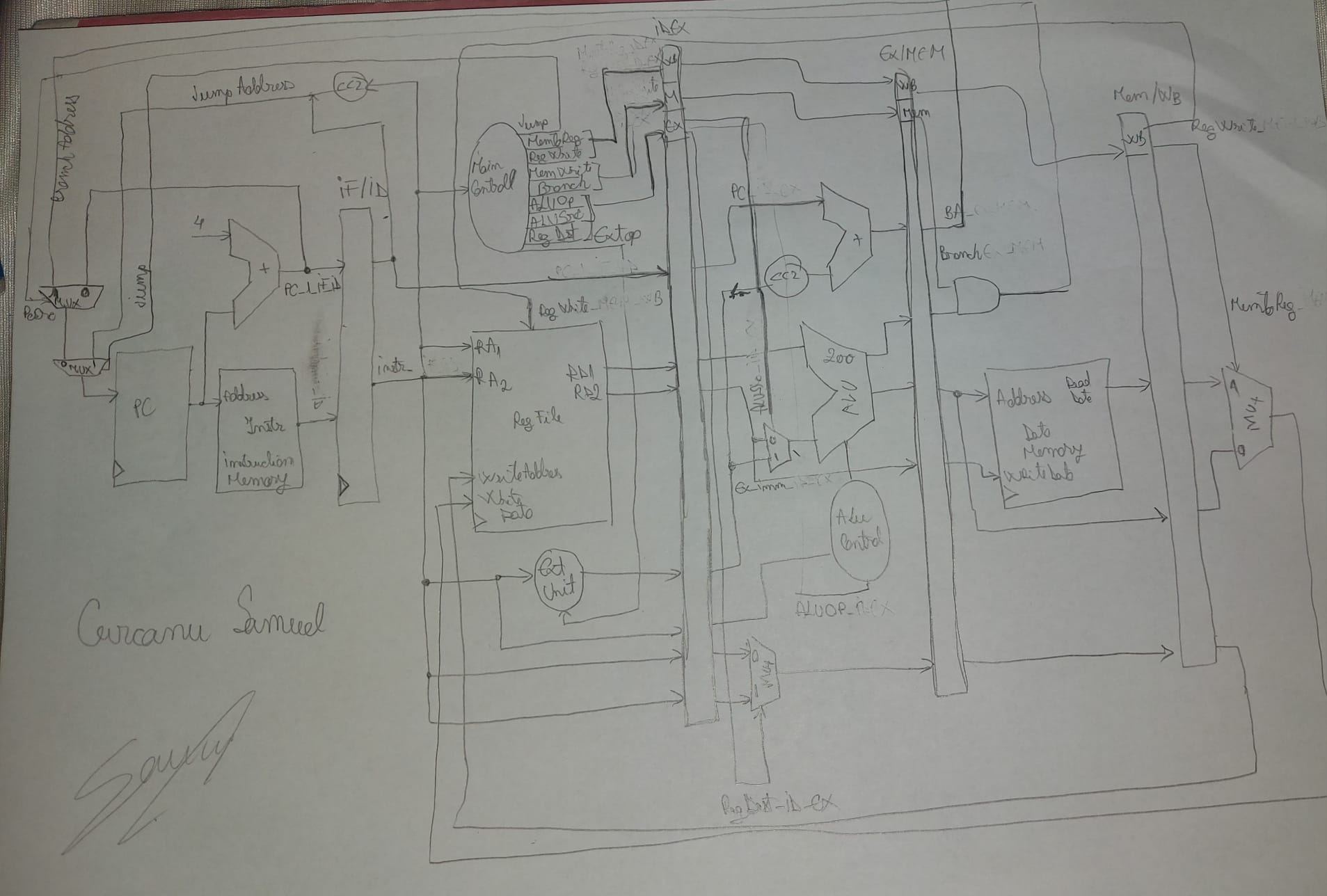
000\_000\_000\_000\_0\_010 --NoOp

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Reg Dest | Ext Op | ALUSrc | Branch | Jump | ALU op | MEM Write | Memto Reg |
| R type op | 1 | 0 | 0 | 0 | 0 | 000 | 0 | 0 |
| addi | 0 | 1 | 0 | 0 | 0 | 001 | 0 | 0 |
| lw | 0 | 1 | 0 | 0 | 0 | 001 | 0 | 1 |
| sw | x | 1 | 0 | 0 | 0 | 001 | 1 | x |
| beq | x | 1 | 1 | 1 | 0 | 010 | 0 | x |
| bne | 0 | 1 | 0 | 1 | 0 | 010 | 0 | 0 |
| li | 0 | 1 | 0 | 0 | 0 | 001 | 0 | 0 |
| j | x | x | x | x | 1 | 100 | 0 | 0 |

b.

