Университет ИТМО

Факультет программной инженерии и компьютерной техники
Кафедра вычислительной техники

Лабораторная работа № 4 по дисциплине "Схемотехника ЭВМ"

Вариант: 2

Выполнил: Айтуганов Д. А.
Чебыкин И. Б.

Группа: P3301

Проверяющий: Баевских А. Н.

СПб, 2016

Содержание

1 Задание

Реализовать выполнение команды JAL, JR

2 Формат команд

31 – 26	25 – 0
JAL 000011	instr_index
6	26

31 – 26	25 – 21	20 – 11	10 – 6	5 – 0
SPECIAL 000000	rs	0 00 0000 0000	hint	JR 001000
6	5	10	5	6

3 Внесенные изменения

```
diff --git a/circuitry/lab4/src/hdl/control.v b/circuitry/lab4/src/hdl/control.v
index 06a370e..806feab 100644
--- a/circuitry/lab4/src/hdl/control.v
+++ b/circuitry/lab4/src/hdl/control.v
@@ -1,7 +1,6 @@
 ‘timescale 1ns / 1ps

 module control( input [5:0]      opcode,
-      input [5:0]      special,
                 input            branch_eq,       // result of comparison for conditional
    branch

                 output reg [1:0] if_pc_source,
@@ -9,8 +8,7 @@ module control( input [5:0]      opcode,

                 output           ex_imm_command,
                 output reg       ex_alu_src_b,
-      output reg       ex_alu_rslt_src, // ? PC + 8 : alu_result => EX_MEM_alu_rslt
-                output reg [1:0] ex_dst_reg_sel,
+                output reg       ex_dst_reg_sel,
                 output reg [1:0] ex_alu_op,
                 output reg       mem_read,
                 output reg       mem_write,
@@ -21,17 +19,14 @@ module control( input [5:0]      opcode,
      localparam      LW    = 6’b100011,
                      SW    = 6’b101011,
                      BEQ   = 6’b000100,
-                     RTYPE = 6’b000000,
-                     J     = 6’b000010,
-                     JAL   = 6’b000011,
-                     ADDI  = 6’b001000,
+                     RTYPE = 6’b0,
+                     J     = 6’d2,
+                     JAL   = 6’d3,
+                     ADDI  = 6’b001000,
                      ANDI  = 6’b001100,
                      ORI   = 6’b001101,
                      XORI  = 6’b001110,
-                     SLTI  = 6’b001010,
-
- //special
-                     JR    = 6’b001000;
+                     SLTI  = 6’b001010;
      //--------------------------------

      reg memory_op;
@@ -45,15 +40,14 @@ module control( input [5:0]      opcode,
      always @* begin
      //default values
-          if_pc_source     = 0;
-          ex_alu_src_b     = 0;
-   ex_alu_rslt_src = 0;
-          ex_dst_reg_sel   = 0;
-          ex_alu_op        = 0;
-          mem_read         = 0;
-          mem_write        = 0;
-          wb_mem_to_reg    = 0;
-          wb_reg_write     = 0;
+          if_pc_source    = 0;
+          ex_alu_src_b    = 0;
+          ex_dst_reg_sel  = 0;
+          ex_alu_op       = 0;
+          mem_read        = 0;
+          mem_write       = 0;
+          wb_mem_to_reg   = 0;
+          wb_reg_write    = 0;

           memory_op    = ( (opcode == LW) | (opcode == SW) );
           r_type_op    = ( opcode == RTYPE );
@@ -63,7 +57,7 @@ module control( input [5:0]      opcode,

           if (memory_op) begin
                ex_alu_src_b   = 1'b1;  // select sign_extend_offset input
-               ex_dst_reg_sel = 2'b00; // rt
+               ex_dst_reg_sel = 1'b0;  // rt
                ex_alu_op      = 2'b00; // add op
                wb_mem_to_reg  = 1'b1;  // select mem_out

@@ -76,7 +70,7 @@ module control( input [5:0]      opcode,
           end
           else if (r_type_op) begin
                ex_alu_src_b = 1'b0;    // select B input
-               ex_dst_reg_sel = 2'b01; // rd
+               ex_dst_reg_sel = 1'b1;  // rd
                ex_alu_op = 2'b10;      // operaction defined by func code

