|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instructiune | RegDst | RegWrite | ALUSrc | ExtOp | ALUCtrl2:0 | MemWrite | MemtoReg | Branch | Jump |
| add | 1 | 1 | 0 | 0 | 0000 | 0 | 0 | 0 | 0 |
| Sub | 1 | 1 | 0 | 0 | 0001 | 0 | 0 | 0 | 0 |
| Sll | 1 | 1 | 0 | 0 | 0010 | 0 | 0 | 0 | 0 |
| Slr | 1 | 1 | 0 | 0 | 0011 | 0 | 0 | 0 | 0 |
| And | 1 | 1 | 0 | 0 | 0100 | 0 | 0 | 0 | 0 |
| Or | 1 | 1 | 0 | 0 | 0101 | 0 | 0 | 0 | 0 |
| Xor | 1 | 1 | 0 | 0 | 0110 | 0 | 0 | 0 | 0 |
| Sra | 1 | 1 | 0 | 0 | 0111 | 0 | 0 | 0 | 0 |
| Addi | 0 | 1 | 1 | 1 | 0000 | 0 | 0 | 0 | 0 |
| Andi | 0 | 0 | 1 | 1 | 1001 | 0 | 0 | 0 | 0 |
| Beq | 0 | 0 | 0 | 0 | 0001 | 0 | 1 | 1 | 0 |
| Bgtz | 0 | 0 | 0 | 1 | 1010 | 0 | 0 | 1 | 0 |
| Lw | 0 | 1 | 1 | 1 | 1000 | 0 | 1 | 0 | 0 |
| Sw | 1 | 0 | 1 | 1 | 1000 | 1 | 1 | 0 | 0 |
| J | 0 | 0 | 0 | 0 | - | 0 | 0 | 0 | 1 |

Proiectul a fost testat pe o placurta basys3 si merge tot inafara de partea de executare a instructiunilor LW si SW care duce la un raspuns eronat. Eu cred ca problema la aceste instructiuni este in unitatea de executie la implementare. Am folosit switch-urile 1, 2 si 3.

Am ales in plus instructiunile: Bgtz, xor, andi si sra. Valorile semnalelor de control sunt afisate in tabelul de mai sus.

Bgtz are rolul de a compara valorile a doua semnale si de a sari la instructiunea cu adresa specificata daca al doilea este mai mare decat primul, RTLIf $s > 0 then PC ⇓ (PC + 4) + (SE(offset) << 2) else PC ⇓ PC + 4;, format: 000111 sssss 00000 oooooooooooooooo.

Xor are rolul de a face operatia xor intre doi registrii iar rezultatul este memorat in alt registru, RTL: $d ⇓ $s ^ $t; PC ⇓ PC + 4;, format: 000000 sssss ttttt ddddd 00000 100110.

ANDI are rolul de a face operatia de and intre un registru si o valoare imediata iar rezultatul este memorat in alt registru, RTL: $t ⇓ $s & ZE(imm); PC ⇓ PC + 4;, format: 001100 sssss ttttt iiiiiiiiiiiiiiii.

Sra are rolul de a face shiftare aritmetica la dreapta a unui registru cu o valoare imediata iar rezultatul este memorat in alt registru, RTL: $d ⇓ $t >> h; PC ⇓ PC + 4;, format: 000000 00000 ttttt ddddd hhhhh 000011.