

Lab 9- 10: Nano Processor

Group Members

Group: Group 46

- 220322K – Kodisinghe R.R.
- 220440U - Panagoda P.L.A.S.
- 220309D – Karunaratne T.K.M.C.
- 220469P – Perera M.R.S.

Introduction:

In this project, our goal is to design a 4-bit nano-processor capable of executing the following 4 fundamental instructions. We were expected to demonstrate the working implementation in a basys3 board.

Instruction	Description	Format (12-bit instruction)
MOV R, d	Move immediate value d to register R.	1 0 R R R 0 0 0 d d d d
ADD Ra, Rb	Add values in registers Ra and Rb and store the result in Ra.	0 0 Ra Ra Ra Rb Rb Rb 0000
NEG R	2's complement of registers R	0 1 R R R 0 0 0 0 0 0 0 0
JZR R, d	Jump if value in register R is 0, i.e., If $R == 0$ $PC \leq d;$ Else $PC \leq PC + 1;$	1 1 R R R 0 0 0 0 d d d

In the modified version, the following instructions can be executed.

Instruction	Description	Format (14-bit instruction)
-------------	-------------	-----------------------------

MOV R, d	Move immediate value d to register R.	0 0 1 0 R R R 0 0 0 d d d
ADD Ra, Rb	Add the values in registers Ra and Rb and store the result in Ra.	0000 Ra Ra Ra Rb Rb Rb 0000
NEG R	2's complement of registers R	0 0 0 1 R R R 0 0 0 0 0 0
JZR R, d	Jump if value in register R is 0, i.e., If $R == 0$ PC $\leq d$; Else PC $\leq PC + 1$;	0 0 1 1 R R R 0 0 0 0 d d d
SUB Ra, Rb	Subtract the values in registers Ra and Rb and store the result in Ra.	0100 Ra Ra Ra Rb Rb Rb 0000
INC Ra	Increment the value in the register Ra by 1.	0101 Ra Ra Ra 001 0000
DNC Ra	Decrement the value in the register Ra by 1.	0111 Ra Ra Ra 001 0000
MUL Ra, Rb	Multiply the values in registers Ra and Rb and store the result in Ra.	0110 Ra Ra Ra Rb Rb Rb 0000
AND Ra, Rb	Get the bit wise AND value of the values in registers Ra and Rb and store the result in the register Ra.	1100 Ra Ra Ra Rb Rb Rb 0000
OR Ra, Rb	Get the bit wise OR value of the values in registers Ra and Rb and store the result in the register Ra.	1101 Ra Ra Ra Rb Rb Rb 0000
NOT Ra	Get the bit wise NOT value of the value in register Ra and store the result in the register Ra.	1110 Ra Ra Ra 000 0000
XOR Ra, Rb	Get the bit wise XOR value of the values in registers Ra and Rb and store the result in the register Ra.	1111 Ra Ra Ra Rb Rb Rb 0000

COMP Ra, Rb	Compare the value in register Rb with respect to Ra and output whether its greater than, lesser than or equal Ra.	1000 Ra Ra Ra Rb Rb Rb 0000
-------------	---	-----------------------------

To build the processor, the following components should be designed first.

1. Program ROM
2. Instruction Decoder
3. Program Counter
4. 3-bit adder
5. 2 to 1 (3-bit) Multiplexer
6. 2 to 1 (4-bit) Multiplexer
7. Register Bank
8. 8 to 1 (4-bit) Multiplexer
9. Adder subtractor
10. Seven segment display
11. Slow clock

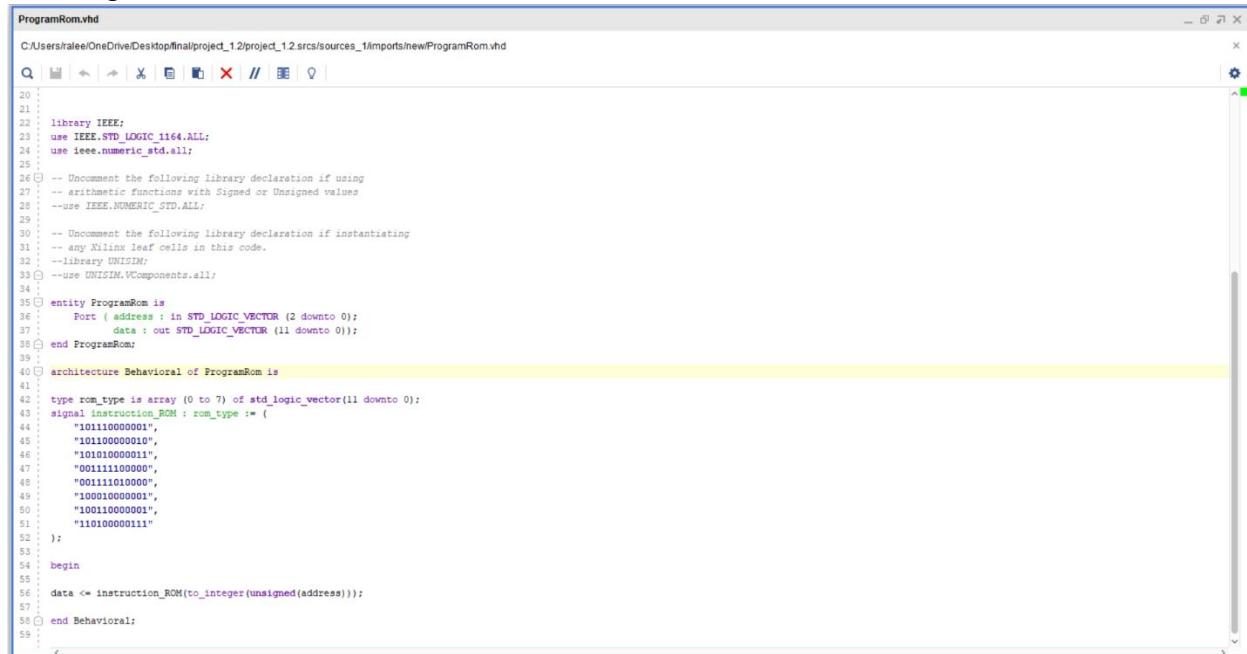
After creating the above components, they were connected using buses.

Then the design was simulated

Finally, we had to implemented it on the BASYS3 board.

Program ROM

1. Design source file



```

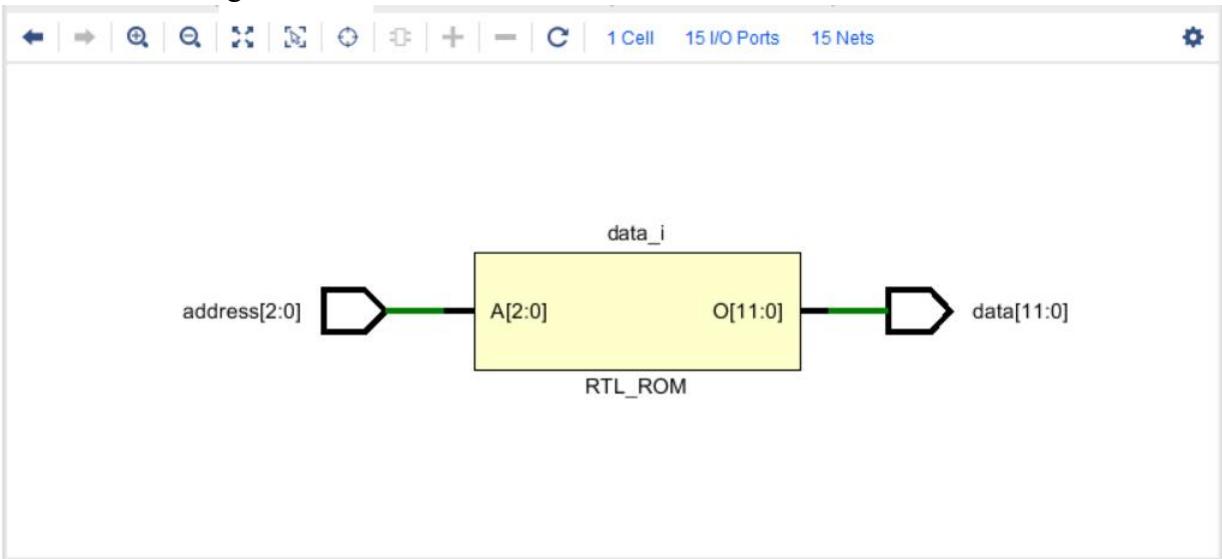
ProgramRom.vhd
C:/Users/ralee/Desktop/final/project_1.2/project_1.2.srcc/sources_1/imports/new/ProgramRom.vhd

entity ProgramRom is
    Port ( address : in STD_LOGIC_VECTOR (2 downto 0);
           data : out STD_LOGIC_VECTOR (11 downto 0));
end ProgramRom;

architecture Behavioral of ProgramRom is
begin
    type rom_type is array (0 to 7) of std_logic_vector(11 downto 0);
    signal instruction_ROM : rom_type := (
        "101110000001",
        "101100000010",
        "101010000001",
        "001111100000",
        "001111010000",
        "100010000001",
        "100110000001",
        "110100000111");
    begin
        data <= instruction_ROM(to_integer(unsigned(address)));
    end Behavioral;
end

```

2. Elaborated design schematic



3. Simulation source file

```
TB_Program_Rom.vhd
C:/Users/alee/Lab 9/Lab 9.srca/sim_1/new/TB_Program_Rom.vhd
? | < | > | X | Q | F | X | S | C | 1 Cell 15 I/O Ports 15 Nets | Settings

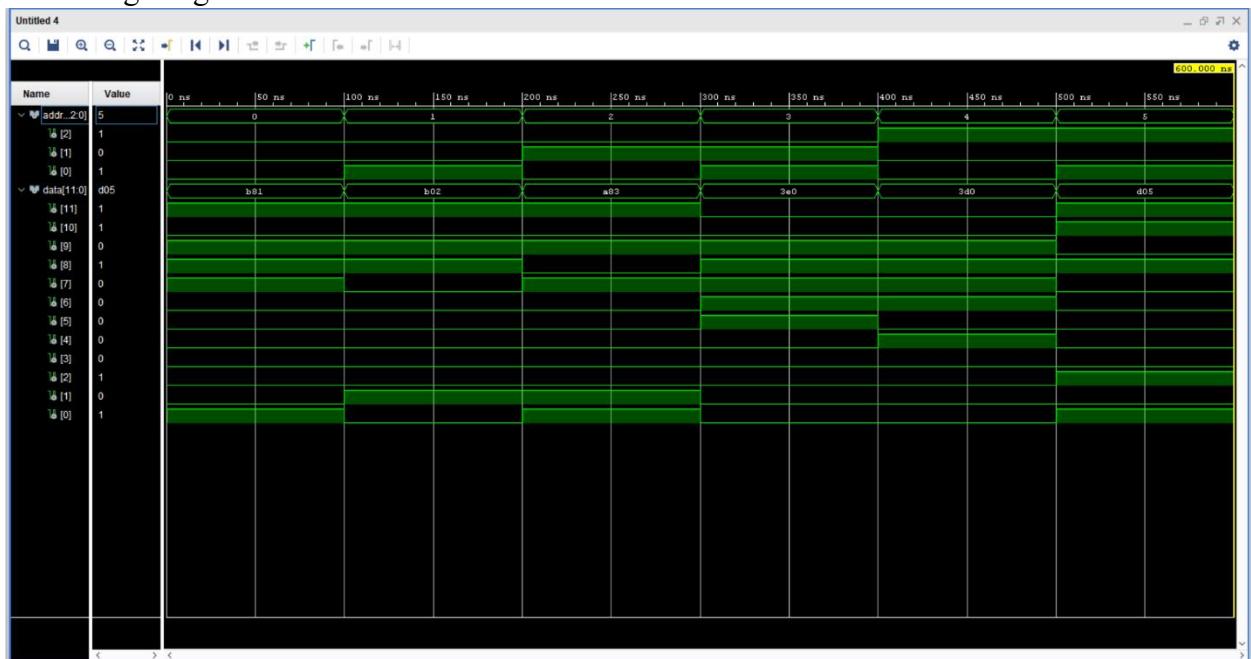
1 -- Company;
2 -- Engineer;
3 --
4 --
5 -- Create Date: 04/18/2024 11:35:36 PM
6 -- Design Name: TB_Program_Rom - Behavioral
7 -- Project Name:
8 -- Target Devices:
9 -- Tool Versions:
10 -- Description;
11 --
12 --
13 -- Dependencies;
14 --
15 -- Revision;
16 -- Revision 0.01 - File Created
17 -- Additional Comments;
18 --
19 -----
20 
21 
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24 use ieee.numeric_std.all;
25 
26 -- Uncomment the following library declaration if using
27 -- arithmetic functions with Signed or Unsigned values
28 --use IEEE.NUMERIC_STD.ALL;
29 
30 -- Uncomment the following library declaration if instantiating
31 -- any Xilinx leaf cells in this code.
32 --library UNISIM;
33 --use UNISIM.VComponents.all;
34 
35 entity TB_Program_Rom is
36 -- Port ();
37 end TB_Program_Rom;
38 
39 architecture Behavioral of TB_Program_Rom is
40 
```

```

41 component Program_Rom
42     Port ( address : in STD_LOGIC_VECTOR (2 downto 0);
43             data : out STD_LOGIC_VECTOR (11 downto 0));
44 end component;
45
46 signal address: STD_LOGIC_VECTOR (2 downto 0);
47 signal data: STD_LOGIC_VECTOR (11 downto 0);
48
49 begin
50
51     U1: Program_Rom
52         port map (
53             address=>address,
54             data=>data
55         );
56
57     process begin
58
59         -- testing all possible values
60         test_loop : for i in 0 to 5 loop
61             address <= std_logic_vector(to_unsigned(i, 3));
62             wait for 100ns;
63         end loop test_loop;
64         wait;
65
66     end process;
67
68 end Behavioral;
69

```

4. Timing Diagram



Instruction Decoder

Instruction decoder is designed using basic logic gates. In this processor instruction decoder decodes the instruction coming from the program ROM and activates/ enables the necessary entities needed for the proper functionality of the specific instruction. This works as the central entity in this processor connecting and commanding all the other entities.

