| ≨ 1 ₹ | Msgs | | | | | | | | | | | | | | | |
|---|---------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|--|----------|----------|----------|--------|
| /MipsProcessor_TB/Clock | 1'h1 | | | | | | | | | | | | | | | |
| → /MipsProcessor_TB/PcOut | 32h00000000 | 00000000 | | | | 00000001 | | | | 00000002 | | | 00000000 | | | |
| ■ / MipsProcessor_TB/Instruction | 32h20010003 | 20010003 | | | | 20020007 | | | | 08000000 | | | 20010003 | | | |
| ■ / /MipsProcessor_TB/ReadData1 | 32'h00000000 | 00000000 | | | | | | | | | | | | | | |
| <u>★</u> - /MipsProcessor_TB/ReadData2 | 32'hxxxxxxxxx | | 00000000 | | 00000003 | | 00000003 | | 00000007 | 00000000 | | | 00000003 | 00000000 | | 100000 |
| <u>→</u> /MipsProcessor_TB/AluResult | 32'h00000000 | 00000000 | | 00000003 | | | | 00000007 | | | 00000000 | | | | 00000003 | |
| | 32'hxxxxxxxxx | | | | | | | | | | | | | | | |
| <u>★</u> <u>/</u> /MipsProcessor_TB/WriteData | 32'h00000000 | 00000000 | | 00000003 | | | | 00000007 | | | 00000000 | | | | 00000003 | |
| I Image: Ima | | 01 | | | | 02 | | | | 00 | | | 01 | | | |
| I → /MipsProcessor_TB/op I → / MipsProcessor_TB/op | 4'h2 | 2 | | | | | | | | | | | | | | |
| ■ / MipsProcessor_TB/ALUop | 2'h0 | 0 | | | | | | | | | | | | | | |
| /MipsProcessor_TB/RegDst | 1'h0 | | | | | | | | | | | | | | | |
| /MipsProcessor_TB/Branch | 1'h0 | | | | | | | | | | | | | | | |
| /MipsProcessor_TB/MemRead | 1'h1 | | | | | | | | | | | | | | | |
| /MipsProcessor_TB/MemtoReg | 1'h0 | | | | | | | | | | | | | | | |
| /MipsProcessor_TB/MemWrite | 1'h0 | | | | | | | | | | | | | | | |
| /MipsProcessor_TB/ALUsrc | 1'h1 | | | | | | | | | | | | | | | |
| /MipsProcessor_TB/RegWrite | 1'h1 | | | | | | | | | | | | | | | |
| /MipsProcessor_TB/Jump | 1'h0 | | | | | | | | | | | | | | | |
| /MipsProcessor_TB/Jal | 1'h0 | | | | | | | | | | | | | | | |
| /MipsProcessor_TB/Jr | 1'h0 | | | | | | | | | | | | | | | |
| /MipsProcessor_TB/Zero | 1'hx | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |

در شکل بالا مشخص است که پس از اجرای دستورات 0 و 1 در دستور شماره 2 قصد داریم به ابتدای برنامه جامپ کنیم همانطور ک معلوم است این جامپ به درستی انجام شده روش جامپ به این نحو است که 6 بیت پر ارزش رجیستر 1+2 بیت اول دستور concat شده و این حاصل وارد یک ماکس میشود که در صورتی که select آن یک باشد این آدرس در گام بعدی وارد 1 می شود. Select را کافی است کنترل 1 انتخاب کنیم البته در قسمت بعد برای 1 هم این حاصل را نیاز داریم به همین دلیل 1 select نهایت و 1 و

