```
1   LIBRARY ieee;
2   USE ieee.std_logic_1164.all;
3
4   LIBRARY work;
5
6   ENTITY alu8_final IS
7      PORT
8      (
9         enable :  IN  STD_LOGIC;
10        ld :  IN  STD_LOGIC;
11        clr :  IN  STD_LOGIC;
12        clk :  IN  STD_LOGIC;
13        a :  IN  STD_LOGIC_VECTOR(7 DOWNTO 0);
14        b :  IN  STD_LOGIC_VECTOR(7 DOWNTO 0);
15        op :  IN  STD_LOGIC_VECTOR(2 DOWNTO 0);
16        hex0 :  OUT  STD_LOGIC_VECTOR(6 DOWNTO 0);
17        hex1 :  OUT  STD_LOGIC_VECTOR(6 DOWNTO 0);
18        hex4 :  OUT  STD_LOGIC_VECTOR(6 DOWNTO 0);
19        hex5 :  OUT  STD_LOGIC_VECTOR(6 DOWNTO 0);
20        hex6 :  OUT  STD_LOGIC_VECTOR(6 DOWNTO 0);
21        hex7 :  OUT  STD_LOGIC_VECTOR(6 DOWNTO 0)
22     );
23  END alu8_final;
24
25  ARCHITECTURE bdf_type OF alu8_final IS
26
27  COMPONENT op_inv
28     PORT(enable : IN STD_LOGIC;
29          op : IN STD_LOGIC_VECTOR(2 DOWNTO 0);
30          op_out : OUT STD_LOGIC_VECTOR(2 DOWNTO 0)
31     );
32  END COMPONENT;
33
34  COMPONENT alu8
35     PORT(a : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
36          b : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
37          op : IN STD_LOGIC_VECTOR(2 DOWNTO 0);
38          result : INOUT STD_LOGIC_VECTOR(7 DOWNTO 0);
39          cout : OUT STD_LOGIC;
40          zero : OUT STD_LOGIC
41     );
42  END COMPONENT;
43
44  COMPONENT register8
45     PORT(ld : IN STD_LOGIC;
46          clr : IN STD_LOGIC;
47          clk : IN STD_LOGIC;
48          d : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
49          Q : OUT STD_LOGIC_VECTOR(7 DOWNTO 0)
50     );
51  END COMPONENT;
52
53  COMPONENT octal_7seg
54     PORT(neg : IN STD_LOGIC;
55          octal_in : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
56          hexDown : OUT STD_LOGIC_VECTOR(6 DOWNTO 0);
57          hexTop : OUT STD_LOGIC_VECTOR(6 DOWNTO 0);
58          sign : OUT STD_LOGIC_VECTOR(6 DOWNTO 0)
59     );
60  END COMPONENT;
61
62  SIGNAL   SYNTHESIZED_WIRE_7 :  STD_LOGIC_VECTOR(7 DOWNTO 0);
```

```
63    SIGNAL    SYNTHESIZED_WIRE_8 :  STD_LOGIC_VECTOR(7 DOWNTO 0);
64    SIGNAL    op_inv1 :  STD_LOGIC_VECTOR(2 DOWNTO 0);
65    SIGNAL    zero1 :  STD_LOGIC;
66    SIGNAL    SYNTHESIZED_WIRE_4 :  STD_LOGIC_VECTOR(7 DOWNTO 0);
67
68
69    BEGIN
70
71
72
73    b2v_inst : op_inv
74    PORT MAP(enable => enable,
75           op => op,
76           op_out => op_inv1);
77
78
79    b2v_inst1 : alu8
80    PORT MAP(a => A_2_ALU,
81           b => B_2_ALU,
82           op => op_inv1,
83           result => result1,
84           zero => zero1);
85
86
87    b2v_inst2 : register8
88    PORT MAP(ld => ld,
89           clr => clr,
90           clk => clk,
91           d => a,
92           Q => A_2_ALU);
93
94
95    b2v_inst3 : octal_7seg
96    PORT MAP(neg => zero1,
97           octal_in => result1,
98           hexDown => hex1,
99           hexTop => hex0);
100
101
102   b2v_inst4 : register8
103   PORT MAP(ld => ld,
104          clr => clr,
105          clk => clk,
106          d => b,
107          Q => B_2_ALU);
108
109
110   b2v_inst5 : octal_7seg
111   PORT MAP(octal_in => A_2_ALU,
112          hexDown => hex6,
113          hexTop => hex7);
114
115
116   b2v_inst6 : octal_7seg
117   PORT MAP(octal_in => B_2_ALU,
118          hexDown => hex5,
119          hexTop => hex4);
120
121
122   END bdf_type;
```