```vhdl
-- Joe Leveille & Jon Bayert
-- Final VHDL Assignment - ECE 424 - Computer Architecture
--

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
use work.mips_types.all;

entity datapath is
    PORT ( pc : IN STD_LOGIC_VECTOR ( 31 downto 0 );
           nextPC: OUT STD_LOGIC_VECTOR (31 downto 0) );

END datapath;

architecture dataBoi of datapath is
    COMPONENT alu_control
        port( INSTR: in std_logic_vector(5 DOWNTO 0);
            ALUOP: in std_logic_vector(1 DOWNTO 0);
            ALUCTRL : OUT std_logic_vector(3 DOWNTO 0));
    END COMPONENT;

    COMPONENT control
        port( INSTR: in std_logic_vector(5 DOWNTO 0);
             REGDST: OUT std_logic;
             BRANCH: OUT std_logic;
             MEMREAD: OUT std_logic;
             MEMTOREG: OUT std_logic;
             ALUOP: OUT std_logic_vector(1 DOWNTO 0);
             MEMWRITE: OUT std_logic;
             ALUSRC: OUT std_logic;
            REGWRITE: OUT std_logic);
    END COMPONENT;

    COMPONENT dmemory
       port(
          DIN: in mips_data;
          ADDR: in mips_address;
          DOUT: out mips_data;
          WE, RE: in std_logic;
          CLK: in std_logic
        );
    END COMPONENT;

    COMPONENT imemory
       port(
          ADDR: in mips_address;
          DOUT: out mips_data
        );
    END COMPONENT;

    COMPONENT alu32
        PORT( A, B: in std_logic_vector (31 downto 0);
            ALUOp: in std_logic_vector (3 downto 0);
            RESULT: out std_logic_vector (31 downto 0);
            Z, V, C: out std_logic );
    END COMPONENT;

    COMPONENT mux2a
       port( IN0: in mips_reg_addr;
          IN1: in mips_reg_addr;
          SEL: in STD_LOGIC;
          DOUT: out mips_reg_addr);
    end component;

    COMPONENT reg_file
      port (a1 : in mips_reg_addr;
            q1 : out mips_data;
```

```
70              a2 : in mips_reg_addr;
71              q2 : out mips_data;
72              a3 : in mips_reg_addr;
73              d3 : in mips_data;
74              write_en : in std_logic;
75              clk : in std_logic);
76      end COMPONENT;
77
78      COMPONENT signextend
79         port( INDATA: in STD_LOGIC_VECTOR(15 DOWNTO 0);
80            OUTDATA: out mips_data);
81      end COMPONENT;
82
83      COMPONENT mux2
84         port( IN0: in mips_data;
85            IN1: in mips_data;
86            SEL: in std_logic;
87            DOUT: out mips_data);
88      end COMPONENT;
89
90      COMPONENT shiftleft
91         port( INDATA: in mips_data;
92            OUTDATA: out mips_data);
93      end COMPONENT;
94
95      SIGNAL sys_clock : std_logic := '0';
96      CONSTANT Tcycle : time := 100 ns;
97
98      SIGNAL inst, ALUResult, outMuxIn1, branchALUin1, wDataMuxin1 : std_logic_vector( 31
        downto 0);
99      SIGNAL defNextPC, WriteData, aluIn0, aluIn1, mux2in0, mux2in1  : std_logic_vector(
        31 downto 0);
100     SIGNAL REGDST, BRANCH, MEMREAD, MEMTOREG, MEMWRITE, ALUSRC, REGWRITE, Zero,
        outMuxSel : std_logic;
101     SIGNAL ALUOpDatapath : std_logic_vector(1 downto 0);
102     SIGNAL wRegIn : std_logic_vector(4 downto 0);
103     SIGNAL mainALUctl : std_logic_vector(3 downto 0);
104
105     BEGIN
106
107       -- create clock process
108       clk_gen: process
109       begin
110         sys_clock <= '1' after Tcycle/3, '0' after Tcycle;
111         wait until sys_clock = '0';
112       end process clk_gen;
113
114     Imem: imemory PORT MAP(
115         ADDR => pc, DOUT => inst
116     );
117
118     PCplus4: alu32 PORT MAP(
119         A => pc, B=> X"00000004", ALUOp => "0010", RESULT=>defNextPC, Z => oPeN, V=>
            OPen, C=>oPEn
120     );
121
122     CTL: control PORT MAP(
123         INSTR => inst(31 downto 26), REGDST => RegDst,BRANCH => Branch,
124         MEMREAD => MEMREAD,MEMTOREG => MEMTOREG,MEMWRITE => MEMWRITE,ALUSRC => ALUSRC,
125         REGWRITE => REGWRITE,ALUOP => ALUOPDatapath
126     );
127
128     writeRegSel: mux2a PORT MAP(
129         IN0 => inst(20 downto 16), IN1 => inst(15 downto 11), SEL => REGDST, DOUT =>
            wRegIn
130     );
131
132     reg: reg_file PORT MAP(
133         a1=> inst(25 downto 21), a2 => inst(20 downto 16), a3 => wRegIn,
```

```
134         q1 => aluIn0, q2 => mux2in0, d3=> WriteData, write_en=> REGWRITE, CLK =>
            sys_clock
135     );
136
137     sign: signextend PORT MAP(
138         INDATA => inst(15 downto 0), OUTDATA => mux2in1
139     );
140
141     muxTheSecond: mux2 PORT MAP(
142         IN0=>mux2in0, IN1=> mux2in1, SEL=>ALUSRC, DOUT => aluIn1
143     );
144
145     ALUctl: alu_control PORT MAP(
146         INSTR=> inst(5 downto 0), ALUop => ALUOpDatapath, ALUctrl => mainALUctl
147     );
148
149     mainALU: alu32 PORT MAP(
150         a=>aluIn0, b=>aluIn1, ALUOp=> mainALUctl, RESULT => ALUResult, Z=>Zero,
            V=>open, C=>oPEN
151     );
152
153     shift: shiftleft PORT MAP(
154         INDATA=>mux2in1, OUTDATA=>branchALUin1
155     );
156
157     branchALU: alu32 PORT MAP(
158         a=>defNextPC, b=>branchALUin1, ALUOP=>"0010", RESULT=>outMuxIn1, Z=>open,
            C=>opEn, V=>OpEN
159     );
160
161     outMuxSel <= branch and Zero;
162
163     outMux: mux2 PORT MAP(
164         IN0=>defNextPC,IN1=>outMuxIn1, SEL=> outMuxSel, DOUT => nextPC
165     );
166
167     dmem: dmemory PORT MAP(
168         ADDR => ALUResult, DIN => mux2in0, DOUT => wDataMuxin1, CLK => sys_clock, WE =>
            memWrite, RE =>memRead
169     );
170
171     wDataMux: mux2 PORT MAP(
172         IN0 => ALUResult, IN1=>wDataMuxin1, sel=>memToReg, DOUT=>WriteData
173     );
174
175 end dataBoi;
176
```