| <u> </u> | | | | | | | | | | 12 | | | | | | |
|--|---------------|----------|----------|-------------------|----|---------|-------------------|-----|----------|----|------|----------|---|----------|--|--|
| <u></u> | Msg: | 5 | | | | | | | | | | | | | | |
| //MipsProcessor_TB/Clock | 1'h1 | | | | | | | | | | | | | | | |
| ★── /MipsProcessor_TB/PcOut | 32'h00000006 | 00000000 | | | | | 0000000 | | | | | 0000000 | 5 | | | |
| ■ / MipsProcessor_TB/Instruction | 32'hxxxxxxxxx | 20010003 | | | | | Oc000009 | j . | | | | 003f1818 | | | | |
| <u>∓</u> → /MipsProcessor_TB/ReadData1 | 32'hxxxxxxxxx | 00000000 | | | | | | | | | | 0000000 | 3 | | | |
| <u>I</u> → /MipsProcessor_TB/ReadData2 | 32'hxxxxxxxxx | | 00000000 | 3 | | 0000003 | 00000000 |) | | | | 0000000 | 2 | | | |
| <u>I</u> | 32'hxxxxxxxxx | 00000000 | 3 | 0000000 | 13 | | | | 00000000 | | | | | 00000005 | | |
| <u>∓</u> - | 32'hxxxxxxxxx | | | | | | | | | | | | | | | |
| | 32'hxxxxxxxxx | 00000000 | 3 | 0000000 | 13 | | 00000002 | 2 | | | | 00000000 | | 00000005 | | |
| MipsProcessor_TB/WriteRegister | 5'hxx | 01 | | | | | 1f | | | | | 03 | | | | |
| → /MipsProcessor_TB/op | 4'h2 | 2 | | | | | | | | | | | | | | |
| ■ / MipsProcessor_TB/ALUop | 2'h2 | 0 | | | | | | | | | | 2 | | | | |
| /MipsProcessor_TB/RegDst | 1'h1 | | | \longrightarrow | | | | | | | | | | | | |
| /MipsProcessor_TB/Branch | 1'h0 | | | | | | | | | | | | | | | |
| /MipsProcessor_TB/MemRead | 1'h0 | | | | | | | | | | | | | | | |
| /MipsProcessor_TB/MemtoReg | 1'h0 | | | \blacksquare | | | | | | | | | _ | | | |
| /MipsProcessor_TB/MemWrite | 1'h0 | | | + | | | | | | | | | _ | | | |
| /MipsProcessor_TB/ALUsrc | 1'h0 | | | | | | \longrightarrow | | | | | _ | _ | | | |
| /MipsProcessor_TB/RegWrite | 1'h1 | | | | | | | | | | | | | | | |
| /MipsProcessor_TB/Jump | 1'h0 | | | ++ | | | | | | | | | _ | | | |
| /MipsProcessor_TB/Jal | 1'h0 | | | \blacksquare | | | | | | | | | | | | |
| / MipsProcessor_TB/Jr | 1'h1 | | | | | | | | | | | | | | | |
| /MipsProcessor_TB/Zero | 1'hx | | | \blacksquare | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |

 $memory[0] = 32'b001000_00000_00001_00000000000011; // \ addi \ R1, \ R0, \ 3// \ Addi \ R1, \ R0, \ A// \ Addi \ R1, \$

 $memory[1] = 32'b000011_0000000000000000000000101; // jal 5//$

 $memory[5] = 32'b000000_00001_111111_00011_00000_011000; // addi R3, R1, R31// addi R3, R31// addi R3// addi R3//$

در شکل بالا مشخص است که ابتدا عدد 8 داخل رجیستر 1 ریخته شده سپس با فراخوانی 1 آدرس 1 + 1 یعنی 2 باید در رجیستر شماره 1 ذخیره شود و سپس به آدرس پرش یعنی 1 پرش کنیم. در گام بعد دستور شماره 1 دخیره شود که نشان می دهد پرش صحیح بوده و سپس مقادیر رجیستر های 1 و 1 را میخواند که همانطور که مشخص است 1 read data read data مقادیر 1 و 1 را اتخاذ کرده اند که نشان می دهد هم دستور اول به درستی اجرا شده و هم دستور دوم علاوه بر پرش آدرس بازگشت را درستی ذخیره کرده است.

