



Universidade do Vale do Itajaí Campus KobraSol

Arquitetura de Computadores
Avaliação 05

Grupo: Mauricio Macário de Farias Junior
André Paulão Aliardi
Professor: Douglas Rossi de Melo

São José, 2017, 08 de Dezembro

Resumo

Neste trabalho será abordado a implementação do suporte às instruções Jump (J), Jump Register (Jr) e Jump and Link (Jal) no MIPS Monociclo, utilizando o programa Quartus II 13.0sp1 (64-bit) Web Edition.

Implementação do Jump

Circuito “control”

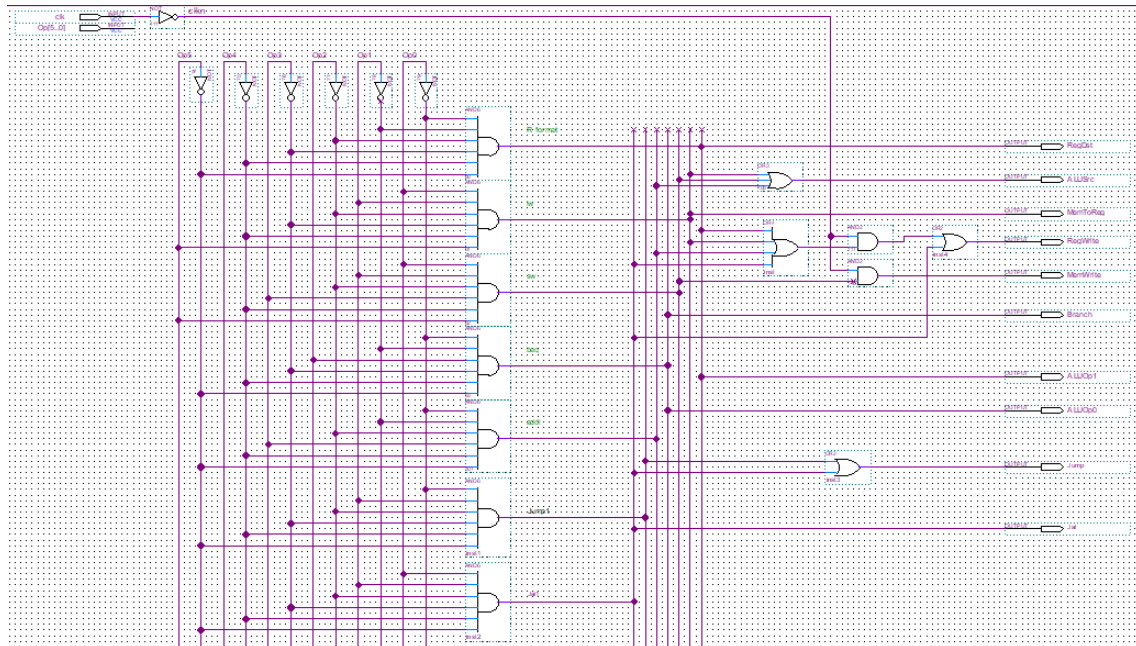


Figura 1

Esse circuito da figura 1 demonstra qual função será ativa no circuito geral, ativando saídas que ativam módulos.

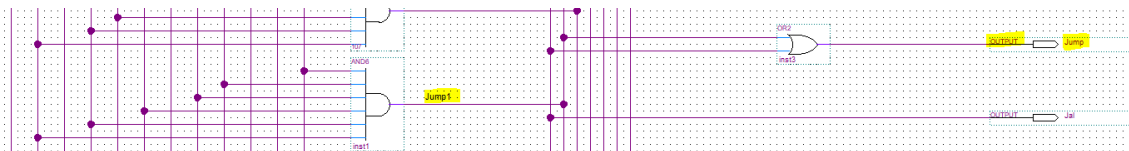


Figura 2

Na figura 2, é visto o circuito de detecção da função que irá ativar a saída “Jump” que será utilizada no circuito geral.

