```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   ENTITY cpu1 is
5   PORT
6      (
7         -- Input ports
8         clk      : in  std_logic;
9         mem_clk  : in  std_logic;
10        rst      : in  std_logic;
11        dataIn   : in  std_logic_vector(31 downto 0);
12        -- Output ports
13        dataOut     : out std_logic_vector(31 downto 0);
14        addrOut     : out std_logic_vector(31 downto 0);
15        wEn         : out std_logic;
16        -- Debug data.
17        dOutA, dOutB   : out std_logic_vector(31 downto 0);
18        dOutC, dOutZ   : out std_logic;
19        dOutIR         : out std_logic_vector(31 downto 0);
20        dOutPC         : out std_logic_vector(31 downto 0);
21        outT           : out std_logic_vector(2 downto 0);
22        wen_mem, en_mem : out std_logic
23     );
24  END cpu1;
25
26  ARCHITECTURE behavior OF cpu1 IS
27
28     COMPONENT reset_circuit
29     PORT
30        (
31           Reset : IN STD_LOGIC;
32           Clk : IN STD_LOGIC;
33           Enable_PD : OUT STD_LOGIC;
34           Clr_PC : OUT STD_LOGIC
35        );
36     END COMPONENT;
37
38     COMPONENT control
39     PORT
40        (
41           clk, mclk: IN STD_LOGIC;
42           enable: IN STD_LOGIC;
43           statusC, statusZ: IN STD_LOGIC;
44           INST: IN STD_LOGIC_VECTOR(31 DOWNTO 0);
45           A_Mux, B_Mux: OUT STD_LOGIC;
46           IM_MUX1, REG_Mux: OUT STD_LOGIC;
47           IM_MUX2, DATA_Mux: OUT STD_LOGIC_VECTOR(1 DOWNTO 0);
48           ALU_op: OUT STD_LOGIC_VECTOR(2 DOWNTO 0);
49           inc_PC, ld_PC: OUT STD_LOGIC;
50           clr_IR: OUT STD_LOGIC;
51           ld_IR: OUT STD_LOGIC;
52           clr_A, clr_B, clr_C, clr_Z: OUT STD_LOGIC;
53           ld_A, ld_B, ld_C, ld_Z: OUT STD_LOGIC;
54           T: OUT STD_LOGIC_VECTOR(2 DOWNTO 0);
55           wen, en: OUT STD_LOGIC
56        );
57     end COMPONENT;
58
59     COMPONENT datapath
60     PORT
61        (
62           Clk,mClk: IN STD_LOGIC; -- clock Signal
63           --Memory Signals
64           WEN, EN : IN STD_LOGIC;
65           -- Register Control Signals (CLR and LD).
66           Clr_A , Ld_A : IN STD_LOGIC;
67           Clr_B , Ld_B : IN STD_LOGIC;
68           Clr_C , Ld_C : IN STD_LOGIC;
```

```
69            Clr_Z , Ld_Z : IN STD_LOGIC;
70            Clr PC , Ld PC : IN STD LOGIC;
71            Clr IR , Ld IR : IN STD LOGIC;
72
73            -- Register outputs (Some needed to feed back to control unit. Others pulled out
    for testing.
74            Out_A : OUT STD_LOGIC_VECTOR(31 DOWNTO 0);
75            Out B : OUT STD LOGIC VECTOR(31 DOWNTO 0);
76            Out C : OUT STD LOGIC;
77            Out_Z : OUT STD_LOGIC;
78            Out PC : OUT STD LOGIC VECTOR(31 DOWNTO 0);
79            Out IR : OUT STD LOGIC VECTOR(31 DOWNTO 0);
80
81            -- Special inputs to PC.
82            Inc PC : IN STD LOGIC;
83
84            -- Address and Data Bus signals for debugging.
85            ADDR OUT : OUT STD LOGIC VECTOR(31 DOWNTO 0);
86            DATA_IN : IN STD_LOGIC_VECTOR(31 DOWNTO 0);
87            DATA OUT: OUT STD LOGIC VECTOR(31 DOWNTO 0);--, MEM OUT, MEM IN
88       --     MEM ADDR : OUT STD LOGIC VECTOR(7 DOWNTO 0);
89
90            -- Various MUX controls.
91            DATA MUX: IN STD LOGIC VECTOR(1 DOWNTO 0);
92            Reg_Mux : IN STD_LOGIC;
93            A Mux, B Mux : IN STD LOGIC;
94            IM MUX1 : IN STD LOGIC;
95            IM_MUX2 : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
96
97            -- ALU Operations.
