**Configurare registre MIPS16 Pipeline – varianta 1**

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 (<msb>) din care reiese dimensiunea totală alocată registrului.

|  |  |  |  |
| --- | --- | --- | --- |
| **REG\_IF\_ID(31** – **0)** | **REG\_ID\_EX(83**– **0)** | **REG\_EX\_MEM(58** – **0)** | **REG\_MEM\_WB(36**– **0)** |
| Instruction(31 – 16) | MemtoReg(83) | MemtoReg(58) | MemtoReg(36) |
| PC + 1(15 – 0) | RegWrite(82) | RegWrite(57) | RegWrite(35) |
|  | MemWrite(81) | MemWrite(56) | MemData(34 downto 19) |
|  | Branch1(80) | Branch1(55) | ALURes(18 downto 3) |
|  | Branch2(79) | Branch2(54) | MUXout(2 downto 0) |
|  | Branch3(78) | Branch3(53) |  |
|  | ALUOp(77 downto 76) | BA(52 downto 37) |  |
|  | ALUSrc(75) | Zero(36) |  |
|  | RegDst(74) | Sign(35) |  |
|  | Instructiton(73 downto 58) | ALURes(34 downto 19) |  |
|  | RD1(57 downto 42) | MUXout(2 downto 0) |  |
|  | RD2(41 downto 26) |  |  |
|  | Ext\_Imm(25 downto 10) |  |  |
|  | PC+1(6 downto 4)(9 downto 7) |  |  |
|  | PC+1(9 downto 7)(6 downto 4) |  |  |
|  | Sa(3) |  |  |
|  | Func(2 downto 0) |  |  |
|  |  |  |  |

URL: <https://drive.google.com/file/d/1Yw18tfjFjo-v897vVpd8KNGnpx5U8i9n/view?usp=sharing>

**Configurare registre MIPS16 Pipeline – varianta 2**

Se introduc pe coloane numele utilizate în codul VHDL pentru semnalele de date și control implementate ca registre, pe categorii. Se introduce în paranteză dimensiunea în biți.

|  |  |  |  |
| --- | --- | --- | --- |
| **IF/ID** | **ID/EX** | **EX/MEM** | **MEM/WB** |
| Instruction (16) | MemtoReg(1) | MemtoReg(1) | MemtoReg(1) |
| PC (16) | RegWrite(82) | RegWrite(1) | RegWrite(1) |
|  | MemWrite(1) | MemWrite(1) | MemData(16) |
|  | Branch1(1) | Branch1(1) | ALURes(16) |
|  | Branch2(1) | Branch2(1) | MUXout(3) |
|  | Branch3(1) | Branch3(1) |  |
|  | ALUOp(16) | BA(16) |  |
|  | ALUSrc(1) | Zero(1) |  |
|  | RegDst(1) | Sign(1) |  |
|  | Instructiton(16) | ALURes(16) |  |
|  | RD1(16) | MUXout(3) |  |
|  | RD2(16) |  |  |
|  | Ext\_Imm(16) |  |  |
|  | PC(3) |  |  |
|  | PC(3) |  |  |
|  | Sa(1) |  |  |
|  | Func(3) |  |  |
|  |  |  |  |

URL: <https://drive.google.com/file/d/1Yw18tfjFjo-v897vVpd8KNGnpx5U8i9n/view?usp=sharing>

OBS) Nu am o denumire speciala pentru iesirea din mux-ul ce calculeaza adresa de scriere in memorie, in program m-am folosit de registrii pentru a o specifica. In tabelele de mai sus am denumit-o MUXout.