                wb_mem_to_reg = 1'b0;   // alu_out
@@ -84,7 +78,7 @@ module control( input [5:0]      opcode,
           end
           else if (immediate_op) begin
                ex_alu_src_b = 1'b1;    // select sign_extend_offset input
-               ex_dst_reg_sel = 2'b00; // rt
+               ex_dst_reg_sel = 1'b0;  // rt
                ex_alu_op = 2'b10;      // operation defined by function code

                wb_mem_to_reg = 1'b0;   // alu_out
@@ -99,23 +93,9 @@ module control( input [5:0]      opcode,
           end
           else if (jump_op)
                if_pc_source = 2'b10;  // PC <= jump_addr
-     if( opcode == JAL ) begin
-       ex_dst_reg_sel = 2'b10;
-       ex_alu_rslt_src = 1'b1; // EX_MEM_alu_result <= PC + 8
-
-       wb_reg_write  = 1'b1;
-     end
-          else if (~|opcode) begin
-     if(special == JR) begin
-       if_pc_source = 2'b11;
-     end
-     else begin
-       //NOP
-     end
+          else begin
+               //NOP
           end
-   else begin
-     //NOP
-   end
      end

 endmodule
```

```
diff --git a/circuitry/lab4/src/hdl/ex_stage.v b/circuitry/lab4/src/hdl/ex_stage.v
index 5207328..5a7e3d3 100644
--- a/circuitry/lab4/src/hdl/ex_stage.v
+++ b/circuitry/lab4/src/hdl/ex_stage.v
@@ -6,8 +6,7 @@ module ex_stage( input             clk,
                  input             mem_write,
                  input             ex_imm_command,
                  input             ex_alu_src_b,
-                 input             ex_alu_rslt_src,    //
-                 input  [1:0]      ex_dst_reg_sel,
+                 input             ex_dst_reg_sel,
                  input  [1:0]      ex_alu_op,

                  input pstop_i,
@@ -15,7 +14,6 @@ module ex_stage( input             clk,
                  input  [31:0]     A,
                  input  [31:0]     B,
                  input  [31:0]     sign_extend_offset,
-                 input  [31:0]     next_i_addr,    // PC + 8
                  input  [4:0]      rt,                 // target register
                  input  [4:0]      rd,                 // destination register
                  input  [5:0]      opcode,
@@ -24,10 +22,8 @@ module ex_stage( input             clk,
                  input  [31:0]     mem_fwd_val,        // forwarding from EX_MEM
                  input  [31:0]     wb_fwd_val,         // forwarding from WB

+                 output  [4:0]     ex_dst_reg,
                  output  [5:0]     ex_opcode,
-
-                 output reg [31:0] alu_a_in,
-                 output reg [4:0]  ex_dst_reg,

                  output reg [31:0] EX_MEM_alu_result,
                  output reg [31:0] EX_MEM_B_value,
@@ -42,6 +38,7 @@ module ex_stage( input             clk,
      wire [5:0]  func_code;      // func code for ALU control
      wire [4:0]  alu_ctl;        // ALU control lines

+     reg  [31:0] alu_a_in;
      wire [31:0] alu_b_in;
      reg  [31:0] b_value;

@@ -66,19 +63,11 @@ module ex_stage( input             clk,
           endcase
      end

- always @* begin
-          case(ex_dst_reg_sel)
-               0: ex_dst_reg = rt;
-               1: ex_dst_reg = rd;
-               2: ex_dst_reg = 5'd31;
-               default: ex_dst_reg = 5'd0;    //
-          endcase
-     end
-
      assign alu_b_in   = ex_alu_src_b ? sign_extend_offset : b_value;
      assign func_field = sign_extend_offset [5:0]; // looks wierd, but func code is encoded
    there
      assign func_code  = ex_imm_command ? {{2'b10},{~opcode[2] & opcode[1] & ~opcode[0]},
    opcode[2:0]} : func_field;
      assign ex_opcode  = opcode;
+     assign ex_dst_reg = ex_dst_reg_sel ? rd : rt;

      alu_ctrl aluctl_inst(
          .alu_op    (ex_alu_op),
@@ -104,7 +93,7 @@ module ex_stage( input             clk,