| <u>.</u> | | Msgs | | | | | | | | | | | | | | | | |
|--|--------------------------------|----------------|---------------|----------|----------|---------------|----------|----------|--------|---------------|---------------|---------------|---|---------------|-----------|---------------|----------|----------|
| | MipsProcessor_TB/Clock | 1'h1 | $\overline{}$ | | | $\overline{}$ | | | \neg | $\overline{}$ | $\overline{}$ | $\overline{}$ | | $\overline{}$ | | $\overline{}$ | | Ŧ |
| | MipsProcessor_TB/PcOut | 32'h0000001a | 0000000 | 0 | | | 00000001 | | _= | | 100000002 | | | | 100000003 | | | 00 |
| | MipsProcessor_TB/Instruction | 32'hxxxxxxxxx | 2001000 | | | | 20020003 | | | | 00221818 | | | | ac22000a | | | 8c. |
| | MipsProcessor TB/ReadData1 | 32'hxxxxxxxxx | 0000000 | | | | | | | | 00000003 | | | | | | | |
| - ◆ /* | MipsProcessor_TB/ReadData2 | 32'hxxxxxxxxx | | 00000000 | 7 | 00000003 | | 00000003 | | | | | | | | | | |
| - ◆ /1· | MipsProcessor_TB/AluResult | 32hxxxxxxxxx | 0000 | | 00000003 | | | | | | | 00000006 | | | | 0000181b | 0000000d | |
| - ♦ /1 | MipsProcessor_TB/ReadData | 32h00000003 | | | | | | | | | | | | | | | | |
| ⊢ ♦ /\ | MipsProcessor_TB/WriteData | 32'h00000003 | 0000 | | 00000003 | | | | | | | 00000006 | | | | 0000181b | 0000000d | 5 |
| - ◆ /\ | MipsProcessor_TB/WriteRegister | 5'hxx | 01 | | | | 02 | | | | 03 | | | | 100 | | | 01 |
| - ♦ /1 | MipsProcessor_TB/op | 4'h6 | 2 | | | | | | | | | | | | | | | |
| ⊢ ♦ /\ | MipsProcessor_TB/ALUop | 2'h1 | 0 | | | | | | | | 2 | | | | 0 | | | |
| ♦ /١٠ | MipsProcessor_TB/RegDst | 1'h0 | | | | | | | | | | | | | | | | |
| ♠ /N | MipsProcessor_TB/Branch | 1'h1 | | | | | | | | | | | | | | | | |
| ♠ /N | MipsProcessor_TB/MemRead | 1'h0 | | | | | | | | | | | | | | | | |
| ♦ /١٠ | MipsProcessor_TB/MemtoReg | 1'h1 | | | | | | | | | | | | | | | | |
| // | MipsProcessor_TB/MemWrite | 1'h0 | | | | | | | | | | | | | | | | |
| ♦ /\(\begin{array}{cccccccccccccccccccccccccccccccccccc | MipsProcessor_TB/ALUsrc | 1'h0 | | | | | | | | | | | | | | | | |
| ♦ /\(\begin{array}{cccccccccccccccccccccccccccccccccccc | MipsProcessor_TB/RegWrite | 1'h0 | | | | | | | | | | | | | | | | |
| ♦ /١٠ | MipsProcessor_TB/Jump | 1'h0 | | | | | | | | | | | | | | | | |
| ♦ /\(\begin{array}{cccccccccccccccccccccccccccccccccccc | MipsProcessor_TB/Jal | 1'h0 | | | | | | | | | | | | | | | | |
| ♦ /\(\lambda \) / \(\lam | MipsProcessor_TB/Jr | 1'h0 | | | | | | | | | | | | | | | | |
| ♦ /١٠ | MipsProcessor_TB/Zero | 1'hx | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | عقام |
| | | | | | | | | | | | | | | | | | | |
| <u>•</u> | | Msgs | | | | | | | | | | | | | | | | |
| ♦ / | MipsProcessor_TB/Clock | 1h1 | | | | | | | | | | | | | | | | |
| n-💠 /I | MipsProcessor_TB/PcOut | 32'h0000001a | 000000 | 000000 | 004 | | | 0000000 | 5 | | | 000000 | a | | | | | |
| 1-4-1 | MipsProcessor_TB/Instruction | 32'hxxxxxxxxxx | ac22000 | a 8c4100 | 00a | | | 1022001 | 1 | | | | | | | | | |
| 1 | MipsProcessor_TB/ReadData1 | 32'hxxxxxxxxx | 000000 |)3 | | | | | | | | | | | | | | |
| • - / | MipsProcessor_TB/ReadData2 | 32'hxxxxxxxxx | 000000 | 03 | _ | 000000 | 0.3 | | | | | | | | | | | |