1. Design source file

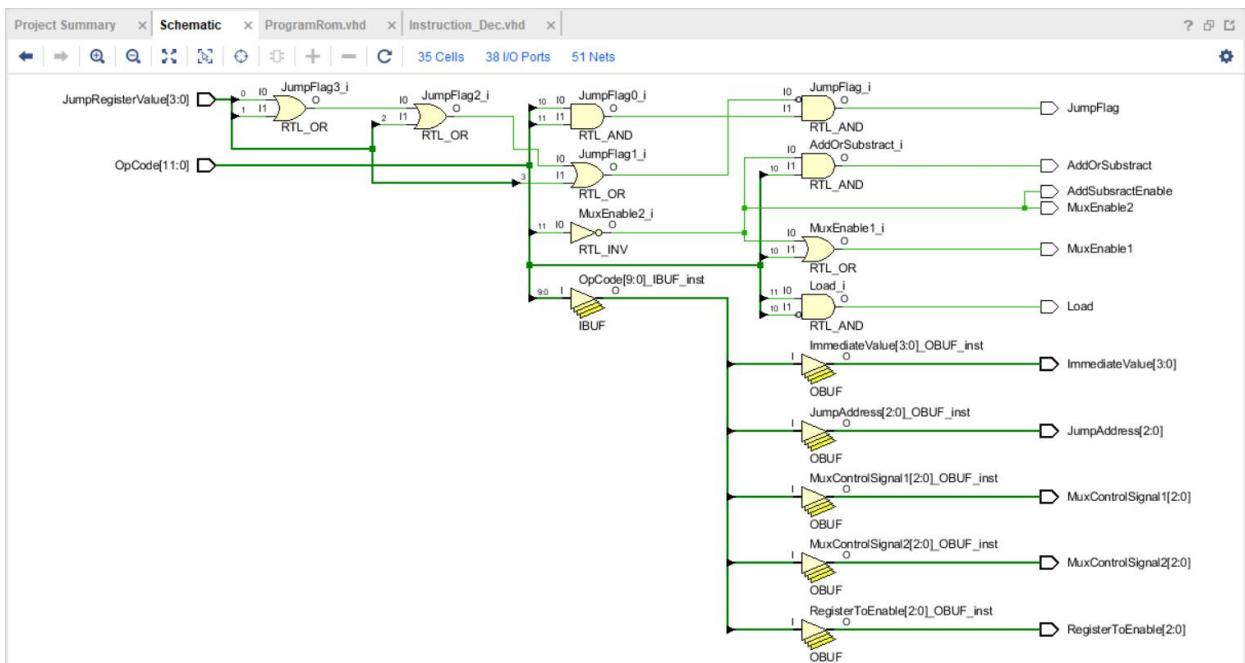
```

Instruction_Dec.vhd
C:/Users/rakeel/Desktop/final/final/new/project_12/project_12/srcs/sources_1/imports/new/Instruction_Dec.vhd

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600

```

2. Elaborated design schematic



3. Simulation source file

```

Sim_Ins_Dec.vhd
C:/Users/raleel/OneDrive/Desktop/final/project_1_2/project_1_2.srca/sim_1/imports/newSim_Ins_Dec.vhd

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity TB_InstructionDecoder is
end TB_InstructionDecoder;
architecture Behavioral of TB_InstructionDecoder is
-- Signals
signal OpCode : STD_LOGIC_VECTOR(11 DOWNTO 0) := (OTHERS => '0');
signal JumpRegisterValue : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0000";
-- Add signals to capture outputs from the module
signal JumpFlag : STD_LOGIC;
signal JumpAddress : STD_LOGIC_VECTOR(2 DOWNTO 0);
signal AddOrSubtract : STD_LOGIC;
signal MuxEnable1 : STD_LOGIC;
signal MuxEnable2 : STD_LOGIC;
signal MuxControlSignal1 : STD_LOGIC_VECTOR(2 DOWNTO 0);
signal MuxControlSignal2 : STD_LOGIC_VECTOR(2 DOWNTO 0);
signal AddSubtractEnable : STD_LOGIC;
--signal RegisterBankEnable : STD_LOGIC;
--signal ImmediateValue : STD_LOGIC_VECTOR(3 DOWNTO 0);
--signal Load : STD_LOGIC;
--signal RegisterToEnable : STD_LOGIC_VECTOR(2 DOWNTO 0);
begin
-- Instantiate the Unit Under Test (UUT)
uut : entity work.Instruction_Dec
port map (
    OpCode => OpCode,
    JumpRegisterValue => JumpRegisterValue,
    JumpFlag => JumpFlag,
    JumpAddress => JumpAddress,
    AddOrSubtract => AddOrSubtract,
    MuxEnable1 => MuxEnable1,
    MuxEnable2 => MuxEnable2,
    MuxControlSignal1 => MuxControlSignal1,
    MuxControlSignal2 => MuxControlSignal2,
    AddSubtractEnable => AddSubtractEnable
);

```

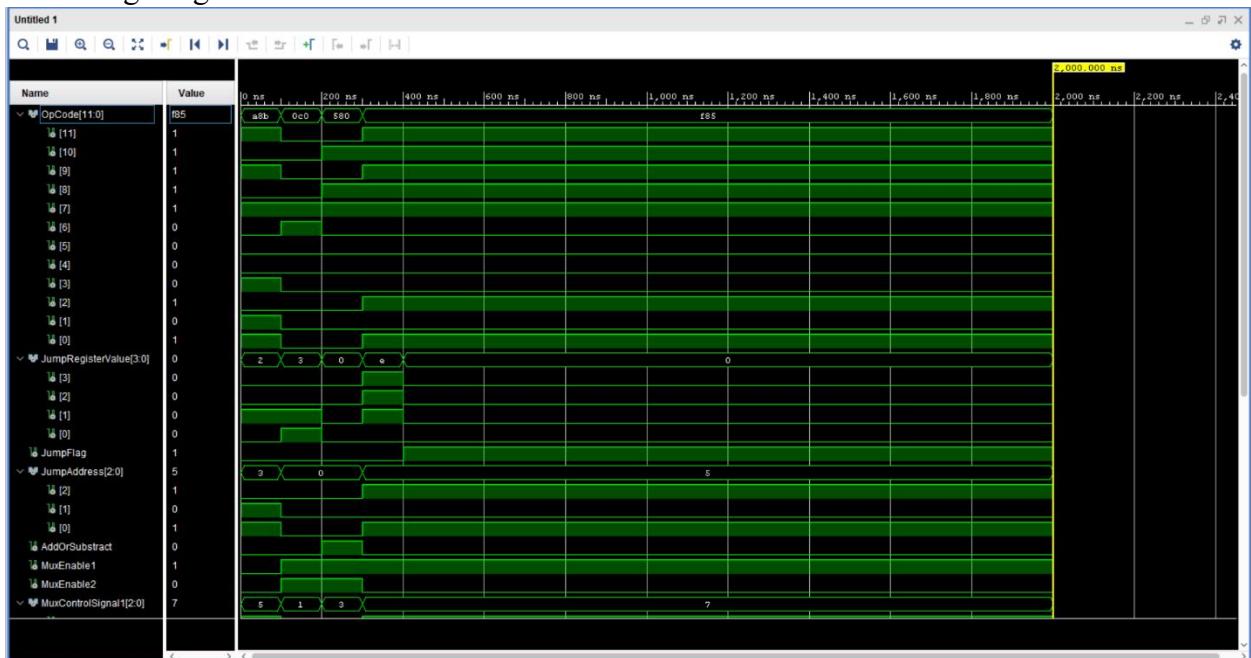
```

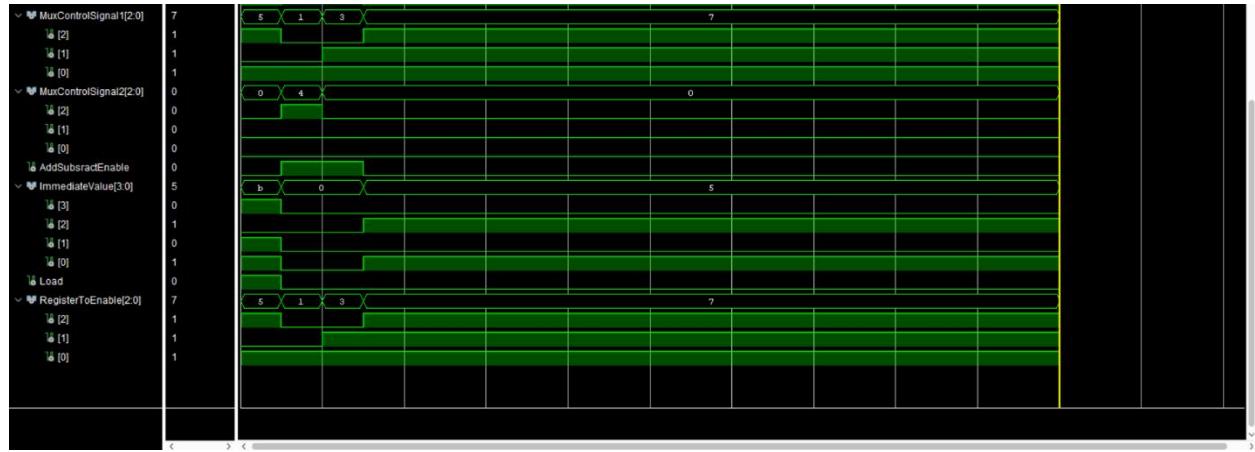
Sim_Ins_Dec.vhd
C:/Users/alee/OneDrive/Desktop/final/final new/project_1_2/project_1_2/srcs/sim_1/imports/newSim_Ins_Dec.vhd

40 :      MuxControlSignal1 => MuxControlSignal1,
41 :      MuxControlSignal2 => MuxControlSignal2,
42 :      --RegisterRankEnable => RegisterRankEnable,
43 :      ImmediateValue => ImmediateValue,
44 :      Load => Load,
45 :      AddSubtractEnable => AddSubtractEnable,
46 :      RegisterToEnable => RegisterToEnable
47 : );
48 :
49 : -- Stimulus process
50 : stim_proc: process
51 : begin
52 :   -- Provide stimulus values here, for example:
53 :   OpCode <= "1010100001011"; -- Sample OpCode value
54 :   JumpRegisterValue <= "0010"; -- Sample JumpRegisterValue value
55 :   wait for 100 ns;
56 :
57 :   -- Add more stimulus if needed
58 :   OpCode <= "0000110000000"; -- Sample OpCode value
59 :   JumpRegisterValue <= "0011"; -- Sample JumpRegisterValue value
60 :   wait for 100 ns;
61 :
62 :   OpCode <= "0101100000000"; -- Sample OpCode value
63 :   JumpRegisterValue <= "0000"; -- Sample JumpRegisterValue value
64 :   wait for 100 ns;
65 :
66 :   OpCode <= "111110000101"; -- Sample OpCode value
67 :   JumpRegisterValue <= "1110"; -- Sample JumpRegisterValue value
68 :   wait for 100 ns;
69 :
70 :   OpCode <= "111110000101"; -- Sample OpCode value
71 :   JumpRegisterValue <= "0000"; -- Sample JumpRegisterValue value
72 :   wait for 100 ns;
73 :
74 :   -- End the simulation
75 :   wait;
76 : end process;
77 :
78 : end Behavioral;
79 :

```

4. Timing Diagram





Program Counter

Program Counter entity holds index of the next instruction to be executed. It is designed using the basic D flip-flops. This entity takes a vector and two logics as inputs and give a vector as the output. When rising edge is taken place it stores the index of the next instruction and returns the index of the instruction which should be executed at the moment.

Each D flip-flop holds one bit (either 1 or 0). All the D flip-flops are related to this entity enabled all the time since Program Rom is a real time component in a processor. Design of the D flip-flop is the design which we were taught in the lectures.