                EX_MEM_wb_mem_to_reg <= 0;
          end
          else begin
-               EX_MEM_alu_result <= ex_alu_rslt_src ? next_i_addr : alu_result;
+               EX_MEM_alu_result <= alu_result;
               EX_MEM_B_value <= b_value;
               EX_MEM_dst_reg <= ex_dst_reg;
               EX_MEM_opcode <= opcode;
diff --git a/circuitry/lab4/src/hdl/id_stage.v b/circuitry/lab4/src/hdl/id_stage.v
index 07e49ca..58d3dd6 100644
```

```
--- a/circuitry/lab4/src/hdl/id_stage.v
+++ b/circuitry/lab4/src/hdl/id_stage.v
@@ -34,8 +34,7 @@ module id_stage( input             clk, rst,
                  output reg        ID_EX_mem_write,
                  output reg        ID_EX_ex_imm_command,
                  output reg        ID_EX_ex_alu_src_b,
-                 output reg        ID_EX_ex_alu_rslt_src,
-                 output reg [1:0]  ID_EX_ex_dst_reg_sel,
+                 output reg        ID_EX_ex_dst_reg_sel,
                  output reg [1:0]  ID_EX_ex_alu_op,

                  output [31:0]     branch_addr, jump_addr,    // branch and jump adresses
@@ -48,9 +47,8 @@ module id_stage( input             clk, rst,
      wire [31:0] sign_extend_offset;

      wire        ex_imm_command;
-     wire        ex_alu_src_b;
- wire      ex_alu_rslt_src;
-     wire [1:0]  ex_dst_reg_sel;
+     wire        ex_alu_src_b;
+     wire        ex_dst_reg_sel;
      wire [1:0]  ex_alu_op;
      wire        mem_read;
      wire        mem_write;
@@ -105,13 +103,11 @@ module id_stage( input             clk, rst,

      control cunit_instance (
          .opcode( instruction [31:26] ),
-         .special( instruction [5:0] ),
          .branch_eq( branch_eq ),
          .id_rt_is_source(id_rt_is_source),
          .if_pc_source(if_pc_source),
          .ex_imm_command(ex_imm_command),
          .ex_alu_src_b(ex_alu_src_b),
-         .ex_alu_rslt_src(ex_alu_rslt_src),
          .ex_dst_reg_sel(ex_dst_reg_sel),
          .ex_alu_op(ex_alu_op),
          .mem_read(mem_read),
@@ -136,7 +132,6 @@ module id_stage( input             clk, rst,
               ID_EX_mem_write <= 0;
               ID_EX_ex_imm_command <= 0;
               ID_EX_ex_alu_src_b <= 0;
-     ID_EX_ex_alu_rslt_src <= 0;
               ID_EX_ex_dst_reg_sel <= 0;
               ID_EX_ex_alu_op <= 0;
          end
@@ -156,15 +151,14 @@ module id_stage( input             clk, rst,

               if(!pstop_i) begin
                   if (is_nop || hazard) begin
-                       ID_EX_wb_reg_write    <= 0;
-                       ID_EX_wb_mem_to_reg   <= 0;
-                       ID_EX_mem_read        <= 0;
-                       ID_EX_mem_write       <= 0;
-                       ID_EX_ex_imm_command  <= 0;
-                       ID_EX_ex_alu_src_b    <= 0;
-          ID_EX_ex_alu_rslt_src <= 0;
-                       ID_EX_ex_dst_reg_sel  <= 0;
-                       ID_EX_ex_alu_op       <= 0;
+                       ID_EX_wb_reg_write   <= 0;
+                       ID_EX_wb_mem_to_reg  <= 0;
+                       ID_EX_mem_read       <= 0;
+                       ID_EX_mem_write      <= 0;
+                       ID_EX_ex_imm_command <= 0;
+                       ID_EX_ex_alu_src_b   <= 0;
+                       ID_EX_ex_dst_reg_sel <= 0;
+                       ID_EX_ex_alu_op      <= 0;
                    end
                    else begin
                        ID_EX_wb_reg_write <= wb_reg_write;
@@ -173,7 +167,6 @@ module id_stage( input             clk, rst,
                        ID_EX_mem_write <= mem_write;
                        ID_EX_ex_imm_command <= ex_imm_command;
                        ID_EX_ex_alu_src_b <= ex_alu_src_b;
-          ID_EX_ex_alu_rslt_src <= ex_alu_rslt_src;