| ≨ 1 ← | Msgs | | | | | | | | | | | | |
|---|----------------|----------|----------|----------|----------|--------|---|----------|--|----------|--|--|---|
| /MipsProcessor_TB/Clock 1 | ľh1 | \Box | | oxdot | | \Box | egthinspace = egt | | | \Box | | | |
| ■ / MipsProcessor_TB/PcOut 3 | 32'h0000001a | 00000003 | 00000004 | | | | 00000005 | | | 0000001a | | | |
| | 32'hxxxxxxxxxx | ac22000a | 8c41000a | | | | 10220014 | | | | | | |
| <u>★</u> /MipsProcessor_TB/ReadData1 3 | 32'hxxxxxxxxx | 00000003 | | | | | | | | | | | _ |
| <u>→</u> /MipsProcessor_TB/ReadData2 3 | 2'hxxxxxxxx | 00000003 | | } | 00000003 | | | | | | | | |
| /MipsProcessor_TB/AluResult | 2'hxxxxxxxx | 0000000d | | | | | | 00000000 | | | | | |
| /MipsProcessor_TB/ReadData | 32'h00000003 | | | 00000003 | | | | | | | | | |
| <u>★</u> - /MipsProcessor_TB/WriteData 3 | 32'h00000003 | 0000000d | - | 00000003 | | | | | | | | | |
| | | 00 | 01 | | | | 02 | | | | | | |
| | ih6 | 2 | | | | | 6 | | | | | | |
| | ?h1 | 0 | | | | | 1 | | | | | | |
| | ľh0 | | | | | | | | | | | | |
| | ľh1 | | | | | | | | | | | | |
| | ľh0 | | | | | | | | | | | | |
| | ľh1 | | | | | | | | | | | | |
| | ľh0 | | | | | | | | | | | | |
| | ľh0 | | | | | | | | | | | | |
| | ľh0 | | | | | | | | | | | | |
| | ľh0 | | | | | | | | | | | | |
| | ľh0 | | | | | | | | | | | | |
| | ľh0 | | | | | | | | | | | | |
| /MipsProcessor_TB/Zero 1 | ľhx | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |

memory[0] = 32'b001000_00000_00001_000000000000011 // addi R1, R0, 3

memory[1] = 32'b001000_00000_00010_000000000000011 // addi R2, R0, 3

 $memory[2] = 32'b000000_00001_00010_00011_00000_011000 // add R3, R1, R2$

 $memory[3] = 32'b101011_00001_00010_00000000001010 // sw R2, 10(R1)$

memory[4] = 32'b100011 00010 00001 00000000001010 // lw R1, 10(R2)

 $memory[5] = 32'b000100_00001_00010_000000000010100 // beq R1, R2, 20$

در شکل بالا مشخص است که ابتدا مقدار 3 در رجیستر 1 ذخیره شده سپس 3 در رجیستر 2 ذخیره شده و در دستورشماره 2 حاصل دو رجیستر خوانده می شود و با هم جمع زده می شود در گام بعد حاصل alu وارد write data بانک رجیستر شده و مقدار 6 در رجیستر 3 ذخیره می گردد.

در گام بعدی دستور شماره 8 اجرا شده و این دستور برای ذخیره کردن مقدار رجیستر 2 در خانه شماره 4 1 حافظه است در دستور بعد قصد داریم همین مقدار را از حافظه بخوانیم و روی رجیستر 4 ذخیره کنیم و با توجه به اینکه خروجی read_data مقدار 4 شده یعنی دستور 4 به درستی اجرا شده بود سپس رجیستر های 4 و 4 خوانده شده و با هم مقایسه می شوند چون دستور beq است کنترل بعدی 4 در این عنال است و zero ی خروجی 4 نیز فعال می شود پس در گام بعد به دستور 4 یعنی 4 پرش می کنیم که در این خانه از دستورات دستوری وجود ندارد. به طور کلی پرش beq از نوع نسبی و پرش های 4 از دستورات دستوری وجود ندارد. به طور کلی پرش beq از نوع نسبی و پرش های 4 از دستورات دستوری وجود ندارد.