(The instructions are extracted from the Program Rom. In here it refers to the index in the Program Rom)

1. Design source file

The screenshot shows two windows of a VHDL editor. The top window is titled 'ProgramCounter.vhd' and contains the VHDL code for a 'ProgramCounter' entity. The bottom window is titled 'D_FF.vhd' and contains the VHDL code for a 'D_FF' component. Both windows have a standard OS X-style interface with a menu bar and toolbar.

```
ProgramCounter.vhd
C:/Users/rakee/OneDrive/Desktop/final/project_1_2/project_1_2.srsc/sources_1/imports/newProgramCounter.vhd

1 | 
2 | -- Company:
3 | -- Engineer:
4 | --
5 | -- Create Date: 04/09/2024 06:24:56 AM
6 | -- Design Name:
7 | -- Module Name: ProgramCounter - Behavioral
8 | -- Project Name:
9 | -- Target Devices:
10 | -- Tool Versions:
11 | -- Description:
12 | --
13 | -- Dependencies:
14 | --
15 | -- Revision:
16 | -- Revision 0.01 - File Created
17 | -- Additional Comments:
18 | --
19 | -----
20 | library IEEE;
21 | use IEEE.STD_LOGIC_1164.ALL;
22 |
23 | -- Uncomment the following library declaration if using
24 | -- arithmetic functions with Signed or Unsigned values
25 | --use IEEE.NUMERIC_STD.ALL;
26 |
27 | -- Uncomment the following library declaration if instantiating
28 | -- any Xilinx leaf cells in this code.
29 | --library UNISIM;
30 | --use UNISIM.VComponents.all;
31 |
32 | entity ProgramCounter is
33 |     Port ( D : in STD_LOGIC_VECTOR (2 downto 0);
34 |            CLK : in STD_LOGIC;
35 |            CLR : in STD_LOGIC;
36 |            Q : out STD_LOGIC_VECTOR (2 downto 0));
37 | end ProgramCounter;
38 |
39 | architecture Behavioral of ProgramCounter is
40 | begin
41 |     -- 

```

```
ProgramCounter.vhd
C:/Users/rakee/OneDrive/Desktop/final/project_1_2/project_1_2.srsc/sources_1/imports/newProgramCounter.vhd

1 | 
2 | -- Company:
3 | -- Engineer:
4 | --
5 | -- Create Date: 04/09/2024 06:24:56 AM
6 | -- Design Name:
7 | -- Module Name: ProgramCounter - Behavioral
8 | -- Project Name:
9 | -- Target Devices:
10 | -- Tool Versions:
11 | -- Description:
12 | --
13 | -- Dependencies:
14 | --
15 | -- Revision:
16 | -- Revision 0.01 - File Created
17 | -- Additional Comments:
18 | --
19 | -----
20 | library IEEE;
21 | use IEEE.STD_LOGIC_1164.ALL;
22 |
23 | -- Uncomment the following library declaration if using
24 | -- arithmetic functions with Signed or Unsigned values
25 | --use IEEE.NUMERIC_STD.ALL;
26 |
27 | -- Uncomment the following library declaration if instantiating
28 | -- any Xilinx leaf cells in this code.
29 | --library UNISIM;
30 | --use UNISIM.VComponents.all;
31 |
32 | entity ProgramCounter is
33 |     Port ( D : in STD_LOGIC_VECTOR (2 downto 0);
34 |            CLK : in STD_LOGIC;
35 |            CLR : in STD_LOGIC;
36 |            Q : out STD_LOGIC_VECTOR (2 downto 0));
37 | end ProgramCounter;
38 |
39 | architecture Behavioral of ProgramCounter is
40 | begin
41 |     -- 

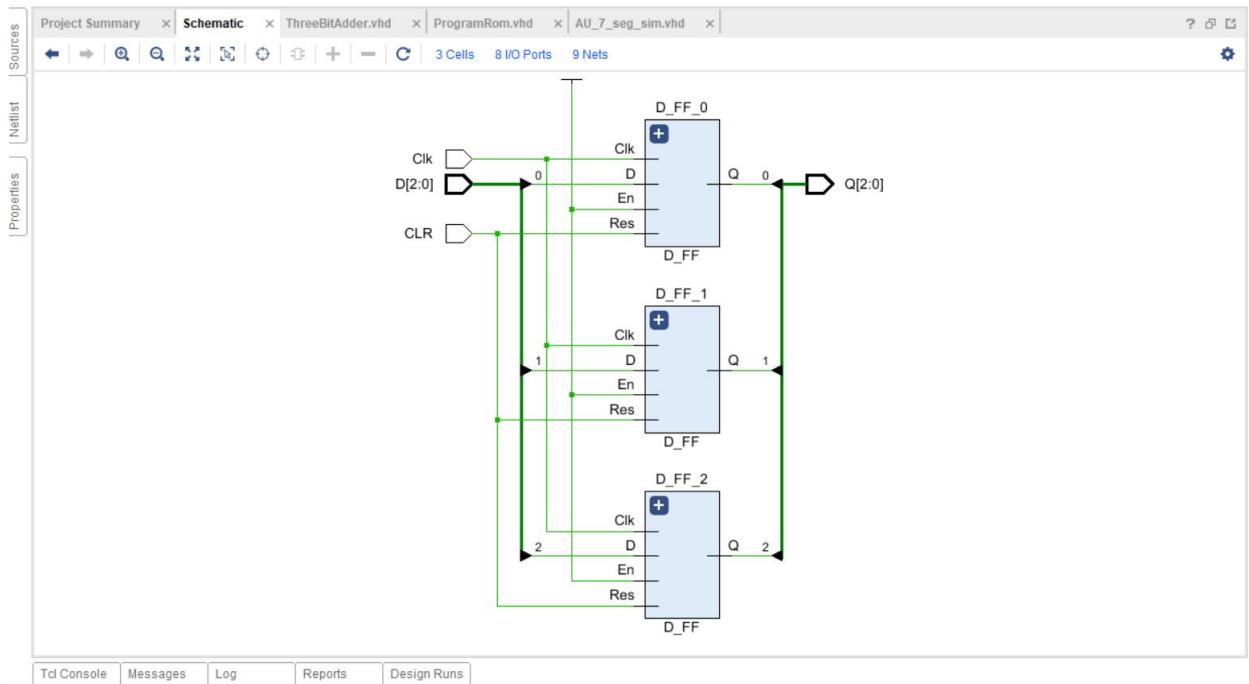
```

```
D_FF.vhd
C:/Users/rakee/OneDrive/Desktop/final/project_1_2/project_1_2.srsc/sources_1/imports/D_FF.vhd

1 | 
2 | component D_FF
3 |     port(
4 |         D : in STD_LOGIC;
5 |         Res : in STD_LOGIC;
6 |         CLK : in STD_LOGIC;
7 |         Q : out STD_LOGIC;
8 |         En : in STD_LOGIC
9 |     );
10 |
11 | end component;
12 |
13 | begin
14 |     D_FF_0 : D_FF
15 |         port map(
16 |             D => D(0),
17 |             Res => CLR,
18 |             CLK => CLK,
19 |             Q => Q(0),
20 |             En => '1'
21 |         );
22 |     D_FF_1 : D_FF
23 |         port map(
24 |             D => D(1),
25 |             Res => CLR,
26 |             CLK => CLK,
27 |             Q => Q(1),
28 |             En => '1'
29 |         );
30 |     D_FF_2 : D_FF
31 |         port map(
32 |             D => D(2),
33 |             Res => CLR,
34 |             CLK => CLK,
35 |             Q => Q(2),
36 |             En => '1'
37 |         );
38 |
39 | end Behavioral;
40 |

```

2. Elaborated design schematic



3. Simulation source file

```

TB_program_counter.vhd
C:/Users/raleel/OneDrive/Desktop/final/project_1.2/project_1.2.srcs/sim_1/imports/new/TB_program_counter.vhd
Q | ̄ | ↻ | ↺ | ↻ | X | // | ̄ | 9 |
1 -- Company;
2 -- Engineer;
3 --
4 --
5 -- Create Date: 04/09/2024 10:55:52 PM
6 -- Design Name:
7 -- Module Name: ProgramCounter_sim - Behavioral
8 -- Project Name:
9 -- Target Devices:
10 -- Tool Versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 --
20 --
21 --
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24 --
25 -- Uncomment the following library declaration if using
26 -- arithmetic functions with Signed or Unsigned values
27 --use IEEE.NUMERIC_STD.ALL;
28 --
29 -- Uncomment the following library declaration if instantiating
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33 --
34 entity TB_ProgramCounter is
35   -- Port ( );
36 end TB_ProgramCounter;
37 --
38 architecture Behavioral of TB_ProgramCounter is
39   --
40 component ProgramCounter is
41   -- Port ( );
42 end component;
43 begin
44   UUT: ProgramCounter
45   port map ( );
46 end Behavioral;

```

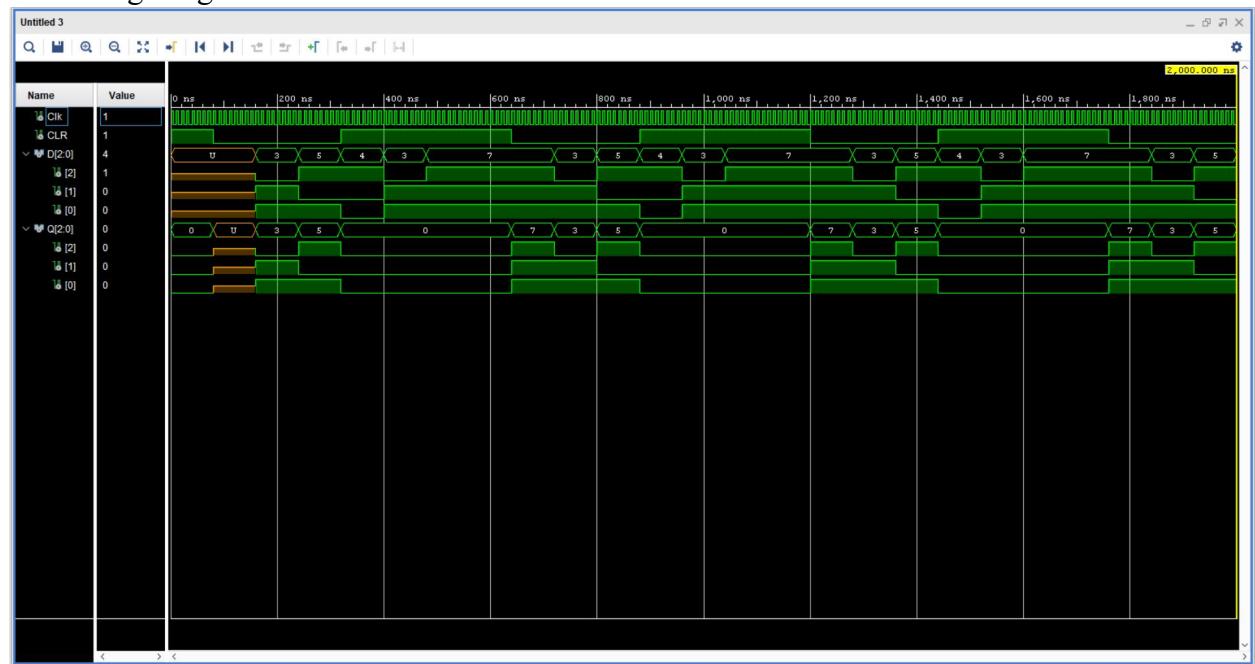
```

TB_program_counter.vhd
C:/Users/rakeel/Desktop/final/project_1_2/project_1_2.srcs/sim_1/imports/new/TB_program_counter.vhd

40 component ProgramCounter is
41   Port ( D : in STD_LOGIC_VECTOR (2 downto 0);
42          Clk : in STD_LOGIC;
43          CLR : in STD_LOGIC;
44          Q : out STD_LOGIC_VECTOR (2 downto 0));
45 end component;
46
47 signal Clk : STD_LOGIC := '0';
48 signal CLR : STD_LOGIC;
49 signal D, Q : STD_LOGIC_VECTOR (2 downto 0);
50
51 begin
52
53   UUT: ProgramCounter
54     PORT MAP (
55       D=>D,
56       CLR=>CLR,
57       Clk=>Clk,
58       Q=>Q
59     );
60
61 process
62 begin
63   begin
64     Clk<=NOT(Clk);
65     wait for 5 ns;
66   end process;
67
68 process
69 begin
70   begin
71     CLR <= '1';
72     wait for 80 ns;
73   end
74   begin
75     CLR <= '0';
76     --D <= "100";
77     wait for 80 ns;
78   end
79   begin
80     D <= "011";
81     wait for 80 ns;
82   end
83   begin
84     D <= "101";
85     wait for 80 ns;
86   end
87   begin
88     D <= "011";
89     wait for 80 ns;
90   end
91   begin
92     D <= "111";
93     wait for 80 ns;
94 end process;
95
96 end Behavioral;
97

