Trabalho CI210 2011-1

Flávio Zavan (GRR20101924) e Gustavo Toshi Komura (GRR20102342)

```
@@ -25,7 +25,7 @@
   -- tipo para os barramentos de 32 bits

   type inst_type is (ADDU, SUBU, AAND, OOR, XXOR, NNOR, LW, SW,
-   ORI, invalid_instruction);
+   ORI, invalid_instruction, LUI, MUL, ADDIU, J, JAL, JR, BGTZ, BGEZ);

   type microinstruction is record
     ce:    std_logic;       -- ce e rw são os controles da memória
@@ -125,16 +125,24 @@

 architecture alu_arq of alu is
 signal alu_int_alu : reg32;
+signal alu_mul : std_logic_vector(63 downto 0);
 begin
     alu_outalu <= alu_int_alu;
     alu_int_alu <=
         alu_op1 - alu_op2      when  alu_op_alu=SUBU  else
         alu_op1 and alu_op2    when  alu_op_alu=AAND else
-        alu_op1 or  alu_op2    when  alu_op_alu=OOR  or alu_op_alu=ORI else
+        alu_op1 or  alu_op2    when  alu_op_alu=OOR
+            or alu_op_alu=ORI or alu_op_alu=LUI                      else
         alu_op1 xor alu_op2    when  alu_op_alu=XXOR               else
         alu_op1 nor alu_op2    when  alu_op_alu=NNOR              else
+        alu_mul(31 downto 0)      when  alu_op_alu=MUL                     else
+        x"00000001" when alu_op_alu=BGTZ and (alu_op1 > 0) else
+        x"00000000" when alu_op_alu=BGTZ                         else
+        x"0000000" & "000" & (not alu_op1(31)) when alu_op_alu=BGEZ else
         alu_op1 + alu_op2;      --- default é a soma
  alu_zero <= '1' when alu_int_alu=x"00000000" else '0';
+    -- Resultado da multiplicação
+    alu_mul <= alu_op1 * alu_op2 when alu_op_alu=MUL else x"0000000000000000";
 end alu_arq;

 --+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
@@ -163,12 +171,20 @@
 begin
```

```
    DP_instR <= '1' when DP_uins.i=ADDU or DP_uins.i=SUBU or
-DP_uins.i=AAND or DP_uins.i=OOR or DP_uins.i=XXOR or DP_uins.i=NNOR else
+ DP_uins.i=AAND or DP_uins.i=OOR or DP_uins.i=XXOR or DP_uins.i=NNOR
+        -- O Mul do SPIM em si não é R, mas isto deve fazer o truque
+        or DP_uins.i=MUL else
             '0';

    --======  Hardware para a busca de instruções  ============================================

-   DP_incpc <= DP_pc + 4;
+   -- Jump
+   DP_incpc <= "0000" & DP_instruction(25 downto 0) & "00"
+        when DP_uins.i=J or DP_uins.i=JAL else
+        DP_R1 when DP_uins.i=JR else
+        DP_pc + ("00000000000000" & DP_instruction(15 downto 0) & "00") when
+        (DP_uins.i=BGTZ or DP_uins.i=BGEZ) and DP_Result=x"0000001" else
+        DP_pc + 4;

    rpc: entity work.reg32_ce
     generic map(INIT_VALUE=>x"00400000") -- ATENÇÃO a este VALOR!!
@@ -187,6 +203,7 @@
    --========= hardware do banco de registradores e extensão de sinal ou de 0 ===============

    DP_adD <= DP_instruction(15 downto 11) when DP_instR='1'  else
+          "11111" when DP_uins.i=JAL else
           DP_instruction(20 downto 16) ;

    REGS: entity work.reg_bank port map
@@ -195,7 +212,9 @@

    -- Extensão de 0 ou extensão de sinal
    DP_ext32 <= x"FFFF" & DP_instruction(15 downto 0) when (DP_instruction(15)='1'
-   and (DP_uins.i=LW or DP_uins.i=SW)) else
+   and (DP_uins.i=LW or DP_uins.i=SW or DP_uins.i=ADDIU)) else
+           -- ADDIU também extense o sinal e extensão diferente pro LUI
+           DP_instruction(15 downto 0) & x"0000" when DP_uins.i=LUI else
      -- LW and SW use signal extension, ORI uses 0-extension
  x"0000" & DP_instruction(15 downto 0);
     -- other instructions do not use this information,
@@ -214,7 +233,9 @@

    DP_data <= DP_R2 when DP_uins.ce='1' and DP_uins.rw='0' else (others=>'Z');

-   DP_reg_dest <=  DP_data when DP_uins.i=LW else DP_result;
+   DP_reg_dest <=  DP_data when DP_uins.i=LW else
+   DP_pc + 4 when DP_uins.i=JAL else
```

```
+   DP_result;

 end datapath_arq;

@@ -249,6 +270,15 @@
          ORI    when CT_ir(31 downto 26)="001101" else
          LW     when CT_ir(31 downto 26)="100011" else
          SW     when CT_ir(31 downto 26)="101011" else
+         -- Novas daqui pra baixo
+         LUI    when CT_ir(31 downto 26)="001111" else
+         MUL    when CT_ir(31 downto 26)="011100" else
+         ADDIU  when CT_ir(31 downto 26)="001001" else
+         J      when CT_ir(31 downto 26)="000010" else
+         JAL      when CT_ir(31 downto 26)="000011" else
+         JR     when CT_ir(31 downto 26)="000000" and CT_ir(10 downto 0)="00000001000" else
+         BGTZ     when CT_ir(31 downto 26)="000111" and CT_ir(20 downto 16)="00000" else
+         BGEZ     when CT_ir(31 downto 26)="000001" and CT_ir(20 downto 16)="00001" else
          invalid_instruction ; -- IMPORTANTE: condição "default" é invalid instruction;

     assert i /= invalid_instruction
@@ -259,7 +289,8 @@

     CT_uins.rw    <= '0' when i=SW  else '1';

-    CT_uins.wreg  <= '0' when i=SW  else '1';
+    --Não permite escrita nas instruções certas
+    CT_uins.wreg  <= '0' when i=SW or i=J or i=JR or i=BGTZ or i=BGEZ else '1';

 end control_unit;
```