Implementação do Jump

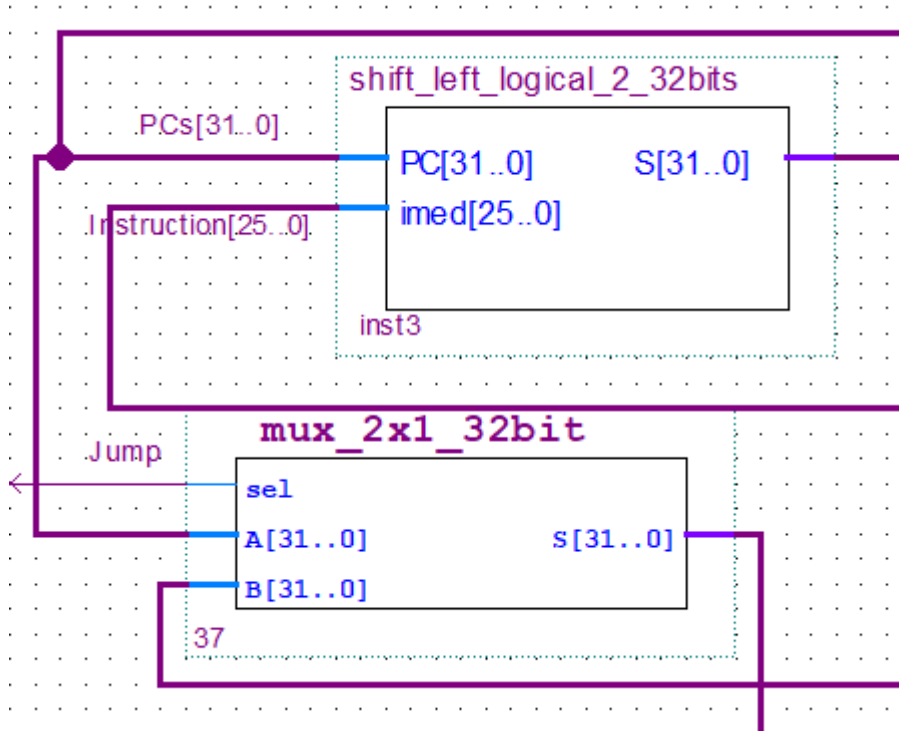


Figura 3

Circuito “Shift_left_logical”

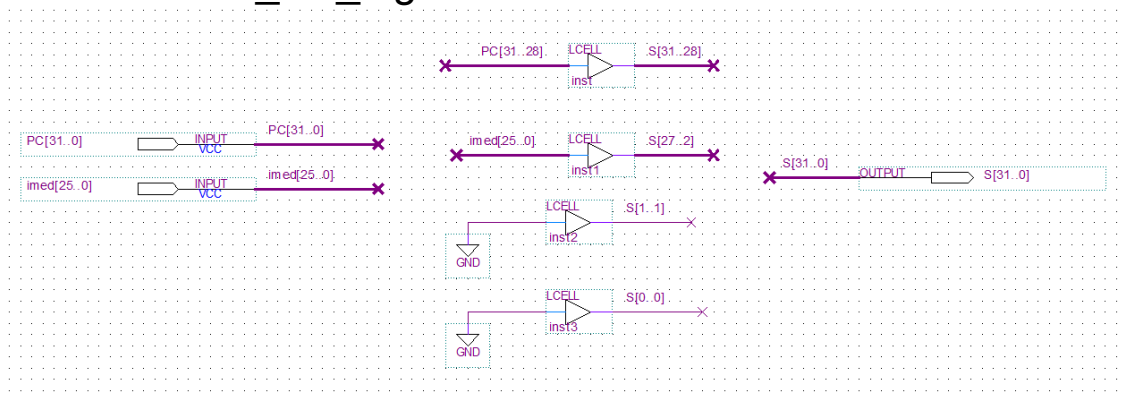


Figura 4

Esse circuito da figura 4, é utilizado para realizar o cálculo do PC. Ele pega os 4 bits mais significativos do PC (4 bits da esquerda) e concatena com os 26 bits do campo “Immediate” da função Jump. Os últimos 2 bits são preenchidos com 0.