| /MipsProcessor_TB/Clock | 1h1 | | | | | | | | | | | | | | |
|---------------------------------|----------------|----------|----------|----------|----------|----------|----------|--|----------|----------|----------|----------|----------|----------|--|
| /MipsProcessor_TB/PcOut | 32'h00000002 | 00000000 | | | | 00000001 | | | 00000003 | | | | 00000004 | | |
| / MipsProcessor_TB/Instruction | 32'h02221818 | 20020007 | | | | 0c000003 | | | 20110004 | | | | 03e00008 | | |
| /MipsProcessor_TB/ReadData1 | 32'h00000004 | 00000000 | | | | | | | | | | | 00000002 | | |
| /MipsProcessor_TB/ReadData2 | 32'h00000007 | | 00000000 | } | 00000007 | 00000000 | | | | 00000000 | 00000003 | 00000004 | 00000000 | | |
| /MipsProcessor_TB/AluResult | 32'h0000000b | 00000000 | | 00000007 | | | 00000000 | | | 00000003 | 00000004 | | | 00000002 | |
| /MipsProcessor_TB/ReadData | 32'hxxxxxxxxxx | | | | | | | | | | | | | | |
| /MipsProcessor_TB/WriteData | 32'h0000000b | 00000000 | | 00000007 | | 00000002 | | | 00000000 | 00000003 | 00000004 | | | 00000002 | |
| /MipsProcessor_TB/WriteRegister | 5ħ03 | 02 | | | | 1f | | | 11 | | | | 00 | | |
| /MipsProcessor_TB/op | 4h2 | 2 | | | | | | | | | | | | | |
| /MipsProcessor_TB/ALUop | 2h2 | 0 | | | | | | | | | | | 2 | | |
| /MipsProcessor_TB/RegDst | 1h1 | | | | | | | | | | | | | | |
| / MipsProcessor_TB/Branch | 1h0 | | | | | | | | | | | | | | |
| /MipsProcessor_TB/MemRead | 1h0 | | | | | | | | | | | | | | |
| /MipsProcessor_TB/MemtoReg | 1ħ0 | | | | | | | | | | | | | | |
| /MipsProcessor_TB/MemWrite | 1ħ0 | | | | | | | | | | | | | | |
| /MipsProcessor_TB/ALUsrc | 1'h0 | | | | | | | | | | | | | | |
| /MipsProcessor_TB/RegWrite | 1h1 | | | | | | | | | | | | | | |
| /MipsProcessor_TB/Jump | 1h0 | | | | | | | | | | | | | | |
| /MipsProcessor_TB/Jal | 1'h0 | | | | | | | | | | | | | | |
| /MipsProcessor_TB/Jr | 1h1 | | | | | | | | | | | | | | |
| /MipsProcessor_TB/Zero | 1ħ0 | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

