## riscv_id_stage

```
  input  clk,
  input  rst_n,

  input  test_en_i,

  input  fetch_enable_i,
  output  ctrl_busy_o,
  output  is_decoding_o,

  // Interface to IF stage
  input  [N_HWLP-1:0] hwlp_dec_cnt_i,
  input  is_hwlp_i,
  input  instr_valid_i,
  input  [31:0] instr_rdata_i,
  output  instr_req_o,
  output  [31:0] jump_target_o,

  // From ID to IF stage signals
  output  clear_instr_valid_o,
  output  pc_set_o,
  output  [2:0]  pc_mux_o,
  output  [1:0]  exc_pc_mux_o,
  output  [4:0]  exc_vec_pc_mux_o,

  // From IF to ID
  input  illegal_c_insn_i,
  input  is_compressed_i,
  input  [31:0] pc_if_i,
  input  [31:0] pc_id_i,

  // Stalls
  output  halt_if_o,      // to IF stage
  output  id_ready_o,     // to IF stage
  input  if_ready_i,
  input  if_valid_i,


// hwloop signals from CS register
input  [N_HWLP_BITS-1:0] csr_hwlp_regid_i,
input  [2:0] csr_hwlp_we_i,
input  [31:0] csr_hwlp_data_i,

// Interface to load store unit
output  data_req_ex_o,
output  data_we_ex_o,
output  [1:0]  data_type_ex_o,
output  data_sign_ext_ex_o,
output  [1:0]  data_reg_offset_ex_o,
output  data_load_event_ex_o,
output  data_misaligned_ex_o,
output  prepost_useincr_ex_o,
input  data_misaligned_i,

// Interrupt signals
input  [31:0] irq_i,
input  irq_enable_i,
output  [5:0]  exc_cause_o,
output  save_exc_cause_o,
output  exc_save_if_o,
output  exc_save_id_o,
output  exc_restore_id_o,
input  lsu_load_err_i,
input  lsu_store_err_i,

// Forward Signals
input  [4:0]  regfile_waddr_wb_i,
input  regfile_we_wb_i,
input  [31:0] regfile_wdata_wb_i,
input  [4:0]  regfile_alu_waddr_fw_i,
input  regfile_alu_we_fw_i,
input  [31:0] regfile_alu_wdata_fw_i,

// from ALU
input  mult_multicycle_i,

// Jumps and branches
output  branch_in_ex_o, //To ex stage & CSR
input  branch_decision_i, //From EX stage

input  ex_ready_i,     // from EX stage

output  id_valid_o,     // To CSR

input  ex_valid_i,
input  wb_valid_i,

// Pipeline ID/EX
output  [31:0] pc_ex_o,
+
output  [31:0] alu_operand_a_ex_o,
output  [31:0] alu_operand_b_ex_o,
output  [31:0] alu_operand_c_ex_o,
output  [ 4:0] bmask_a_ex_o,
output  [ 4:0] bmask_b_ex_o,
output  [ 1:0] imm_vec_ext_ex_o,
output  [ 1:0] alu_vec_mode_ex_o,
+
output  [4:0]  regfile_waddr_ex_o,
output  regfile_we_ex_o,
+
output  [4:0]  regfile_alu_waddr_ex_o,
output  regfile_alu_we_ex_o,

// ALU
output  [ALU_OP_WIDTH-1:0] alu_operator_ex_o,

// MUL
output  [ 2:0] mult_operator_ex_o,
output  [31:0] mult_operand_a_ex_o,
output  [31:0] mult_operand_b_ex_o,
output  [31:0] mult_operand_c_ex_o,
output  mult_en_ex_o,
output  mult_sel_subword_ex_o,
output  [ 1:0] mult_signed_mode_ex_o,
output  [ 4:0] mult_imm_ex_o,

output  [31:0] mult_dot_op_a_ex_o,
output  [31:0] mult_dot_op_b_ex_o,
output  [31:0] mult_dot_op_c_ex_o,
output  [ 1:0] mult_dot_signed_ex_o,

// CSR ID/EX
output  csr_access_ex_o,
output  [1:0]  csr_op_ex_o,

// hwloop signals
output  [N_HWLP-1:0] [31:0] hwlp_start_o,
output  [N_HWLP-1:0] [31:0] hwlp_end_o,
output  [N_HWLP-1:0] [31:0] hwlp_cnt_o,

// Debug Unit Signals
input  [DBG_SETS_W-1:0] dbg_settings_i,
input  dbg_req_i,
output  dbg_ack_o,
input  dbg_stall_i,
output  dbg_trap_o,
input  dbg_reg_rreq_i,
input  [ 4:0] dbg_reg_raddr_i,
output  [31:0] dbg_reg_rdata_o,
input  dbg_reg_wreq_i,
input  [ 4:0] dbg_reg_waddr_i,
input  [31:0] dbg_reg_wdata_i,
input  dbg_jump_req_i,

// Performance Counters
output  perf_jump_o,
output  perf_jr_stall_o,
output  perf_ld_stall_o
```