```

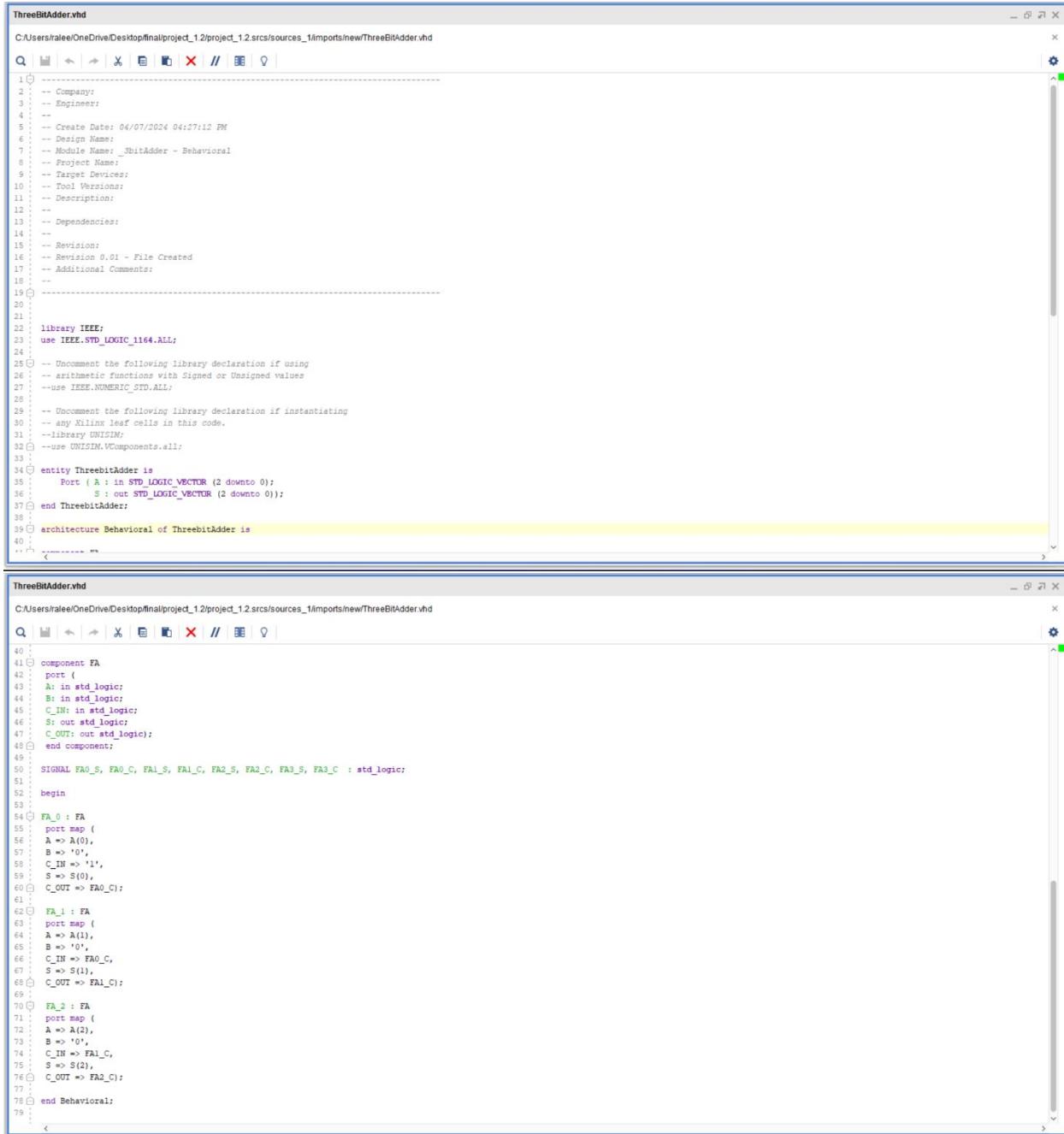
4. Timing Diagram



Three-bit Adder

This component generates the index number of the next instruction to be executed. (in this case index number is incremented by one) This takes 3 bit vector as a input and returns a vector as a output. It is designed using Full Adders.

1. Design source file



The image shows two side-by-side code editors displaying VHDL code for a three-bit adder. The top editor shows the entity and architecture declarations, while the bottom editor shows the internal implementation using full adders.

```
ThreeBitAdder.vhd
C:/Users/raleel/OneDrive/Desktop/final/project_1.2/project_1.2.srsc/sources_1/imports/new/ThreeBitAdder.vhd

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Threbitadder is
    Port ( A : in STD_LOGIC_VECTOR (2 downto 0);
           S : out STD_LOGIC_VECTOR (2 downto 0));
end Threbitadder;

architecture Behavioral of ThrebitAdder is
begin
    -- -----
end Behavioral;
```



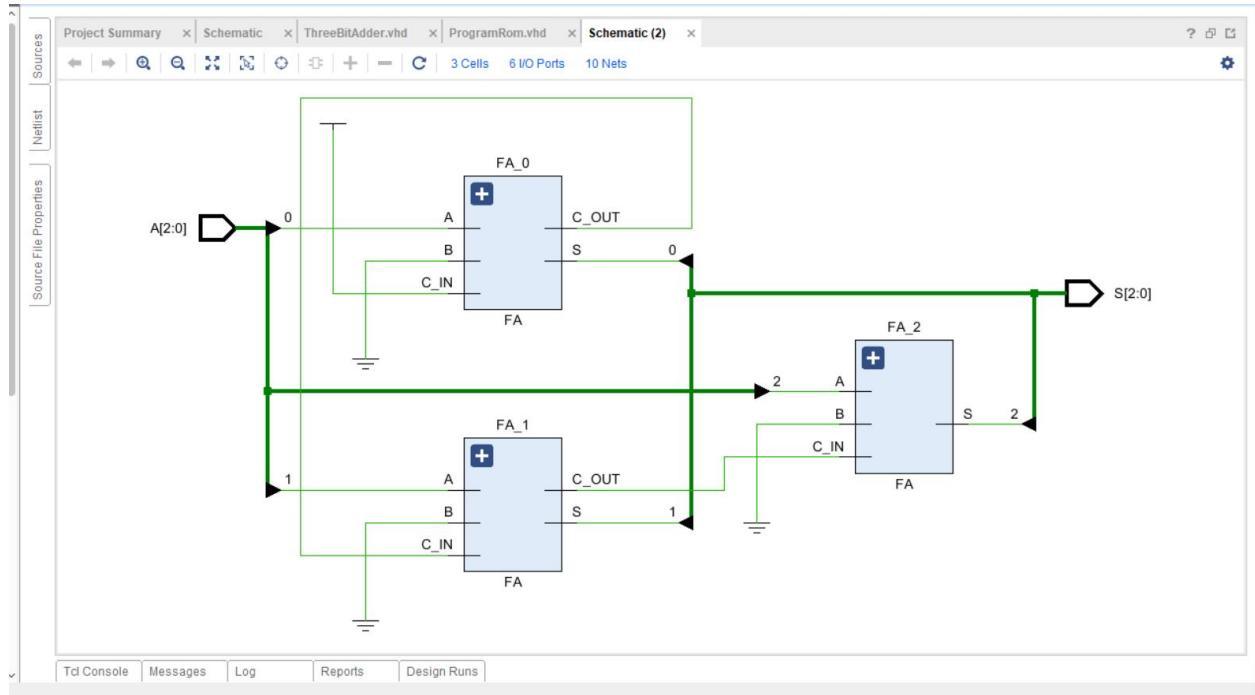
```
ThreeBitAdder.vhd
C:/Users/raleel/OneDrive/Desktop/final/project_1.2/project_1.2.srsc/sources_1/imports/new/ThreeBitAdder.vhd

component FA
    port (
        A: in std_logic;
        B: in std_logic;
        C_IN: in std_logic;
        S: out std_logic;
        C_OUT: out std_logic);
end component;

SIGNAL FA0_S, FA0_C, FA1_S, FA1_C, FA2_S, FA2_C, FA3_S, FA3_C : std_logic;

begin
    begin
        FA_0 : FA
            port map (
                A => A(0),
                B => '0',
                C_IN => '1',
                S => S(0),
                C_OUT => FA0_C);
        FA_1 : FA
            port map (
                A => A(1),
                B => '0',
                C_IN => FA0_C,
                S => S(1),
                C_OUT => FA1_C);
        FA_2 : FA
            port map (
                A => A(2),
                B => '0',
                C_IN => FA1_C,
                S => S(2),
                C_OUT => FA2_C);
    end Behavioral;
```

2. Elaborated design schematic



3. Simulation source file

```

ThreeBitAdder.vhd
C:/Users/raleel/OneDrive/Desktop/final/project_1.2/project_1.2/srcs/sources_1/imports/new/ThreeBitAdder.vhd
Q | L | < | > | X | D | R | X | // | B | ? |
1 -->
2 -- Company:
3 -- Engineer:
4 --
5 -- Create Date: 04/07/2024 04:27:12 PM
6 -- Design Name:
7 -- Module Name: _3bitAdder - Behavioral
8 -- Project Name:
9 -- Target Devices:
10 -- Tool Versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 -----
20 
21 
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24 
25 -- Uncomment the following library declaration if using
26 -- arithmetic functions with Signed or Unsigned values
27 --use IEEE.NUMERIC_STD.ALL;
28 
29 -- Uncomment the following library declaration if instantiating
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33 
34 entity ThreebitAdder is
35   Port ( A : in STD_LOGIC_VECTOR (2 downto 0);
36         S : out STD_LOGIC_VECTOR (2 downto 0));
37 end ThreebitAdder;
38 
39 architecture Behavioral of ThreebitAdder is
40 
41   -----

```

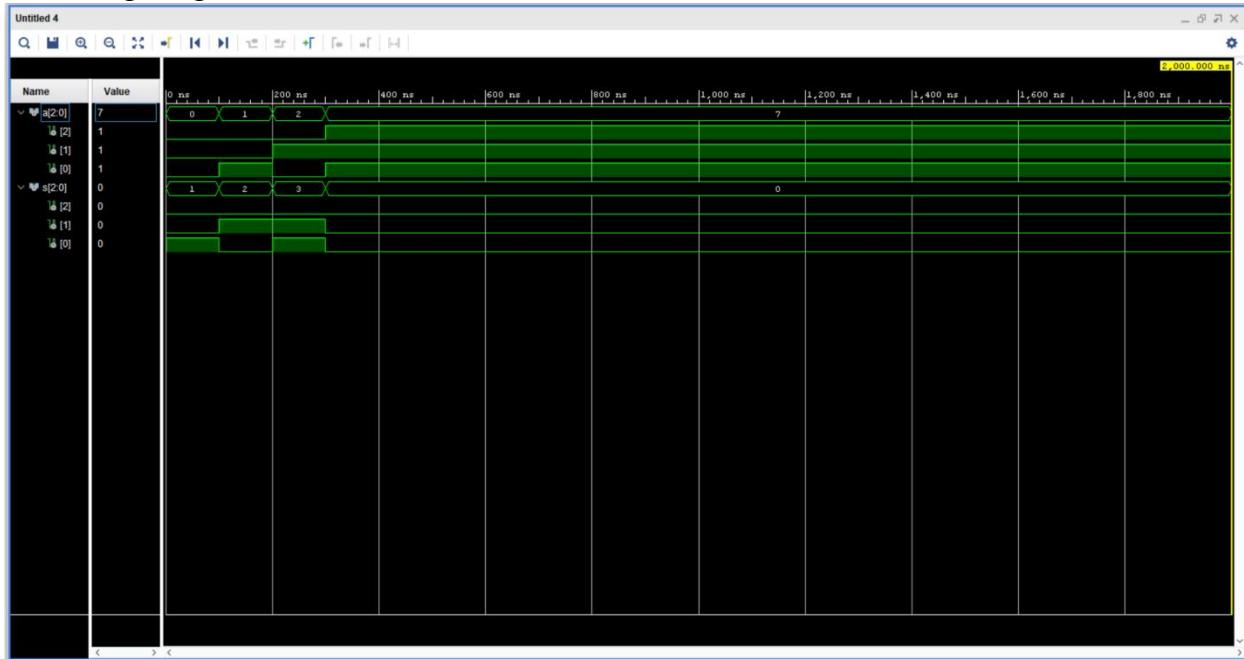
```

ThreeBitAdder.vhd
C:/Users/rakeel/OneDrive/Desktop/final/project_1_2/project_1_2.srcc/sources_1/imports/newThreeBitAdder.vhd

40
41 component FA
42 port (
43 A: in std_logic;
44 B: in std_logic;
45 C_IN: in std_logic;
46 S: out std_logic;
47 C_OUT: out std_logic);
48 end component;
49
50 SIGNAL FA0_S, FA0_C, FA1_S, FA1_C, FA2_S, FA2_C, FA3_S, FA3_C : std_logic;
51
52 begin
53
54 FA_0 : FA
55 port map (
56 A => A(0),
57 B => '0',
58 C_IN => '1',
59 S => S(0),
60 C_OUT => FA0_C);
61
62 FA_1 : FA
63 port map (
64 A => A(1),
65 B => '0',
66 C_IN => FA0_C,
67 S => S(1),
68 C_OUT => FA1_C);
69
70 FA_2 : FA
71 port map (
72 A => A(2),
73 B => '0',
74 C_IN => FA1_C,
75 S => S(2),
76 C_OUT => FA2_C);
77
78 end Behavioral;
79

```

4. Timing Diagram



2-to-1 (3 bit) Multiplexer

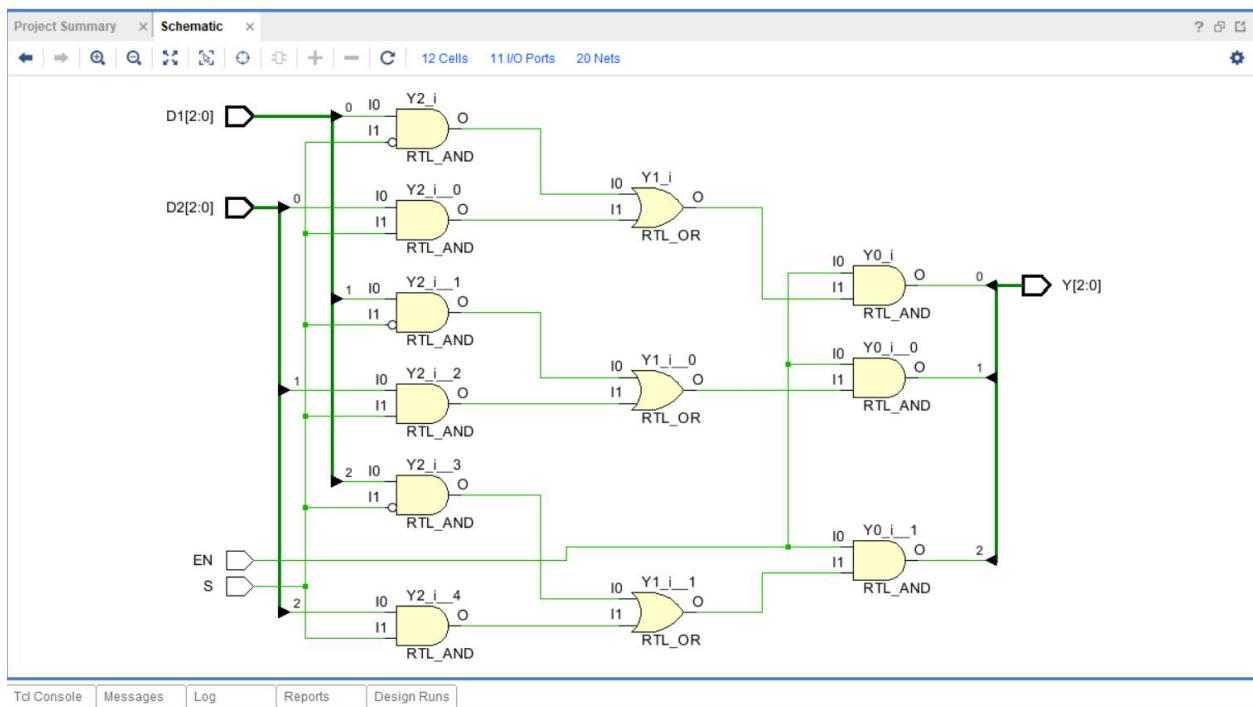
The 2-way 3-bit multiplexer is a circuit designed to select one of two 3-bit numbers based on a selection signal, called (S). This signal determines which of the input numbers is passed through to the output. Typically, when 'S' is low (0), the input from D1 is selected, and when 'S' is high (1), the input from D2 is selected.