Circuito “mux_2x1_32bit”

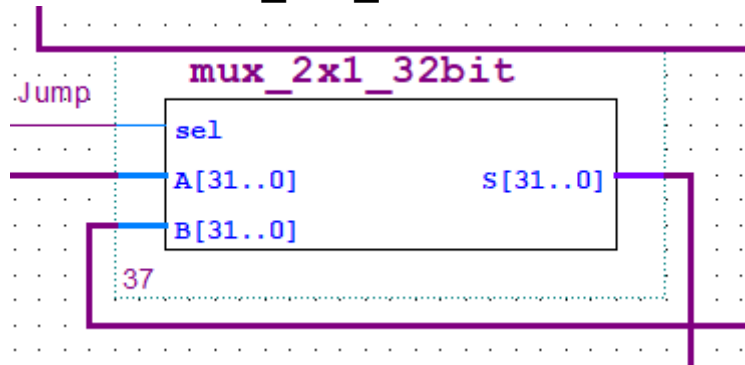


Figura 5

O circuito da figura 5, é utilizado para decidir se o Jump será utilizado ou não.

Comando “Jump and Link”

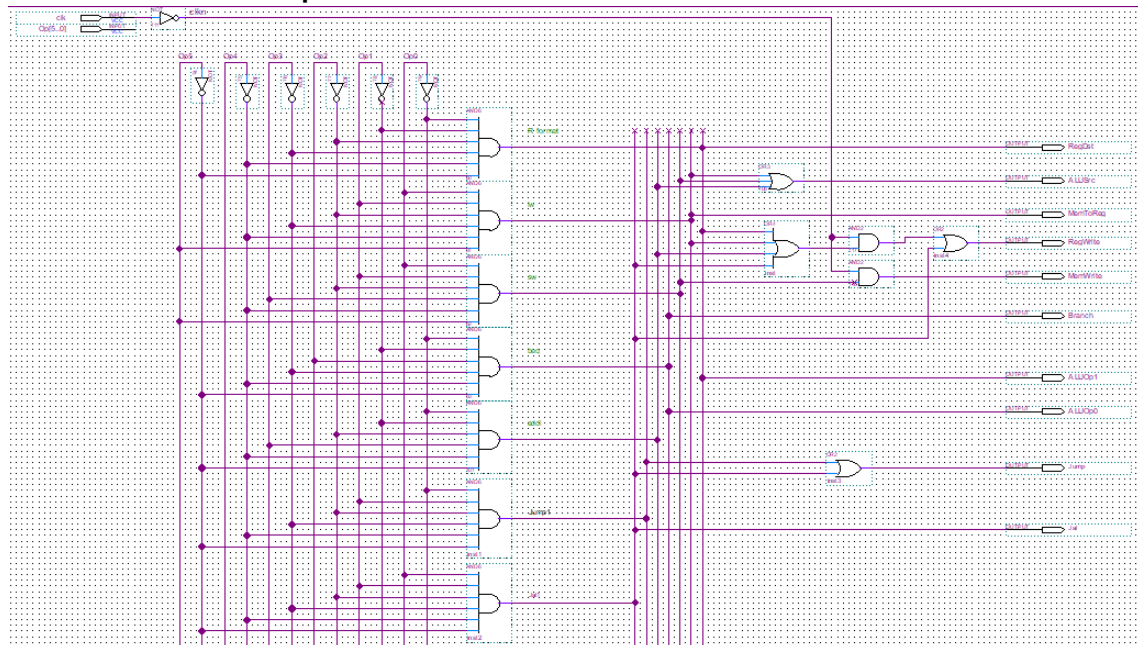


Figura 6

Circuito “Control” com o Jal

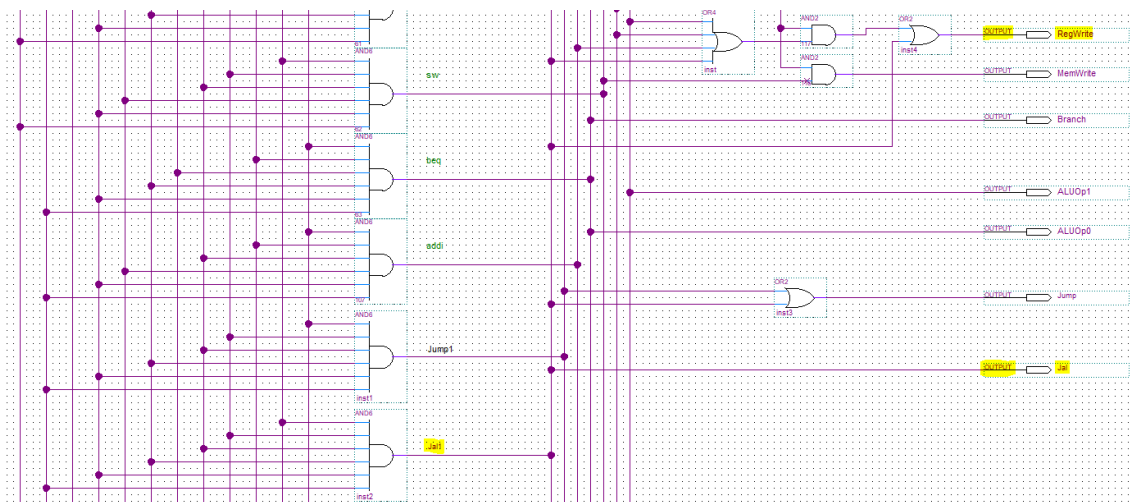


Figura 7

Acima, na figura 7, é visto o circuito de detecção da função que irá ativar a saída “Jump and Link” que será utilizada no circuito geral.

O “mux2x1_32bit”, que mostrado na figura 8, é utilizado para definir se o Jal será usado ou não.

Após o Jal ser ativado, ele irá salvar o valor do PC no registrador “\$ra” (Return Address) que será utilizado posteriormente no comando “Jr”. Para definir o numero do registrador, foi utilizado o circuito “Constant_31”, ele indica