| \$ 1 ₹ 1 | Msgs | | | | | | | | | | | | | | |
|--|--------------|----------|----------|----------|--|----------|----------|--|----------|----------|----------|----------|----------|----------|--|
| /MipsProcessor_TB/Clock | 1'h1 | | | | | | | | | | | | | | |
| /MipsProcessor_TB/PcOut | 32h00000002 | 00000003 | 00000004 | | | 00000002 | | | 00000003 | | | | 00000004 | | |
| <u>I</u> | 32h02221818 | 20110004 | 03e00008 | | | 02221818 | | | 20110004 | | | | 03e00008 | | |
| I → /MipsProcessor_TB/ReadData1 I → /MipsProcessor_TB/ReadData1 | 32h00000004 | 00000000 | 00000002 | | | 00000004 | | | 00000000 | | | | 00000002 | | |
| ■ / MipsProcessor_TB/ReadData2 | 32h00000007 | 00000004 | 00000000 | | | 00000007 | | | 00000004 | 0000000Ь | 00001818 | 00000004 | 00000000 | | |
| MipsProcessor_TB/AluResult | 32h0000000b | 00000004 | | 00000002 | | | 0000000Ь | | | 00001818 | 00000004 | | | 00000002 | |
| <u>→</u> /MipsProcessor_TB/ReadData | 32hxxxxxxxxx | | | | | | | | | | | | | | |
| → /MipsProcessor_TB/WriteData | 32'h0000000b | 00000004 | | 00000002 | | | 0000000b | | | 00001818 | 00000004 | | | 00000002 | |
| ■ / /MipsProcessor_TB/WriteRegister 5 | 5'h03 | 11 | 00 | | | 03 | | | 11 | | | | 00 | | |
| | 4'h2 | 2 | | | | | | | | | | | | | |
| | 2'h2 | 0 | 2 | | | | | | (o | | | | 2 | | |
| | 1'h1 | | | | | | | | | | | | | | |
| | 1'h0 | | | | | | | | | | | | | | |
| | 1'h0 | | | | | | | | | | | | | | |
| | 1'h0 | | | | | | | | | | | | | | |
| | 1'h0 | | | | | | | | | | | | | | |
| | 1'h0 | | | | | | | | | | | | | | |
| | 1'h1 | | | | | | | | | | | | | | |
| | 1'h0 | | | | | | | | | | | | | | |
| /MipsProcessor_TB/Jal | 1'h0 | | | | | | | | | | | | | | |
| | 1'h1 | | | | | | | | | | | | | | |
| / /MipsProcessor_TB/Zero | 1'h0 | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

memory[0] = 32'b001000_00000_00010_00000000000111;//// Addi R2, R0, 7

 $memory[1] = 32'b000011_0000000000000000000000011; // jal 3 //$

memory[2] = 32'b000000 10001 00010 00011 00000 011000;//add R3, R17, R2//

memory[3] = 32'b001000_00000_10001_00000000000000000;//addi R17, 4//

memory[4] = 32'b000000 11111 00000 00000 00000 001000;//jr R31//

همانطور که درشکل مشخص است ابتدا عدد 7 در رجیستر 2 ذخیره شده سپس در دستور شماره 1 3 اموجود است که در طی اجرا ان باید به دستور شماره 3 پرش و آدرس 2 در رجیستر 31 ذخیره شود. همانطور که مشخص است دستور بعدی دستور شماره 3 است که اجرا میشود پس پرش درست بوده به علاوه در دستور شماره 3 مقدار 4 در داخل رجیستر 17 قرار میگیرد و در دستور بعد باید مقدار رجیستر بازگشت یعنی رجیستر شماره 31 خوانده شود و در گام بعد pc بیاربر این مقدار قرار بگیرد همانطور که مشخص است در گام بعد دستور شماره 2 اجرا میشود پس هم jal آدرس بازگشت را به درستی ذخیره کرده و هم jr به درستی پرش می کند. نحوه پیاده سازی این 2 دستور به این صورت خواهد بود که اگر دستور از نوع jal باشد پرش دقیقا مشابه jump توضیح داده شده خواهد بود ولی ذخیره آدرس بازگشت در رجیستر شماره 31 نیازمند 2 ماکس یکی در ورودی write data و دیگری در ورودی ورودی ای که در سخت افزار اولیه خواهد بود. که سلکت هر دو همان کنترل jal بوده و ماکس ورودی write data از میان رجیستر 18 و حاصل ماکس سخت افزار اولیه (که از بود یکی را انتخاب می کند و ماکس موجود در ورودی یکی را انتخاب می کند به علاوه دستور با با گذاشتن یک ماکس بر سر راه ورودی میان 2 بخش دستور رجیستر مقصد را مشخص میکرد) یکی را انتخاب می کند به علاوه دستور آز با گذاشتن یک ماکس بر سر راه ورودی و pc

pc + 1 یکی را انتخاب میکرد. سلکت این ماکس زمانی 1 می شود که دستور از نوع jr باشد یعنی func دستور jr متعلق به دستور jr باشد.