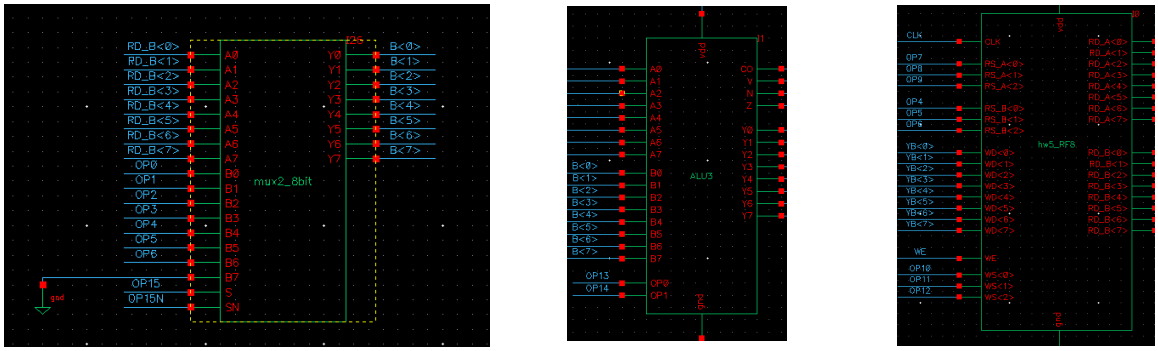
The design of decoder

The input instruction has 16 bits OP[15:0], and its bit definition is illustrated as follows.

Bit IDX	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R type	OP			RD			RS			RA			Unused			
I type	OP			RD			RS			IM						

The decoder is shown in Fig.1. OP15 is used to select the the immediate value OP[6:0] or the value RD_B[7:0] from the register file. A two 8-bit input multiplexer is used to implement this functionality, as shown in Fig.1(a). OP[14:13] are connected to the operation code of ALU, as shown in Fig.1(b). OP[12:10] are connected to the write address WS[2:0] of register file, OP[9:7] are connected to the read address RS_A[2:0] and OP[6:4] are connected to another read address RS_B[2:0], as shown in Fig.1(c).



(a) (b) (c)

Fig.1 The design of decoder

The top level design

The top level design is shown in Fig.2. The output of ALU are buffered to the Register File's write data inputs. One 8-bit input of ALU is from the Register File's output RD_A[7:0], and the other 8-bit input of ALU is from the output of multiplexer, which select Register File's output RD_B[7:0] or the immediate value OP[6:0] based on OP15.