|  |  |  |  |
| --- | --- | --- | --- |
| IF/ID | ID/EX | EX/MEM | MEM/WB |
| IF\_ID(31 downto 16) <= PCinc | ID\_EX(81) <= MemtoReg | EX\_MEM(55 downto 54) <= ID\_EX(81 downto 80) | MEM\_WB(36 downto 35) <= EX\_MEM(55 downto 54) |
| IF\_ID(15 downto 0) <= Instr | ID\_EX(80) <= RegWrite | EX\_MEM(53 downto 52) <= ID\_EX(79 downto 78) | MEM\_WB(34 downto 19) <= EX\_MEM(34 downto 19) |
|  | ID\_EX(79) <= MemWrite | EX\_MEM(51 downto 36) <= BA | MEM\_WB(18 downto 3) <= EX\_MEM(18 downto 3) |
|  | ID\_EX(78) <= Branch | EX\_MEM(35) <= '0' | MEM\_WB(2 downto 0) <= EX\_MEM(2 downto 0) |
|  | ID\_EX(77 downto 73) <= ALUOp & ALUSrc & RegDst | EX\_MEM(34 downto 19) <= ALURes |  |
|  | ID\_EX(72 downto 57) <= IF\_ID(31 downto 16) | EX\_MEM(18 downto 3) <= ID\_EX(40 downto 25) |  |
|  | ID\_EX(56 downto 41) <= RD1 | EX\_MEM(2 downto 0)<=WA |  |
|  | ID\_EX(40 downto 25) <= RD2 |  |  |
|  | ID\_EX(24 downto 9) <= Ext\_imm |  |  |
|  | ID\_EX(8 downto 6) <= func |  |  |
|  | ID\_EX(5 downto 3) <= rd |  |  |
|  | ID\_EX(2 downto 0) <= rt |  |  |

c.



d.

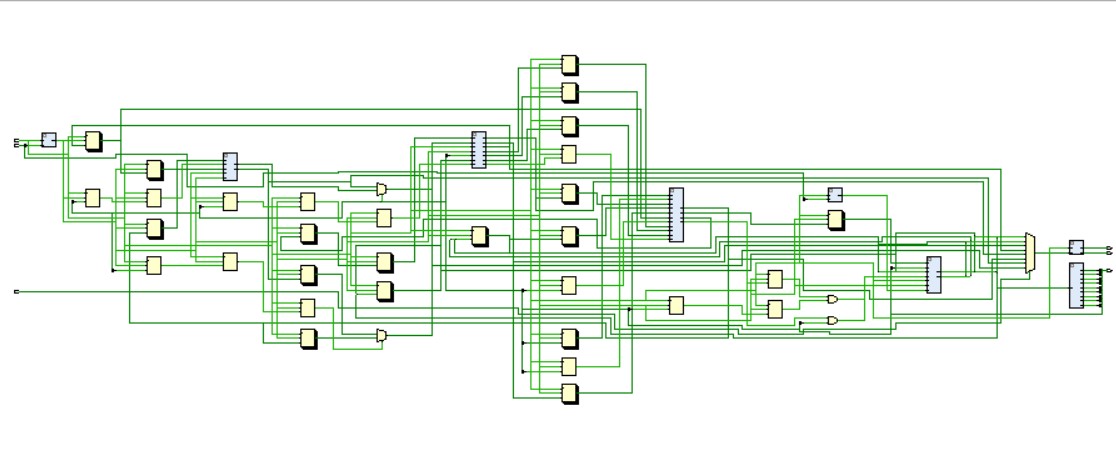
Intre instructiunea 3 si instructiunea 5 este hazard Raw dupa registrul $2 si intre instructiunea 5 si 6 este hazard Raw dupa registrul $1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instr/Clk | CC1 | CC2 | CC3 | CC4 | CC5 | CC6 | CC7 | CC8 | CC9 |
| addi $1,$0,10 | IF | ID | EX | MEM | WB($1) |  |  |  |  |
| addi $4,$0,8 |  | IF | ID($0) | EX | MEM | WB($4) |  |  |  |
| addi $2,$0,$0 |  |  | IF | ID($0) | EX | MEM | WB($2) |  |  |
| addi $3,$0,0 |  |  |  | IF | ID($0) | EX | MEM | WB($3) |  |
| add $1,$2,$2 |  |  |  |  | IF | ID($2) | EX | MEM | WB($1) |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instr/Clk | CC4 | CC5 | CC6 | CC7 | CC8 | CC9 | CC10 | CC11 |
| addi $3,$0,0 | IF | ID($0) | EX | MEM | WB($3) |  |  |  |
| add $1,$2,$2 |  | IF | ID($2) | EX | MEM | WB($1) |  |  |
| addi $1,$1,2 |  |  | IF | ID($1) | EX | MEM | WB($1) |  |
| beq $3,$4,1 |  |  |  | IF | ID($4) | EX | MEM | WB |

|  |  |
| --- | --- |
| Instr. Nr. | Program |
| 1 | addi $1,$0,10 |
| 2 | addi $4,$0,8 |
| 3 | addi $2,$0,$0 |
| 4 | addi $3,$0,0 |
| 5 | NoOp |
| 6 | add $1,$2,$2 |
| 7 | NoOp |
| 8 | NoOp |
| 9 | beq $3,$4,1 |

e. nu exista parti incomplete.

f. nu exista erori 

e. Cred ca nu functioneaza in totalitate corect , nu a fost testat pe placa