98            ALU_OP : IN STD_LOGIC_VECTOR(2 DOWNTO 0)
99
100        );
101     END COMPONENT;
102  -- Internal signals--------
103  signal          Clr A1 , Ld A1              :  STD LOGIC;
104  signal          Clr_B1 , Ld_B1              :  STD_LOGIC;
105  signal          Clr C1 , Ld C1              :  STD LOGIC;
106  signal          Clr Z1 , Ld Z1              :  STD LOGIC;
107  signal          Clr_PC1 , Ld_PC1,Inc_PC1    :  STD_LOGIC;
108  signal          Clr IR1 , Ld IR1         :  STD LOGIC;
109  signal          WEN1, EN1                :  STD LOGIC;
110  signal          A_Mux1,B_Mux1            :  STD_LOGIC;
111  signal          IM MUX11, REG Mux1          :  STD LOGIC;
112  signal          IM MUX21, DATA Mux1         :  STD LOGIC VECTOR(1 DOWNTO 0);
113  signal          ALU_OP1                  :  STD_LOGIC_VECTOR(2 DOWNTO 0);
114  signal          enable1                  :  STD LOGIC;
115  signal          statusC1,statusZ1        :  STD LOGIC;
116  signal          Out_C1,Out_Z1: STD_LOGIC;
117  signal          useless1,useless2,useless3,useless4,useless5 :  STD LOGIC VECTOR(31 DOWNTO 0
     );
118  signal          us7                   :  STD_LOGIC_VECTOR(7 DOWNTO 0);
119
120     Begin
121
122     Dpth:datapath
123     PORT MAP
124        (   -- clock Signal
125           Clk=>clk,
126           mClk=>mem clk,
127           --Memory Signals
128           WEN=>WEN1,
129           EN=>EN1,
130           -- Register Control Signals (CLR and LD).
131           Clr A=>CLR A1,
132           Ld A=>Ld A1,
133           Clr_B=>Clr_B1 ,
134           Ld_B=>Ld_B1,
```

```
135             Clr_C=>Clr_C1 ,
136             Ld C=>Ld C1,
137             Clr Z=>Clr Z1 ,
138             Ld_Z=>Ld_Z1,
139             Clr PC=>Clr PC1,
140             Ld PC=>Ld PC1,
141             Clr_IR=>Clr_IR1 ,
142             Ld IR =>Ld IR1,
143             -- Register outputs (Some needed to feed back to control unit. Others pulled out
    for testing.
144             Out A=>dOutA,
145             Out B=>dOutB,
146             Out_C=>Out_C1,
147             Out Z=>Out Z1,
148             Out PC=>dOutPC,
149             Out_IR=>dOutIR,
150             -- Special inputs to PC.
151             Inc PC=>Inc PC1,
152             -- Address and Data Bus signals for debugging.
153             ADDR OUT=>addrOut,
154              DATA IN=>dataIn,
155             DATA_OUT=>dataOut,
156             -- Various MUX controls.
157             DATA MUX=>DATA MUX1,
158             Reg_Mux=>Reg_Mux1,
159             A Mux=>A Mux1,
160             B Mux=>B Mux1,
161             IM_MUX1=>IM_MUX11,
162             IM MUX2=>IM MUX21,
163             -- ALU Operations.
164             ALU_OP=>ALU_OP1
165         );
166
167     C1:control
168     PORT MAP
169         (
170             enable=>enable1,
171             statusC=>Out C1,
172             statusZ=>Out Z1,
173             INST=>dataIn,
174             A Mux=>A Mux1,
175             B Mux=>B Mux1,
176             IM_MUX1=>IM_MUX11,
177             REG Mux=>REG Mux1,
178             IM MUX2=>IM MUX21,
179             DATA_Mux=> DATA_Mux1,
180             ALU OP=>ALU OP1,
181             INC PC=>INC PC1,
182             LD_PC=>  LD_PC1,
183             CLR IR=>CLR IR1,
184             LD IR=>  LD IR1,
185             CLR_A=>CLR_A1,
186             CLR B=>CLR B1,
187             CLR C=>CLR C1,
188             CLR_Z=>CLR_Z1,
189             LD A=>LD A1,
190             LD B=>LD B1,
191             LD_C=>LD_C1,
192             LD Z=>LD Z1,
193             T=>outT,
194             wen=>WEN1,
195             en=>EN1,
196             clk=>clk,
197             mclk=>mem_clk
198         );
199
200     R1:reset_circuit
201     PORT MAP
```

```
202         (
203            Reset=>rst,
204            Clk=>clk,
205            Enable_PD=>enable1,
206            Clr_PC=>Clr_PC1
207         );
208         --Final assignments
209         dOutC<=Out_C1;
210         dOutZ<=Out_Z1;
211         wen_mem<=WEN1;
212         en_mem<=EN1;
213         wEn <='0';
214
215   END behavior;
```