1. Design source file

```
Two_3_bit_MUX.vhd
C:/Users/rakeel/Desktop/final/project_1.2/project_1.2/srcs/sources_1/imports/new/Two_3_bit_MUX.vhd

1 --- Company:
2 --- Engineer:
3 --- 
4 --- Create Date: 04/07/2024 11:53:08 PM
5 --- Design Name:
6 --- Module Name: Multiplexer - Behavioral
7 --- Project Name:
8 --- Target Devices:
9 --- Tool Versions:
10 --- Description:
11 --- 
12 --- Dependencies:
13 --- 
14 --- Revision:
15 --- Revision 0.01 - File Created
16 --- Additional Comments:
17 --- 
18 library IEEE;
19 use IEEE.STD_LOGIC_1164.ALL;
20 -- Uncomment the following library declaration if using
21 -- arithmetic functions with Signed or Unsigned values
22 --use IEEE.NUMERIC_STD.ALL;
23 -- 
24 -- Uncomment the following library declaration if instantiating
25 -- any Xilinx leaf cells in this code.
26 --library UNISIM;
27 --use UNISIM.VComponents.all;
28 -- 
29 entity Two_3_bit_MUX is
30     Port ( D1 : in STD_LOGIC_VECTOR (2 downto 0);
31            D2 : in STD_LOGIC_VECTOR (2 downto 0);
32            EN : in STD_LOGIC;
33            S : in STD_LOGIC;
34            Y : out STD_LOGIC_VECTOR (2 downto 0));
35 end Two_3_bit_MUX;
36 -- 
37 architecture Behavioral of Two_3_bit_MUX is
38 begin
39     begin
40         Y(0) <= EN AND ((D1(0) AND NOT(S)) OR (D2(0) AND S));
41         Y(1) <= EN AND ((D1(1) AND NOT(S)) OR (D2(1) AND S));
42         Y(2) <= EN AND ((D1(2) AND NOT(S)) OR (D2(2) AND S));
43     end Behavioral;
44 end;
```

2. Elaborated design schematic



3. Simulation source file

TB_Multiplexer.vhd

```

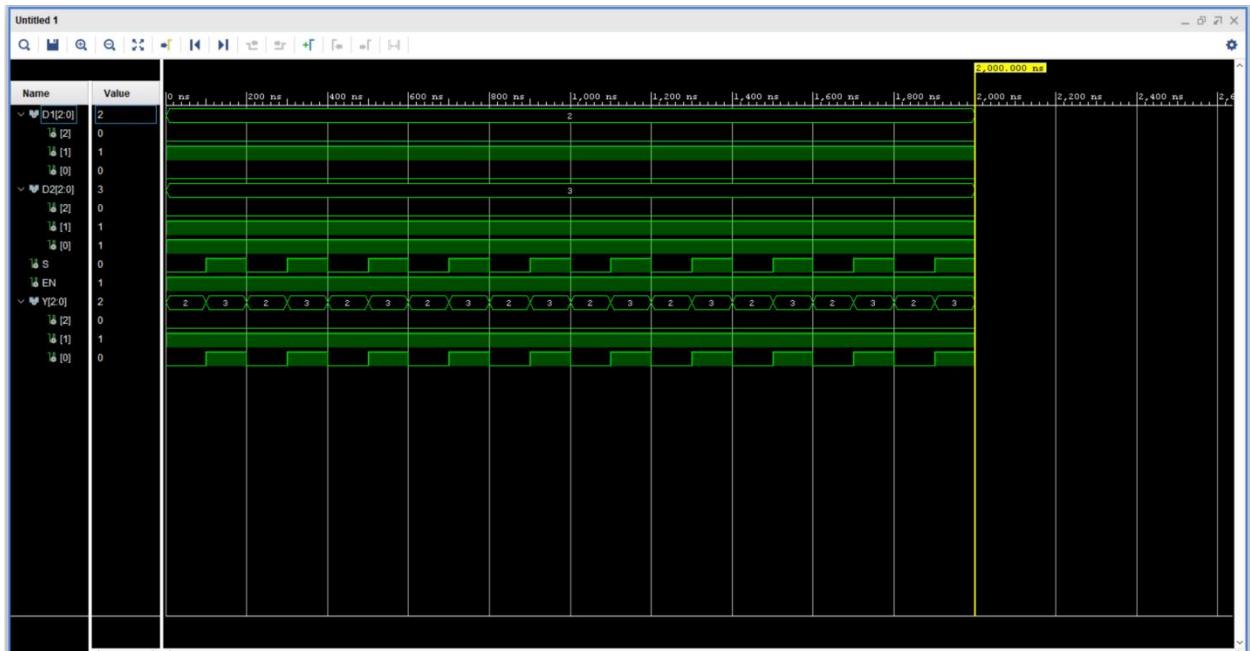
1 -- Company:
2 -- Engineer:
3 --
4 -- Create Date: 04/07/2024 11:32:42 PM
5 -- Design Name:
6 -- Module Name: TB_Multiplexer - Behavioral
7 -- Project Name:
8 -- Target Devices:
9 -- Tool Versions:
10 -- Description:
11 -- Additional Comments:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 
20 
21 
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24 
25 -- Uncomment the following library declaration if using
26 -- arithmetic functions with Signed or Unsigned values
27 --use IEEE.NUMERIC_STD.ALL;
28 
29 -- Uncomment the following library declaration if instantiating
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33 
34 entity TB_Two_3_MUX is
35   -- Port ( );
36 end TB_Two_3_MUX;
37 
38 architecture Behavioral of TB_Two_3_MUX is
39 
40 COMPONENT Two_3_bit_MUX
41   PORT(D1 : in STD_LOGIC_VECTOR (2 downto 0);
42        D2 : in STD_LOGIC_VECTOR (2 downto 0);
43        S : in STD_LOGIC;
44        EN : in STD_LOGIC;
45        Y : out STD_LOGIC_VECTOR (2 downto 0));
46 END COMPONENT;
47 
48 SIGNAL D1: STD_LOGIC_VECTOR (2 downto 0);
49 SIGNAL D2: STD_LOGIC_VECTOR (2 downto 0);
50 SIGNAL S: STD_LOGIC;
51 SIGNAL EN: STD_LOGIC;
52 SIGNAL Y: STD_LOGIC_VECTOR (2 downto 0);
53 
54 begin
55 
56 UUT: Two_3_bit_MUX PORT MAP(
57   D1=>D1,
58   D2=>D2,
59   S=>S,
60   EN=>EN,
61   Y=>Y
62 );
63 
64 process
65 begin
66 
67 EN<='1';
68 D1 <= "010";
69 D2 <= "011";
70 S <> '0';
71 
72 WAIT FOR 100 ns;
73 S <= '1';
74 
75 WAIT FOR 100 ns;
76 
77 end process;
78 
79 end Behavioral;
80 
```

TB_Multiplexer.vhd

```

41 PORT(D1 : in STD_LOGIC_VECTOR (2 downto 0);
42        D2 : in STD_LOGIC_VECTOR (2 downto 0);
43        S : in STD_LOGIC;
44        EN : in STD_LOGIC;
45        Y : out STD_LOGIC_VECTOR (2 downto 0));
46 END COMPONENT;
47 
48 SIGNAL D1: STD_LOGIC_VECTOR (2 downto 0);
49 SIGNAL D2: STD_LOGIC_VECTOR (2 downto 0);
50 SIGNAL S: STD_LOGIC;
51 SIGNAL EN: STD_LOGIC;
52 SIGNAL Y: STD_LOGIC_VECTOR (2 downto 0);
53 
54 begin
55 
56 UUT: Two_3_bit_MUX PORT MAP(
57   D1=>D1,
58   D2=>D2,
59   S=>S,
60   EN=>EN,
61   Y=>Y
62 );
63 
64 process
65 begin
66 
67 EN<='1';
68 D1 <= "010";
69 D2 <= "011";
70 S <> '0';
71 
72 WAIT FOR 100 ns;
73 S <= '1';
74 
75 WAIT FOR 100 ns;
76 
77 end process;
78 
79 end Behavioral;
80 
```

4. Timing Diagram



2-to-1 (4 bit) Multiplexer

The 2-way 4-bit multiplexer is a circuit designed to select one of two 4-bit numbers based on a selection signal, called (S). This signal determines which of the input numbers is passed through to the output. Typically, when 'S' is low (0), the input from D1 is selected, and when 'S' is high (1), the input from D2 is selected.

1. Design source file

```

Two_4bit_MUX.vhd
C:/Users/raleel/Desktop/final/project_1.2/project_1.2.srsc/sources_1/imports/new/Two_4bit_MUX.vhd
Q | L | < | > | X | U | D | R | // | M | ? |
1 -- Company:
2 -- Engineer:
3 --
4 --
5 -- Create Date: 04/07/2024 11:24:14 PM
6 -- Design Name:
7 -- Module Name: Multiplexer - Behavioral
8 -- Project Name:
9 -- Target Devices:
10 -- Tool Versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 -----
20 --
21 --
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24 --
25 -- Uncomment the following library declaration if using
26 -- arithmetic functions with Signed or Unsigned values
27 --use IEEE.Numeric_STD.ALL;
28 --
29 -- Uncomment the following library declaration if instantiating
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33 --
34 entity Two_4bit_MUX is
35     Port ( D1 : in STD.LOGIC_VECTOR (3 downto 0);
36            D2 : in STD.LOGIC_VECTOR (3 downto 0);
37            S : in STD.LOGIC;
38            EN : in STD.LOGIC;
39            Y : out STD.LOGIC_VECTOR (3 downto 0));
40 end Two_4bit_MUX;
41

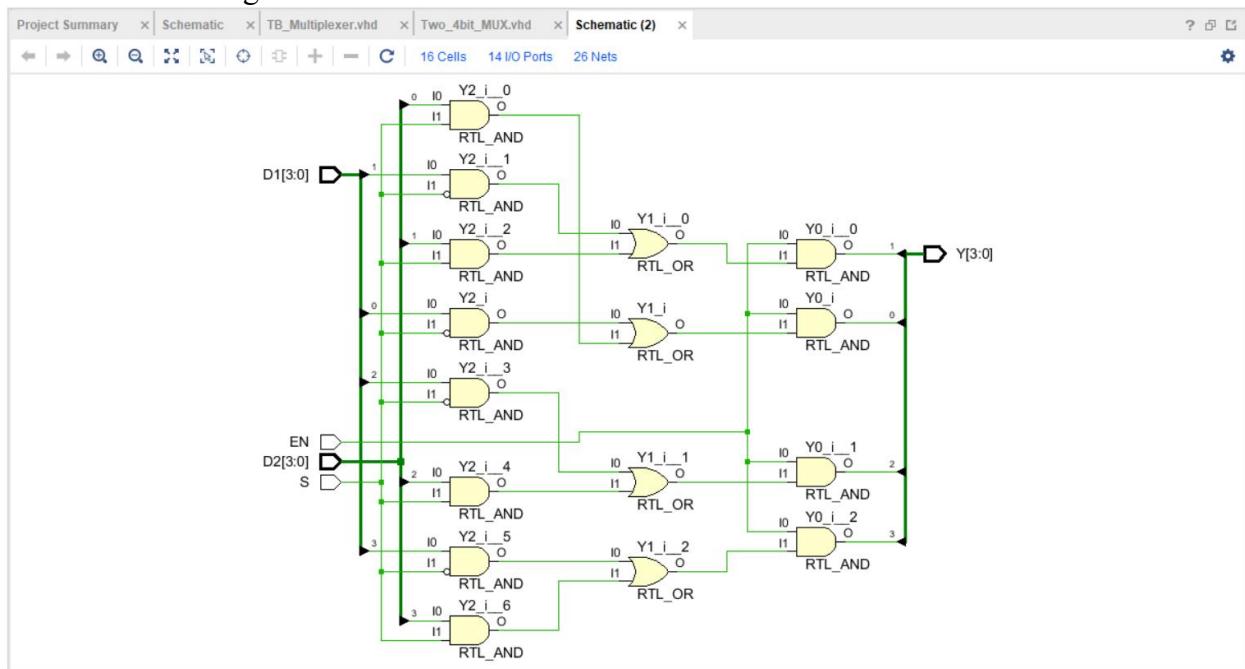
```

```

41
42 architecture Behavioral of Two_4bit_MUX is
43 begin
44
45     Y(0) <= EN AND ((D1(0) AND NOT(S)) OR (D2(0) AND S));
46     Y(1) <= EN AND ((D1(1) AND NOT(S)) OR (D2(1) AND S));
47     Y(2) <= EN AND ((D1(2) AND NOT(S)) OR (D2(2) AND S));
48     Y(3) <= EN AND ((D1(3) AND NOT(S)) OR (D2(3) AND S));
49
50 end Behavioral;
51