o 32º registrador, no caso, o \$ra, através de uma constante com 5 bits ativados ($2^5 = 32$). Essa implementação pode ser vista na figura 8.

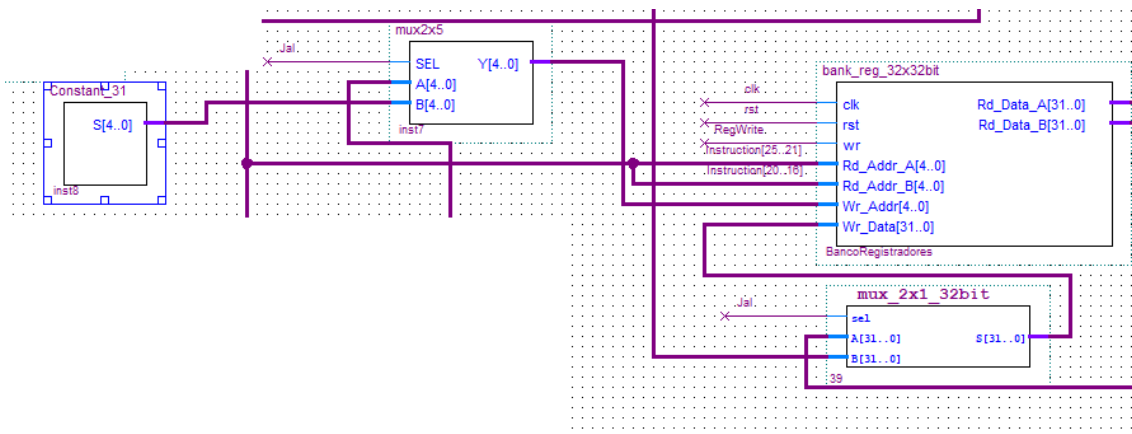


Figura 8

Comando “Jump Register” (jr)

O comando JR não tem uma detecção no controle, pois é ele uma função do tipo R (função que usa apenas registradores), por isso ele não aparece no circuito “Control”.

Esse comando é o complemento do Jal. Quando o Jal é utilizando, ele irá salvar o valor do PC no registrador \$ra, após isso, quando o comando Jr for executado, ele irá pegar o valor que foi salvo no \$ra e irá pular para essa linha de comando.

O AND só será ativado quando o RegWrite for ativado e ele nunca será ativado quando o Jr também estiver ativado, que pode ser conferido na figura 9.