riscv_register_file
cluster_clock_gating
riscv_decoder
riscv_controller
riscv_exc_controller
riscv_hwloop_regs

riscv_register_file

```
input  test_en_i,
//Read port R1
input  [ADDR_WIDTH-1:0]  raddr_a_i,
//Read port R2
input  [ADDR_WIDTH-1:0]  raddr_b_i,
//Read port R3
input  [ADDR_WIDTH-1:0]  raddr_c_i,
// Write port W1
input  [ADDR_WIDTH-1:0]   waddr_a_i,
input  [DATA_WIDTH-1:0]   wdata_a_i,
input  we_a_i,
// Write port W2
input  [ADDR_WIDTH-1:0]   waddr_b_i,
input  [DATA_WIDTH-1:0]   wdata_b_i,
input  we_b_i

output  [DATA_WIDTH-1:0]  rdata_a_o,
output  [DATA_WIDTH-1:0]  rdata_b_o,
output  [DATA_WIDTH-1:0]  rdata_c_o,
```

riscv_decoder

```
// singals running to/from controller
input  deassert_we_i,
input  data_misaligned_i,
input  mult_multicycle_i,
// from IF/ID pipeline
input  [31:0] instr_rdata_i,
input  illegal_c_insn_i,

-----------------------------------------------------------
// register file related signals
output  regfile_mem_we_o,
output  regfile_alu_we_o,
output  regfile_alu_waddr_sel_o,
// CSR manipulation
output  csr_access_o,
output  [1:0]  csr_op_o,
// LD/ST unit signals
output  data_req_o,
output  data_we_o,
output  prepost_useincr_o,
output  [1:0]  data_type_o,
output  data_sign_extension_o,
output  [1:0]  data_reg_offset_o,
output  data_load_event_o,
// hwloop signals
output  [2:0]  hwloop_we_o,
output  hwloop_target_mux_sel_o,
output  hwloop_start_mux_sel_o,
output  hwloop_cnt_mux_sel_o,
// jump/branches
output  [1:0]  jump_in_dec_o,
output  [1:0]  jump_in_id_o,
output  [1:0]  jump_target_mux_sel_o

// singals running to/from controller
output  illegal_insn_o,
output  ebrk_insn_o,
output  eret_insn_o,
output  ecall_insn_o,
output  pipe_flush_o,
output  rega_used_o,
output  regb_used_o,
output  regc_used_o,
output  bmask_needed_o,
output  [ 0:0] bmask_a_mux_o,
output  [ 1:0] bmask_b_mux_o,
// ALU signals
output  [ALU_OP_WIDTH-1:0] alu_operator_o,
output  [1:0]  alu_op_a_mux_sel_o,
output  [1:0]  alu_op_b_mux_sel_o,
output  [1:0]  alu_op_c_mux_sel_o,
output  [1:0]  alu_vec_mode_o,
output  scalar_replication_o,
output  [0:0]  imm_a_mux_sel_o,
output  [3:0]  imm_b_mux_sel_o,
output  [1:0]  regc_mux_o,
// MUL related control signals
output  [2:0]  mult_operator_o,
output  mult_int_en_o,
output  mult_dot_en_o,
output  [0:0]  mult_imm_mux_o,
output  mult_sel_subword_o,
output  [1:0]  mult_signed_mode_o,
output  [1:0]  mult_dot_signed_o,
```