```

2. Elaborated design schematic



3. Simulation source file

```

TB_Multiplexer.vhd (2)
C:/Users/raleel/OneDrive/Desktop/final/project_1.2/project_1.2.srsc/sim_1/imports/nana_pro/2-way-4-bit-multiplexer/2-way-4-bit-multiplexer/4 bit 2 to 1 multiplier.srsc/sim_1/new/TB_Multiplexer.vhd

1 -- Company;
2 -- Engineer;
3 --
4 --
5 -- Create Date: 04/07/2024 11:32:42 PM
6 -- Design Name: TB_Multiplexer - Behavioral
7 -- Module Name: TB_Multiplexer
8 -- Project Name:
9 -- Target Devices:
10 -- Tool Versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 -----
20
21
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24
25 -- Uncomment the following library declaration if using
26 -- arithmetic functions with Signed or Unsigned values
27 --use IEEE.NUMERIC_STD.ALL;
28
29 -- Uncomment the following library declaration if instantiating
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33
34 entity TB_Two_4_MUX is
35   -- Port ();
36 end TB_Two_4_MUX;
37
38 architecture Behavioral of TB_Two_4_MUX is
39
40 COMPONENT Two_4bit_MUX
41   PORT ( );
42 end component;

```

```

TB_Multiplexer.vhd (2)
C:\Users\alee\OneDrive\Desktop\final\project_1.2\project_1.2.srs\sim_1\imports\nana_pro\2-way-4-bit-multiplexer\2-way-4-bit-multiplexer\4 bit 2 to 1 multiplier.srs\sim_1\new\TB_Multiplexer.vhd

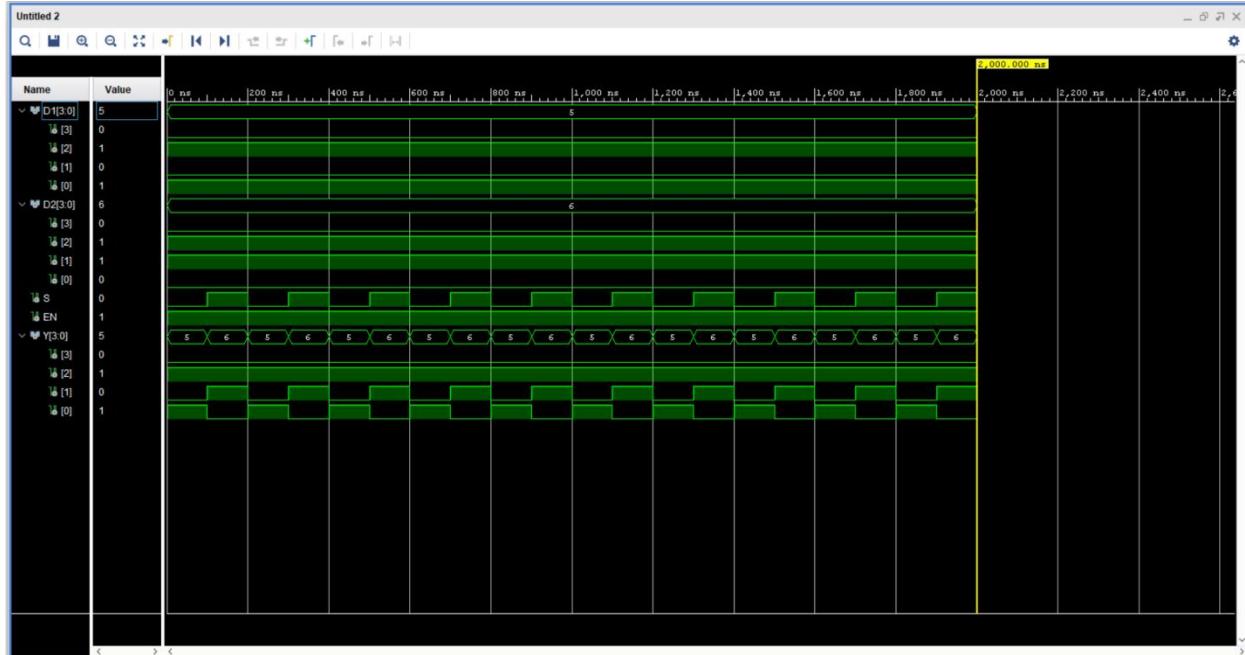
PORT(D1 : in STD_LOGIC_VECTOR (3 downto 0);
      D2 : in STD_LOGIC_VECTOR (3 downto 0);
      S : in STD_LOGIC;
      EN : in STD_LOGIC;
      Y : out STD_LOGIC_VECTOR (3 downto 0));
END COMPONENT;

SIGNAL D1: STD_LOGIC_VECTOR (3 downto 0);
SIGNAL D2: STD_LOGIC_VECTOR (3 downto 0);
SIGNAL S : STD_LOGIC;
SIGNAL EN : STD_LOGIC;
SIGNAL Y : STD_LOGIC_VECTOR (3 downto 0);

begin
  UUT: Two_4bit_MUX PORT MAP(
    D1=>D1,
    D2=>D2,
    S=>S,
    EN=>EN,
    Y=>Y
  );
  process
  begin
    EN<='1';
    D1 <="0101";
    D2 <="0110";
    S <='0';
    WAIT FOR 100 ns;
    S <='1';
    WAIT FOR 100 ns;
  end process;
end Behavioral;

```

4. Timing Diagram



Register Bank

1. Design source file

```

Reg_Bank.vhd
C:/Users/raleel/OneDrive/Desktop/final/project_12/project_12.srcs/sources_1/imports/newReg_Bank.vhd

Q | ̄ | ← | → | X | B | // | ■ | Q |

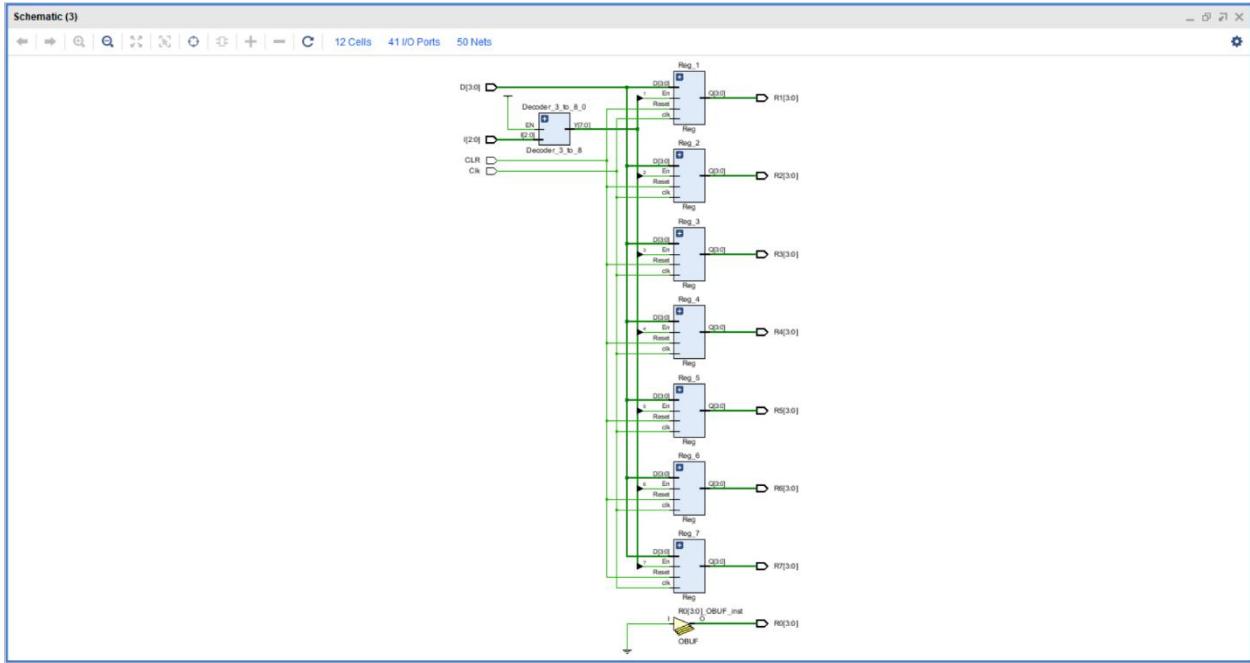
1 | -- Company:
2 | --- Engineer:
3 |
4 |
5 | -- Create Date: 04/07/2024 09:40:53 AM
6 | -- Design Name:
7 | --- Module Name: Reg_Bank - Behavioral
8 | -- Project Name:
9 | --- Target Devices:
10 | --- Tool Versions:
11 | --- Description:
12 |
13 | --- Dependencies:
14 |
15 | --- Revision:
16 | --- Revision 0.01 - File Created
17 | --- Additional Comments:
18 |
19 -----
20 |
21 |
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24 |
25 -- Uncomment the following library declaration if using
26 -- arithmetic functions with Signed or Unsigned values
27 --use IEEE.NUMERIC_STD.ALL;
28 |
29 -- Uncomment the following library declaration if instantiating
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33 |
34 entity Reg_Bank is
35     Port ( D : in STD_LOGIC_VECTOR (3 downto 0);
36             I : in STD_LOGIC_VECTOR (2 downto 0);
37             Clk : in STD_LOGIC;
38             CLR : in STD_LOGIC; --Reset from push Button
39             R0 : out STD_LOGIC_VECTOR (3 downto 0);
40             R1 : out STD_LOGIC_VECTOR (3 downto 0);
41             R2 : out STD_LOGIC_VECTOR (3 downto 0);
42             R3 : out STD_LOGIC_VECTOR (3 downto 0);
43             R4 : out STD_LOGIC_VECTOR (3 downto 0);
44             R5 : out STD_LOGIC_VECTOR (3 downto 0);
45             R6 : out STD_LOGIC_VECTOR (3 downto 0);
46             R7 : out STD_LOGIC_VECTOR (3 downto 0));
47 end Reg_Bank;
48 |
49 architecture Behavioral of Reg_Bank is
50 |
51 component Decoder_3_to_8
52     Port ( I : in STD_LOGIC_VECTOR (2 downto 0);
53             EN : in STD_LOGIC;
54             Y : out STD_LOGIC_VECTOR (7 downto 0));
55 end component;
56 |
57 component Reg
58     Port ( D : in STD_LOGIC_VECTOR (3 downto 0);
59             En : in STD_LOGIC;
60             Reset: in STD_LOGIC;
61             Clk : in STD_LOGIC;
62             Q : out STD_LOGIC_VECTOR (3 downto 0)
63         );
64 end component;
65 |
66 signal Reg_Sel: STD_LOGIC_VECTOR (7 downto 0);
67 begin
68 |
69 |
70 Decoder_3_to_8_0: Decoder_3_to_8
71     port map(
72         I => I,
73         EN => '1',
74         Y => Reg_Sel
75     );
76 |
77 R0 <= "0000";
78 |
79 Reg_1: Reg
80     port map(

```

```
Reg_Bank.vhd
C:/Users/rakee/Desktop/Final/project_1.2/project_1.2.srcs/sources_1/imports/newReg_Bank.vhd

75 Reg_1: Reg
76     port map(
77         D => D,
78         EN => Reg_Sel(1),
79         Reset => CLR,
80         Clk => Clk,
81         Q => R1
82     );
83
84 Reg_2: Reg
85     port map(
86         D => D,
87         EN => Reg_Sel(2),
88         Reset => CLR,
89         Clk => Clk,
90         Q => R2
91     );
92
93 Reg_3: Reg
94     port map(
95         D => D,
96         EN => Reg_Sel(3),
97         Reset => CLR,
98         Clk => Clk,
99         Q => R3
100    );
101
102 Reg_4: Reg
103     port map(
104         D => D,
105         EN => Reg_Sel(4),
106         Reset => CLR,
107         Clk => Clk,
108         Q => R4
109     );
110
111 Reg_5: Reg
112     port map(
113         D => D,
114         EN => Reg_Sel(5),
115         Reset => CLR,
116         Clk => Clk,
117         Q => R5
118     );
119
120 Reg_6: Reg
121     port map(
122         D => D,
123         EN => Reg_Sel(6),
124         Reset => CLR,
125         Clk => Clk,
126         Q => R6
127     );
128
129 Reg_7: Reg
130     port map(
131         D => D,
132         EN => Reg_Sel(7),
133         Reset => CLR,
134         Clk => Clk,
135         Q => R7
136     );
137
138 end Behavioral;
139
140
141
142
143
```

