File: C:/my_designs/Pipeline_Test/Pipeline/src/Pipeline.vhd

```
1   ------------------------------------------------------------------------
    ----
2   --
3   -- Title       : Pipeline
4   -- Design      : Pipeline
5   -- Author      : Aaron Lin and Hang Chen
6   -- Company     : Stony Brook University
7   --
8   ------------------------------------------------------------------------
    ----
9   --
10  -- File        : c:\my_designs\Pipeline\Pipeline\src\Pipeline.vhd
11  -- Generated   : Fri May  1 15:23:57 2020
12  -- From        : interface description file
13  -- By          : Itf2Vhdl ver. 1.22
14  --
15  ------------------------------------------------------------------------
    ----
16  --
17  -- Description : This is the top level design of the pipeline.
18  --
19  ------------------------------------------------------------------------
    ----
20  library work;
21  use work.all;
22  library IEEE;
23  use IEEE.std_logic_1164.all;
24  use IEEE.NUMERIC_STD.all;
25
26  entity Pipeline is
27       port(
28       ALU_Output : out std_logic_vector(127 downto 0);
29       Input_rs3 : out std_logic_vector(127 downto 0);
30       Input_rs2 : out std_logic_vector(127 downto 0);
31       Input_rs1 : out std_logic_vector(127 downto 0);
32       W_En_inWB : out std_logic;
33       Forward_Mux1 : out std_logic;
34       Forward_Mux2 : out std_logic;
35       Forward_Mux3 : out std_logic;
36       Instruction_inIF : out std_logic_vector(24 downto 0);
37       Instruction_inID : out std_logic_vector(24 downto 0);
38       ALUop_inID : out std_logic_vector(7 downto 0);
39       ALUop_inEX : out std_logic_vector(7 downto 0);
40       Data_inWB : out std_logic_vector(127 downto 0);
41
42       Instruction_Index : in std_logic_vector(5 downto 0);
43       Instruction : in std_logic_vector(24 downto 0);
44       CLK : in STD_LOGIC
45       );
46  end Pipeline;
47
48  --}} End of automatically maintained section
49
50  architecture Pipeline of Pipeline is
51  signal Instruction_toReg, Instruction_toIF : std_logic_vector(24 downto 0);

52  signal W_EN_toIDReg, W_EN_toEXReg, W_EN_toForwarder, forward_Selector1,
    forward_Selector2, forward_Selector3 : std_logic;
```

```
53  signal regA_Data, regB_Data, regC_Data, ALU_regA, ALU_regB, ALU_regC,
    Res_toEXReg, forward_Data: std_logic_vector(127 downto 0);
54  signal regA_toIDReg, regB_toIDReg, regC_toIDReg, regD_toEXReg, regD_toIDReg
    , regD_WB, regA_Forward, regB_Forward, regC_Forward : std_logic_vector(4
    downto 0);
55  signal data_WB : std_logic_vector(127 downto 0);
56  signal ALUop_toIDReg, ALUop_toEX : std_logic_vector(19 downto 0);
57  begin
58      ALU_Output <= Res_toEXReg;
59      Input_rs3 <= ALU_regC;
60      Input_rs2 <= ALU_regB;
61      Input_rs1 <= ALU_regA;
62      W_En_inWB <= W_EN_toForwarder;
63      Forward_Mux1 <= forward_Selector1;
64      Forward_Mux2 <= forward_Selector2;
65      Forward_Mux3 <= forward_Selector3;
66      Instruction_inIF <= Instruction_toIF;
67      Instruction_inID <= Instruction_toReg;
68      ALUop_inID <= ALUop_toIDReg(17 downto 10);
69      ALUop_inEX <= ALUop_toEx(17 downto 10);
70      Data_inWB <= data_WB;
71
72
73      u0 : entity IF_Stage port map (CLK => CLK, Instruction =>
    Instruction_toIF, Instruction_Data => Instruction, Instruction_Index =>
    Instruction_Index);
74      u1 : entity IF_ID port map (CLK => CLK, I_Instruction =>
    Instruction_toIF, O_Instruction => Instruction_toReg);
75
76      u2 : entity ID_Stage port map (CLK => CLK, Instruction =>
    Instruction_toReg, I_W_EN => W_EN_toForwarder, I_DataIn => data_WB, I_regD
    => regD_WB,
77          O_ALUop => ALUop_toIDReg, O_regD => regD_toIDReg, O_W_EN =>
    W_EN_toIDReg, O_regA_Data => regA_Data, O_regB_Data => regB_Data,
78          O_regC_Data => regC_Data, O_regA => regA_toIDReg, O_regB =>
    regB_toIDReg, O_regC => regC_toIDReg);
79
80
81      u3 : entity ID_EX port map (CLK => CLK, I_W_EN => W_EN_toIDReg, I_regA
    => regA_toIDReg,
82          I_regB => regB_toIDReg, I_regC => regB_toIDReg, I_regD =>
    regD_toIDReg,
83          I_ALUop => ALUop_toIDReg, I_regA_Data => regA_Data, I_regB_Data =>
    regB_Data, I_regC_Data => regC_Data, O_W_EN => W_EN_toEXReg,
84          O_regA => regA_Forward, O_regB => regB_Forward, O_regC =>
    regC_Forward, O_regD => regD_toEXReg, O_ALUop => ALUop_toEX,
85          O_regA_Data => ALU_regA, O_regB_Data => ALU_regB, O_regC_Data =>
    ALU_regC);
86
87      u4 : entity EX port map (I_regA => ALU_regA, I_regB => ALU_regB, I_regC
    => ALU_regC, ALUop => ALUop_toEX, Res => Res_toEXReg,
88          I_forward => forward_Data, Mux1_Selector => forward_Selector1,
    Mux2_Selector => forward_Selector2, Mux3_Selector => forward_Selector3);
89
90      u5 : entity EX_WB port map (CLK => CLK, I_W_EN => W_EN_toEXReg, I_Res
    => Res_toEXReg, I_regD => regD_toEXReg, O_W_EN => W_EN_toForwarder,
91          O_Res => data_WB, O_regD => regD_WB);
```

```
92
93      u6 : entity Forwarder port map (W_EN => W_EN_toForwarder, regA =>
    regA_Forward, regB => regB_Forward,
94          regC => regC_Forward, regD => regD_WB,
95          Mux1_Selector => forward_Selector1, Mux2_Selector =>
    forward_Selector2, Mux3_Selector => forward_Selector3);
96  end Pipeline;
97
```