riscv_controller

riscv_exc_controller

riscv_hwloop_regs

riscv_register_file

riscv_decoder

riscv_controller

```
input  fetch_enable_i,
// decoder related signals
input  illegal_insn_i,
input  eret_insn_i,
input  pipe_flush_i,
input  rega_used_i,
input  regb_used_i,
input  regc_used_i,
// from IF/ID pipeline
input  instr_valid_i,
input  [31:0] instr_rdata_i,
// LSU
input  data_req_ex_i,
input  data_misaligned_i,
input  data_load_event_i,
// from ALU
input  mult_multicycle_i,
// jump/branch signals
input  branch_taken_ex_i,
input  [1:0]  jump_in_id_i,
input  [1:0]  jump_in_dec_i,
// Exception Controller Signals
input  exc_req_i,
// Debug Signals
input  dbg_req_i,
input  dbg_stall_i,
input  dbg_jump_req_i,
// Forwarding signals from regfile
input  [4:0]  regfile_waddr_ex_i,
input  regfile_we_ex_i,
input  [4:0]  regfile_waddr_wb_i,
input  regfile_we_wb_i,
input  [4:0]  regfile_alu_waddr_fw_i,
input  regfile_alu_we_fw_i,
// forwarding detection signals
input  reg_d_ex_is_reg_a_i,
input  reg_d_ex_is_reg_b_i,
input  reg_d_ex_is_reg_c_i,
input  reg_d_wb_is_reg_a_i,
input  reg_d_wb_is_reg_b_i,
input  reg_d_wb_is_reg_c_i,
input  reg_d_alu_is_reg_a_i,
input  reg_d_alu_is_reg_b_i,
input  reg_d_alu_is_reg_c_i,
// stall signals
input  id_ready_i,
input  if_valid_i,
input  ex_valid_i,
input  wb_valid_i,

output  ctrl_busy_o,
output  is_decoding_o,
// decoder related signals
output  deassert_we_o,
// from prefetcher
output  instr_req_o,
// to prefetcher
output  pc_set_o,
output  [2:0]  pc_mux_o,
// Exception Controller Signals
output  exc_ack_o,
output  exc_save_if_o,
output  exc_save_id_o,
output  exc_restore_id_o,
// Debug Signals
output  dbg_ack_o,
// forwarding signals
output  [1:0]  operand_a_fw_mux_sel_o,
output  [1:0]  operand_b_fw_mux_sel_o,
output  [1:0]  operand_c_fw_mux_sel_o,
// stall signals
output  halt_if_o,
output  halt_id_o,
output  misaligned_stall_o,
output  jr_stall_o,
output  load_stall_o,
// Performance Counters
output  perf_jump_o,
output  perf_jr_stall_o,
output  perf_ld_stall_o
```

riscv_exc_controller

riscv_hwloop_regs

riscv_register_file

riscv_decoder

riscv_controller

riscv_exc_controller

```
// handshake signals to controller
input  ack_i,
// interrupt lines
input  [31:0] irq_i,
input  irq_enable_i,
// from decoder
input  ebrk_insn_i,
input  illegal_insn_i,
input  ecall_insn_i,
input  eret_insn_i,
input  lsu_load_err_i,
input  lsu_store_err_i,
// from debug unit
input  [DBG_SETS_W-1:0] dbg_settings_i
```

```
// handshake signals to controller
output  req_o,

output  trap_o,
// to IF stage
output  [1:0] pc_mux_o,
output  [4:0] vec_pc_mux_o,


// to CSR
output  [5:0]  cause_o,
output  save_cause_o,
```

riscv_hwloop_regs

```
// from ex stage
input  [31:0]    hwlp_start_data_i,
input  [31:0]    hwlp_end_data_i,
input  [31:0]    hwlp_cnt_data_i,
input  [2:0]    hwlp_we_i,
input  [N_REG_BITS-1:0]
hwlp_regid_i,

// from controller
input  valid_i,

// from hwloop controller
input  [N_REGS-1:0]
hwlp_dec_cnt_i,
```

```
// to hwloop controller
output  [N_REGS-1:0] [31:0]
hwlp_start_addr_o,
output  [N_REGS-1:0] [31:0]
hwlp_end_addr_o,
output  [N_REGS-1:0] [31:0]
hwlp_counter_o
```