2. Elaborated design schematic



3. Simulation source file

```

TB_Reg_Bank.vhd
C:\Users\ralee\Lab 9\Lab 9.srcc\sim_1\new\TB_Reg_Bank.vhd

1 --> Company;
2 --> Engineer;
3 --> Create Date: 04/07/2024 10:22:36 AM
4 --> Design Name:
5 --> Module Name: TB_Reg_Bank - Behavioral
6 --> Project Name:
7 --> Target Devices:
8 --> Tool Versions:
9 --> Description:
10 --> Dependencies:
11 --> Revision:
12 --> Revision 0.01 - File Created
13 --> Additional Comments:
14 -->
15 -->
16 -->
17 -->

18 -->
19 -->
20 -->
21 -->
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24 -->
25 --> Uncomment the following library declaration if using
26 --> arithmetic functions with Signed or Unsigned values
27 -->use IEEE.NUMERIC_STD.ALL;
28 -->
29 --> Uncomment the following library declaration if instantiating
30 --> any Xilinx leaf cells in this code.
31 -->library UNISIM;
32 -->use UNISIM.VComponents.all;
33 -->
34 entity TB_Reg_Bank is
35 --> Port ();
36 end TB_Reg_Bank;
37 -->
38 architecture Behavioral of TB_Reg_Bank is
39 -->
40 component Reg_Bank
41

```

TB_Reg_Bank.vhd

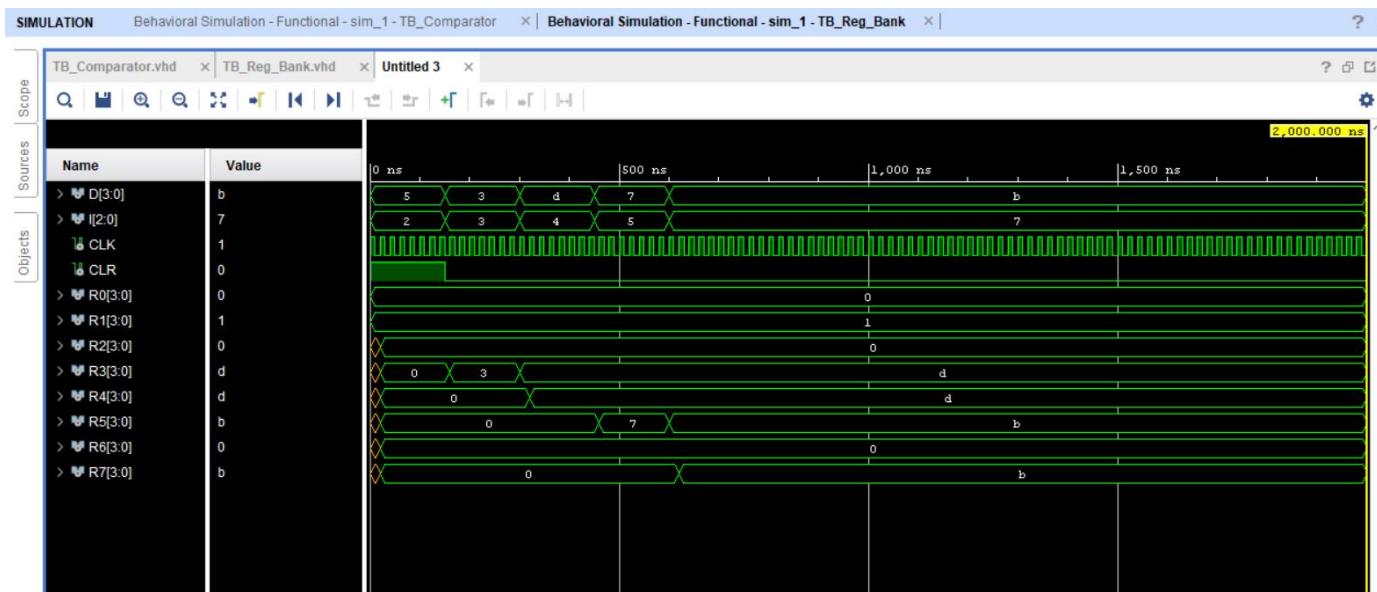
```

C:/Users/steelLab 9/Lab 9/srcs/sim_1/new/TB_Reg_Bank.vhd

Q | I | D | X | // | S | Q |
1 Port ( D : in STD_LOGIC_VECTOR (3 downto 0);
2    I : in STD_LOGIC_VECTOR (2 downto 0);
3    Clk : in STD_LOGIC;
4    CLR : in STD_LOGIC;
5    R0 : out STD_LOGIC_VECTOR (3 downto 0);
6    R1 : out STD_LOGIC_VECTOR (3 downto 0);
7    R2 : out STD_LOGIC_VECTOR (3 downto 0);
8    R3 : out STD_LOGIC_VECTOR (3 downto 0);
9    R4 : out STD_LOGIC_VECTOR (3 downto 0);
10   R5 : out STD_LOGIC_VECTOR (3 downto 0);
11   R6 : out STD_LOGIC_VECTOR (3 downto 0);
12   R7 : out STD_LOGIC_VECTOR (3 downto 0));
13 end component;
14
15 signal D : STD_LOGIC_VECTOR (3 downto 0);
16 signal I : STD_LOGIC_VECTOR (2 downto 0);
17 signal Clk : STD_LOGIC;
18 signal CLR : STD_LOGIC;
19 signal R0 : STD_LOGIC_VECTOR (3 downto 0);
20 signal R1 : STD_LOGIC_VECTOR (3 downto 0);
21 signal R2 : STD_LOGIC_VECTOR (3 downto 0);
22 signal R3 : STD_LOGIC_VECTOR (3 downto 0);
23 signal R4 : STD_LOGIC_VECTOR (3 downto 0);
24 signal R5 : STD_LOGIC_VECTOR (3 downto 0);
25 signal R6 : STD_LOGIC_VECTOR (3 downto 0);
26 signal R7 : STD_LOGIC_VECTOR (3 downto 0);
27
28 begin
29
30 UUT: Reg_Bank
31 port map(
32    D => D,
33    I => I,
34    Clk => Clk,
35    CLR => CLR,
36    R0 => R0,
37    R1 => R1,
38    R2 => R2,
39    R3 => R3,
40    R4 => R4,
41    R5 => R5,
42    R6 => R6,
43    R7 => R7
44 );
45
46
47 process
48 begin
49
50    clk<='1';
51    wait for 10 ns;
52
53    clk<='0';
54    wait for 10 ns;
55
56 end process;
57
58
59 process
60 begin
61
62    CLR <= '1';
63    D <= "0101";
64    I <= "010";
65
66    wait for 150ns;
67    CLR <= '0';
68    D <= "0011";
69    I <= "011";
70
71    wait for 150ns;
72    D <= "1101";
73    I <= "100";
74
75    wait for 150 ns;
76    D <= "0111";
77    I <= "101";
78
79    wait for 150 ns;
80    D <= "1011";
81    I <= "111";
82    wait;
83
84
85 end process;
86
87
88 end Behavioral;

```

4. Timing Diagram



8-to-1 Multiplexer

2_to_4 decoder

2-to-4 decoder takes a 2-bit input and enables one of four output lines based on the input combination, allowing for the selection of one out of four possible outputs. The EN (enable) signal controls whether the decoder is active or not. When the enable signal (EN) is 1, the selected output line becomes active (1), while the others remain inactive (0).

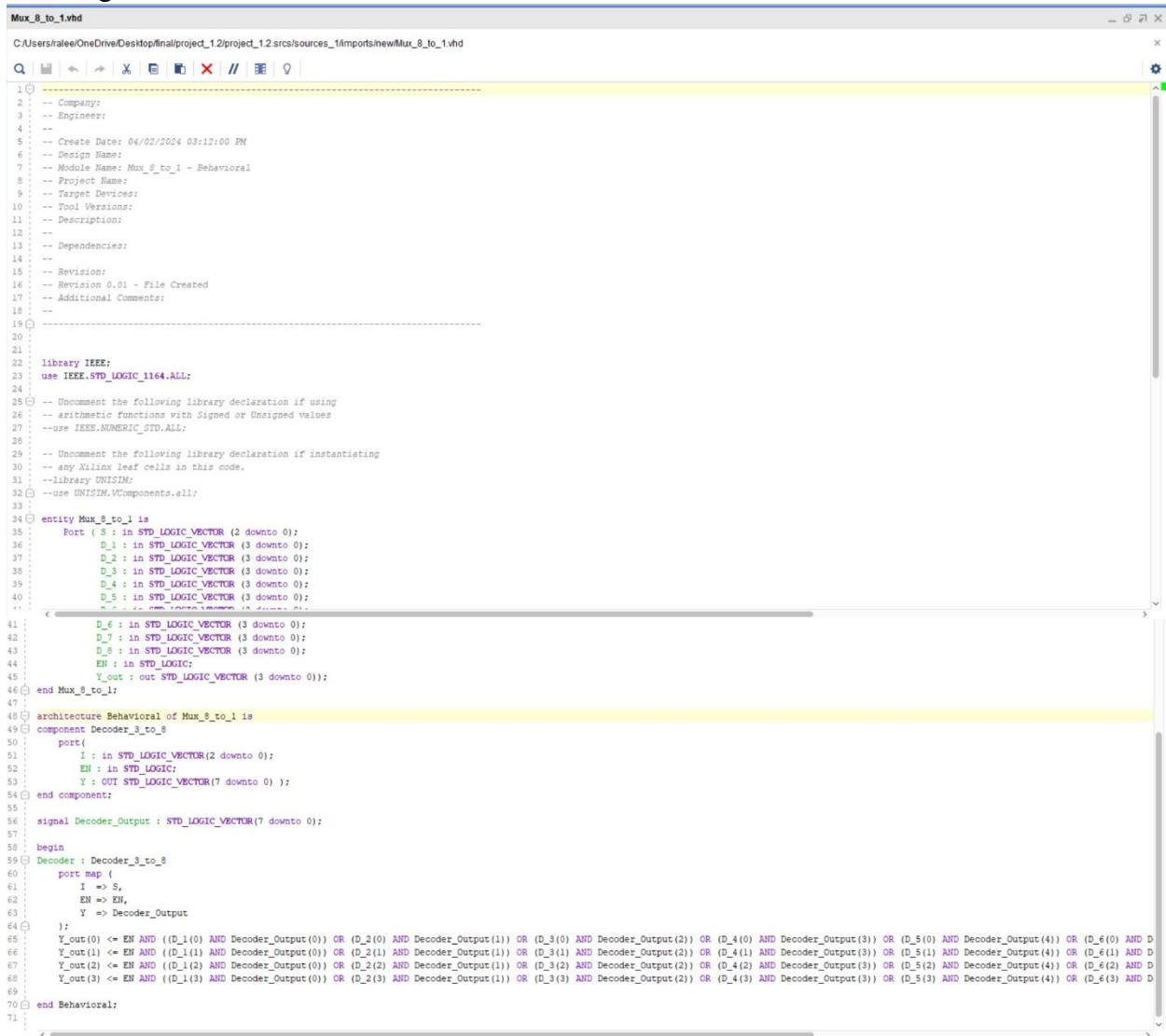
3_to_8 decoder

This entity has three inputs (I) and one enable signal (EN), producing eight outputs (Y). It uses two 2 to 4 decoders. The first 2-to-4 decoder (connected to I(0)) is active when I(2) is 0, and the second one (connected to I(1)) is active when I(2) is 1. The outputs of the two decoders are concatenated to form the 8-bit output Y, with the first four bits coming from the first decoder and the last four bits from the second decoder. The enable signal (EN) controls whether the decoders are active or not.

8-way-4-bit multiplexer

We designed a 8-way 4-bit multiplexer to select one of four 4-bit numbers. The design utilizes a separate component called Decoder_3_to_8. To choose one number, I provide a 3-bit selection signal (S) to the multiplexer. Decoder_3_to_8 decodes the 'S' into eight individual control signals for selecting the inputs from D_1 to D_8.