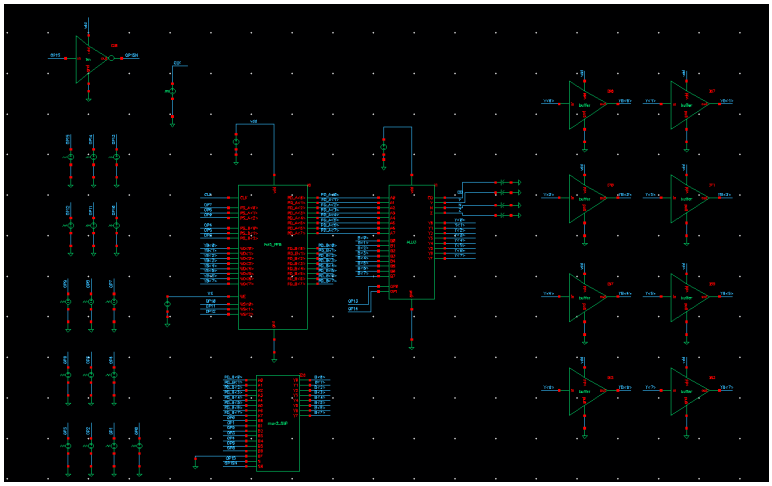


Fig.2 Top level design

Functionality check

The Matlab script and the result for functionality check is shown as follows.

```
close all
clear
clc
simulation_filename = 'result.csv';
expected_filename = 'expected_state.csv';
input_names = {'OP15', 'OP14', 'OP13', 'OP12', 'OP11', 'OP10', 'OP9', 'OP8', 'OP7', 'OP6', 'OP5', 'OP4', 'OP3',
'OP2', 'OP1', 'OP0'};
output_names = {'I0.Q0_7', 'I0.Q0_6', 'I0.Q0_5', 'I0.Q0_4', 'I0.Q0_3', 'I0.Q0_2', 'I0.Q0_1', 'I0.Q0_0', ...
    'I0.Q1_7', 'I0.Q1_6', 'I0.Q1_5', 'I0.Q1_4', 'I0.Q1_3', 'I0.Q1_2', 'I0.Q1_1', 'I0.Q1_0', ...
    'I0.Q2_7', 'I0.Q2_6', 'I0.Q2_5', 'I0.Q2_4', 'I0.Q2_3', 'I0.Q2_2', 'I0.Q2_1', 'I0.Q2_0', ...
    'I0.Q3_7', 'I0.Q3_6', 'I0.Q3_5', 'I0.Q3_4', 'I0.Q3_3', 'I0.Q3_2', 'I0.Q3_1', 'I0.Q3_0', ...
    'I0.Q4_7', 'I0.Q4_6', 'I0.Q4_5', 'I0.Q4_4', 'I0.Q4_3', 'I0.Q4_2', 'I0.Q4_1', 'I0.Q4_0', ...
    'I0.Q5_7', 'I0.Q5_6', 'I0.Q5_5', 'I0.Q5_4', 'I0.Q5_3', 'I0.Q5_2', 'I0.Q5_1', 'I0.Q5_0', ...
    'I0.Q6_7', 'I0.Q6_6', 'I0.Q6_5', 'I0.Q6_4', 'I0.Q6_3', 'I0.Q6_2', 'I0.Q6_1', 'I0.Q6_0', ...
    'I0.Q7_7', 'I0.Q7_6', 'I0.Q7_5', 'I0.Q7_4', 'I0.Q7_3', 'I0.Q7_2', 'I0.Q7_1', 'I0.Q7_0'};
logic_1 = 3;

flag = cpu_check(simulation_filename, expected_filename,
input_names,output_names,logic_1);
```

Fig.3 Matlab script for functionality check

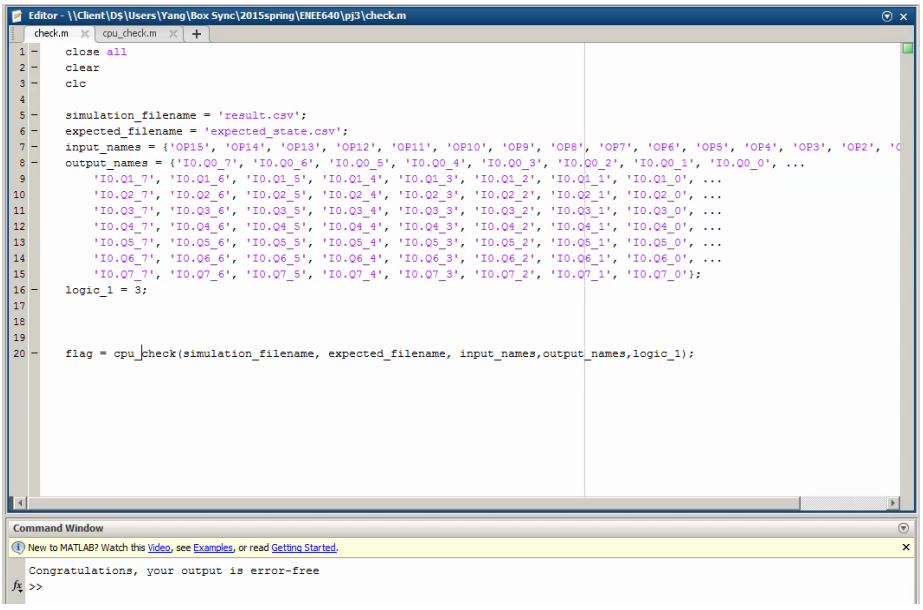


Fig.4 Result of functionality check