                        ID_EX_ex_dst_reg_sel <= ex_dst_reg_sel;
```

```
                         ID_EX_ex_alu_op <= ex_alu_op;
                     end
diff --git a/circuitry/lab4/src/hdl/if_stage.v b/circuitry/lab4/src/hdl/if_stage.v
index c37f303..02f4cd0 100644
--- a/circuitry/lab4/src/hdl/if_stage.v
+++ b/circuitry/lab4/src/hdl/if_stage.v
@@ -15,7 +15,7 @@ module if_stage( input             clk, rst,
                  output     [31:0] i_addr,

                  input      [31:0] i_instr_in,
-                 input      [31:0] jump_addr, branch_addr, reg_data_1,
+                 input      [31:0] jump_addr, branch_addr,

                  output reg [31:0] IF_ID_next_i_addr,
                  output reg [31:0] IF_ID_instruction );
@@ -40,8 +40,7 @@ module if_stage( input             clk, rst,
           case (pc_source)
                2'b00: pc_next = next_i_addr;
                2'b01: pc_next = branch_addr;
-              2'b10: pc_next = jump_addr;
-     2'b11: pc_next = reg_data_1;
+              2'b10: pc_next = jump_addr;
           endcase
      end

diff --git a/circuitry/lab4/src/hdl/pipeline.v b/circuitry/lab4/src/hdl/pipeline.v
index 5762ea0..bd18ddc 100644
--- a/circuitry/lab4/src/hdl/pipeline.v
+++ b/circuitry/lab4/src/hdl/pipeline.v
@@ -33,7 +33,6 @@ module pipeline ( input wire         clk,

      wire [4:0]  ex_dst_reg;
      wire [5:0]  ex_opcode;
- wire [31:0] ex_reg_data_1;    // for jr

      wire [4:0]  id_rs;
      wire [4:0]  id_rt;
@@ -45,8 +44,7 @@ module pipeline ( input wire         clk,
      wire        ID_EX_mem_write;
      wire        ID_EX_ex_imm_command;
      wire        ID_EX_ex_alu_src_b;
-     wire        ID_EX_ex_alu_rslt_src;
-     wire [1:0]  ID_EX_ex_dst_reg_sel;
+     wire        ID_EX_ex_dst_reg_sel;
      wire [1:0]  ID_EX_ex_alu_op;
      wire [31:0] ID_EX_A;
      wire [31:0] ID_EX_B;
@@ -94,7 +92,6 @@ module pipeline ( input wire         clk,
           .i_instr_in        ( i_instr_in),
           .jump_addr         ( jump_addr ),
           .branch_addr       ( branch_addr ),
-   .reg_data_1      ( ex_reg_data_1 ),
           .IF_ID_instruction ( i_fetched ),
           .IF_ID_next_i_addr ( next_i_addr ));

@@ -167,7 +164,6 @@ module pipeline ( input wire         clk,
           .ID_EX_mem_write ( ID_EX_mem_write ),
           .ID_EX_ex_imm_command ( ID_EX_ex_imm_command ),
           .ID_EX_ex_alu_src_b ( ID_EX_ex_alu_src_b ),
-          .ID_EX_ex_alu_rslt_src ( ID_EX_ex_alu_rslt_src ),
           .ID_EX_ex_dst_reg_sel ( ID_EX_ex_dst_reg_sel ),
           .ID_EX_ex_alu_op ( ID_EX_ex_alu_op ),

@@ -188,13 +184,11 @@ module pipeline ( input wire         clk,
           .mem_write ( ID_EX_mem_write ),
           .ex_imm_command ( ID_EX_ex_imm_command ),
           .ex_alu_src_b ( ID_EX_ex_alu_src_b ),
-          .ex_alu_rslt_src ( ID_EX_ex_alu_rslt_src ),
           .ex_dst_reg_sel ( ID_EX_ex_dst_reg_sel ),
           .ex_alu_op ( ID_EX_ex_alu_op ),
           .A ( ID_EX_A ),
           .B ( ID_EX_B ),
           .sign_extend_offset ( ID_EX_sign_extend_offset ),
-   .next_i_addr ( next_i_addr ),     // execute: PC + 8
           .rt ( ID_EX_rt ),                             // target register
           .rd ( ID_EX_rd ),                             // destination register
           .opcode ( ID_EX_opcode ),
@@ -203,7 +197,6 @@ module pipeline ( input wire         clk,
           .mem_fwd_val ( EX_MEM_alu_result ),               // forwarding from MEM
           .wb_fwd_val ( wreg_data ),                        // forwarding from WB
           .ex_dst_reg ( ex_dst_reg ),
-    .alu_a_in ( ex_reg_data_1 ),
           .ex_opcode ( ex_opcode ),
           .EX_MEM_alu_result ( EX_MEM_alu_result ),
           .EX_MEM_B_value ( EX_MEM_B_value ),
```

Листинг 1: diff