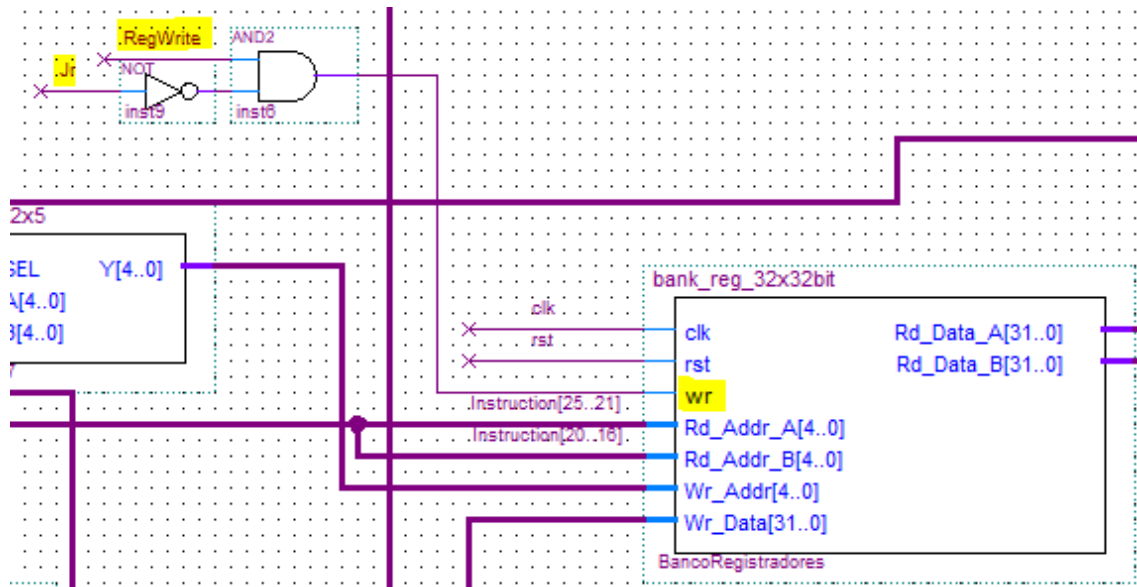


Figura 9

O “mux_2x1_32bit” é utilizado para ver se o valor será lido de um registrador ou de algum outro lugar.

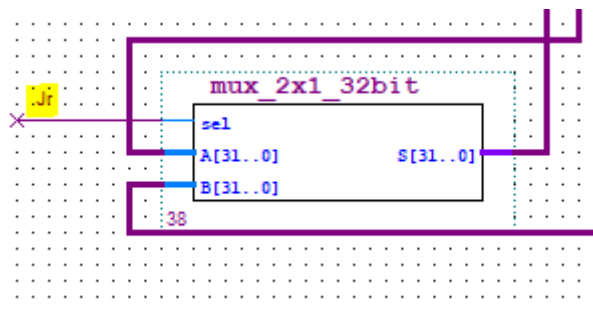


Figura 10

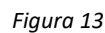
A saída do “Jr_Detector” é utilizada como parâmetro da porta AND e também no mux_2x1_32bit (que foram explicados acima).


```

.text                                     # segmento de código (programa)
    j      main
leaf_example:
    add     $t0, $a0, $a1    # $t0 = g + h
    add     $t1, $a2, $a3    # $t1 = i + j
    sub     $v0, $t0, $t1    # f = $t0 - $t1
    jr      $ra              # retorna do procedimento
main:
    addi    $a0, $zero, 4    # inicializa 1º parâmetro (g)
    addi    $a1, $zero, 3    # inicializa 2º parâmetro (h)
    addi    $a2, $zero, 2    # inicializa 3º parâmetro (i)
    addi    $a3, $zero, 1    # inicializa 4º parâmetro (j)
    jal     leaf_example     # chama o procedimento

```

Pode ser observado que logo na primeira instrução do programa, o PC é movido para 00400014, por conta do comando Jump que é executado. Na sexta instrução do programa, o comando Jal é executado movendo o PC para 00400004. Na décima instrução, o Jr é executado e o PC retorna para valor que foi salvo no \$ra, que nesse caso, era a última instrução, por tanto o programa é finalizado.



Conclusão

Pôde ser visto no presente trabalho a implementação em hardware ao suporte de comandos de desvios incondicionais (Jump, Jal e Jr).

Para isso, foi necessário a adição de novos circuitos empregando lógicas para tratar casos específicos desse comandos.