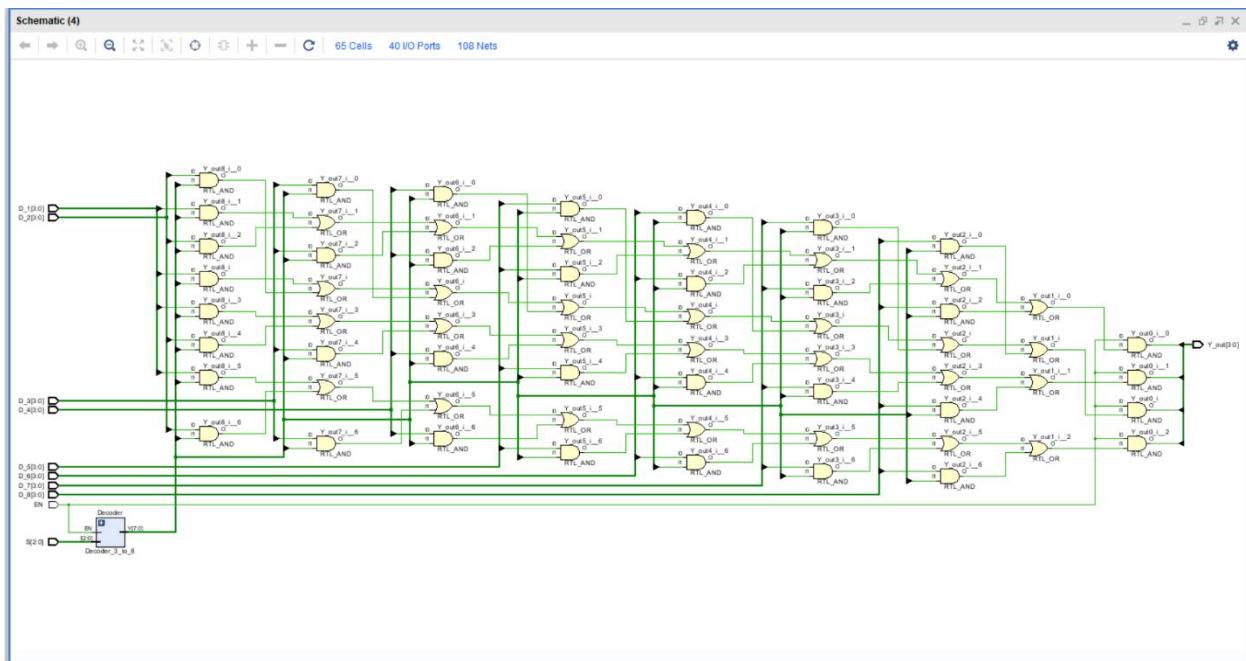
1. Design source file



The screenshot shows a VHDL code editor window titled "Mux_8_to_1.vhd". The code is a behavioral design for a 8-to-1 multiplexer. It includes comments, declarations for IEEE library, component Decoder_3_to_8, and an architecture Behavioral. The architecture details the port map and the logic for generating 7 output lines (Y_out(0) to Y_out(6)) based on the input vector I and enable signal EN.

```
1 -- Company:
2 -- Engineer:
3 --
4 --
5 -- Create Date: 04/02/2024 03:12:00 PM
6 -- Design Name:
7 -- Module Name: Mux_8_to_1 - Behavioral
8 -- Project Name:
9 -- Target Devices:
10 -- Tool Versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19
20
21
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24
25 -- Uncomment the following library declaration if using
26 -- arithmetic functions with Signed or Unsigned values
27 --use IEEE.NUMERIC_STD.ALL;
28
29 -- Uncomment the following library declaration if instantiating
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33
34 entity Mux_8_to_1 is
35     Port ( I : in STD.LOGIC_VECTOR (2 downto 0);
36             D_1 : in STD.LOGIC_VECTOR (3 downto 0);
37             D_2 : in STD.LOGIC_VECTOR (3 downto 0);
38             D_3 : in STD.LOGIC_VECTOR (3 downto 0);
39             D_4 : in STD.LOGIC_VECTOR (3 downto 0);
40             D_5 : in STD.LOGIC_VECTOR (3 downto 0);
41             D_6 : in STD.LOGIC_VECTOR (3 downto 0);
42             D_7 : in STD.LOGIC_VECTOR (3 downto 0);
43             D_8 : in STD.LOGIC_VECTOR (3 downto 0);
44             EN : in STD.LOGIC;
45             Y_out : out STD.LOGIC_VECTOR (3 downto 0));
46 end Mux_8_to_1;
47
48 architecture Behavioral of Mux_8_to_1 is
49 component Decoder_3_to_8
50     port(
51         I : in STD.LOGIC_VECTOR(2 downto 0);
52         EN : in STD.LOGIC;
53         Y : out STD.LOGIC_VECTOR(7 downto 0) );
54 end component;
55
56 signal Decoder_Output : STD.LOGIC_VECTOR(7 downto 0);
57
58 begin
59     Decoder : Decoder_3_to_8
60     port map (
61         I => S,
62         EN => EN,
63         Y => Decoder_Output
64     );
65     Y_out(0) <- EN AND ((D_1(0) AND Decoder_Output(0)) OR (D_2(0) AND Decoder_Output(1)) OR (D_3(0) AND Decoder_Output(2)) OR (D_4(0) AND Decoder_Output(3)) OR (D_5(0) AND Decoder_Output(4)) OR (D_6(0) AND D
66     Y_out(1) <- EN AND ((D_1(1) AND Decoder_Output(0)) OR (D_2(1) AND Decoder_Output(1)) OR (D_3(1) AND Decoder_Output(2)) OR (D_4(1) AND Decoder_Output(3)) OR (D_5(1) AND Decoder_Output(4)) OR (D_6(1) AND D
67     Y_out(2) <- EN AND ((D_1(2) AND Decoder_Output(0)) OR (D_2(2) AND Decoder_Output(1)) OR (D_3(2) AND Decoder_Output(2)) OR (D_4(2) AND Decoder_Output(3)) OR (D_5(2) AND Decoder_Output(4)) OR (D_6(2) AND D
68     Y_out(3) <- EN AND ((D_1(3) AND Decoder_Output(0)) OR (D_2(3) AND Decoder_Output(1)) OR (D_3(3) AND Decoder_Output(2)) OR (D_4(3) AND Decoder_Output(3)) OR (D_5(3) AND Decoder_Output(4)) OR (D_6(3) AND D
69
70 end Behavioral;
71
```

2. Elaborated design schematic



3. Simulation source file

```
TB_Mux_8_to_1.vhd
C:/Users/raleel/OneDrive/Desktop/final/project_1.2/project_1.2.srs/sim_1/imports/nana_pro/8-way-4-bit-multiplexer/8-way-4-bit-multiplexer/project_1.srs/sim_1/new/TB_Mux_8_to_1.vhd

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
entity TB_Mux_8_to_1 is
    Port ();
end TB_Mux_8_to_1;
architecture Behavioral of TB_Mux_8_to_1 is
component Mux_8_to_1 is
    --> MUX_8_TO_1 - 8-INPUT, 1-OUTPUT, 3-BIT SELECT
end component;
```

```

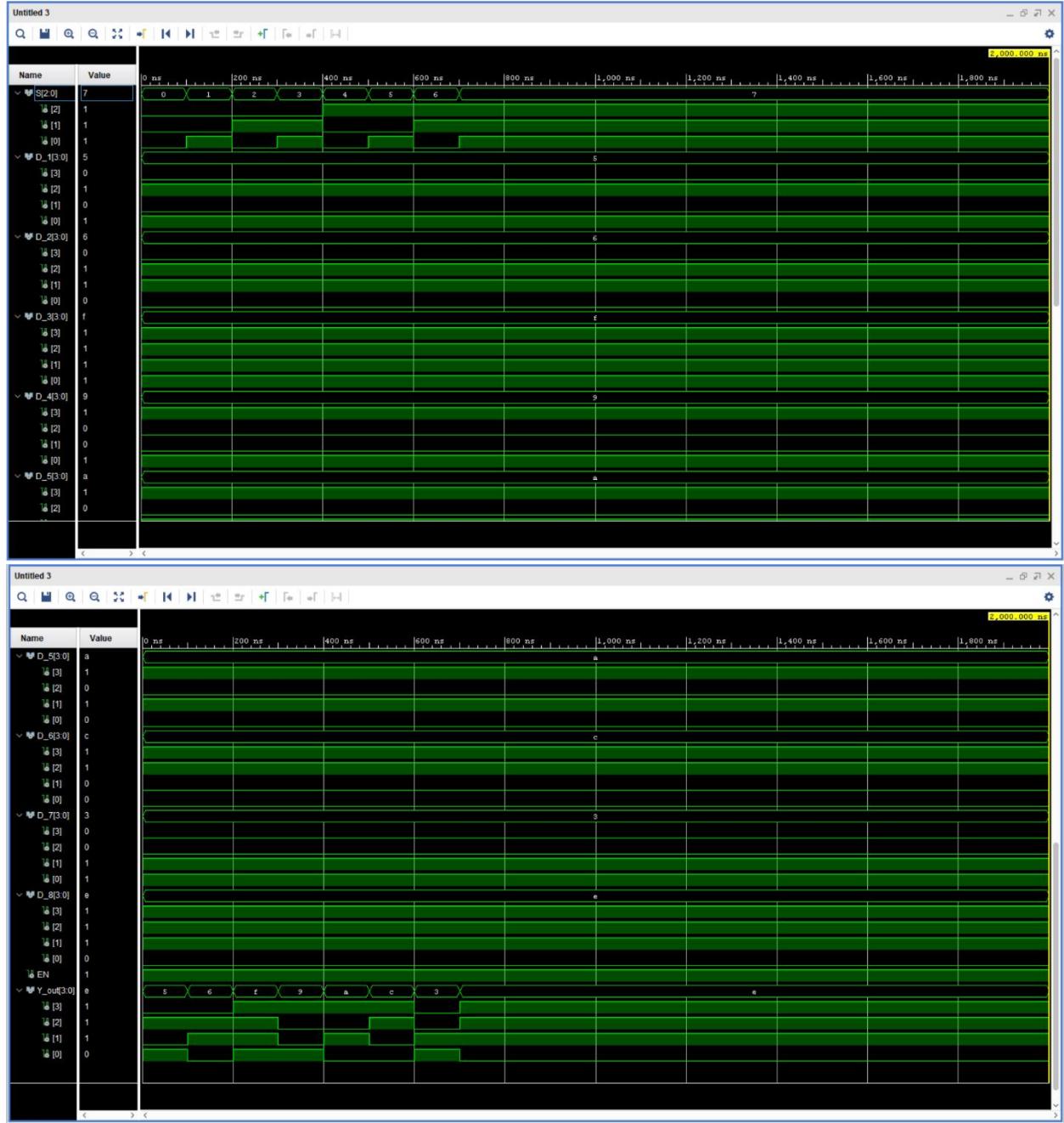
TB_Mux_8_to_1.vhd
C:/Users/raleel/OneDrive/Desktop/final/project_12/project_12.srcs/sim_1/imports/nana_pro/8-way-4-bit-multiplexer/8-way-4-bit-multiplexer/project_1.srcs/sim_1/new/TB_Mux_8_to_1.vhd

Q | H | ← | → | X | // | ■ | ♡ |  |

40 component Mux_8_to_1 is
41     Port ( S : in STD_LOGIC_VECTOR (2 downto 0);
42             D_1 : in STD_LOGIC_VECTOR (3 downto 0);
43             D_2 : in STD_LOGIC_VECTOR (3 downto 0);
44             D_3 : in STD_LOGIC_VECTOR (3 downto 0);
45             D_4 : in STD_LOGIC_VECTOR (3 downto 0);
46             D_5 : in STD_LOGIC_VECTOR (3 downto 0);
47             D_6 : in STD_LOGIC_VECTOR (3 downto 0);
48             D_7 : in STD_LOGIC_VECTOR (3 downto 0);
49             D_8 : in STD_LOGIC_VECTOR (3 downto 0);
50             EN : in STD_LOGIC;
51             Y_out : out STD_LOGIC_VECTOR (3 downto 0));
52 end component;
53
54 signal S : STD_LOGIC_VECTOR (2 downto 0);
55 signal D_1 : STD_LOGIC_VECTOR (3 downto 0);
56 signal D_2 : STD_LOGIC_VECTOR (3 downto 0);
57 signal D_3 : STD_LOGIC_VECTOR (3 downto 0);
58 signal D_4 : STD_LOGIC_VECTOR (3 downto 0);
59 signal D_5 : STD_LOGIC_VECTOR (3 downto 0);
60 signal D_6 : STD_LOGIC_VECTOR (3 downto 0);
61 signal D_7 : STD_LOGIC_VECTOR (3 downto 0);
62 signal D_8 : STD_LOGIC_VECTOR (3 downto 0);
63 signal EN : STD_LOGIC;
64 signal Y_out : STD_LOGIC_VECTOR (3 downto 0);
65
66 begin
67
68 UUT : Mux_8_to_1
69     PORT MAP(
70         S => S,
71         D_1 => D_1,
72         D_2 => D_2,
73         D_3 => D_3,
74         D_4 => D_4,
75         D_5 => D_5,
76         D_6 => D_6,
77         D_7 => D_7,
78         D_8 => D_8,
79         EN => EN,
80         Y_out => Y_out
81 );
82
83 process
84 begin
85     EN <= '1';
86     D_1 <= "0101";
87     D_2 <= "0110";
88     D_3 <= "1111";
89     D_4 <= "1001";
90     D_5 <= "1010";
91     D_6 <= "1100";
92     D_7 <= "0011";
93     D_8 <= "1110";
94     S <= "000";
95
96     WAIT FOR 100 ns;
97     S <= "001";
98
99     WAIT FOR 100 ns;
100    S <= "010";
101
102    WAIT FOR 100 ns;
103    S <= "011";
104
105    WAIT FOR 100 ns;
106    S <= "100";
107
108    WAIT FOR 100 ns;
109    S <= "101";
110
111    WAIT FOR 100 ns;
112    S <= "110";
113
114    WAIT FOR 100 ns;
115    S <= "111";
116
117    WAIT;
118
119
120 end process;
121 end Behavioral;
122

```

4. Timing Diagram



Adder Subtractor

This component is responsible for all the additions and subtractions taken place in this processor. This is designed using a four-bit Ripple Carry Adder. This component takes two vectors which are subjected to the addition or subtraction and one logic to enable the component and another logic to indicate whether the calculation is an addition or subtraction. It returns the result as a vector, and it stores in a register. It is capable of doing calculations between -8 to 7. If the result exceeds

the given range, it will be depicted through the Overflow Flag. If the value is zero it will also be depicted through Zero Flag.

Here we have used the RCA which we designed in classroom. To indicate some functionalities some output terminals are added.

1. Design source file

```

4_bit_Adder_Substrator.vhd
C:/Users/rakeel/Desktop/final/final new/project_1_2/project_1_2.srsc/sources_1/imports/new4_bit_Adder_Substrator.vhd

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
library UNISIM;
--use UNISIM.VComponents.all;
entity Fourbit_Adder_Substrator is
    Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
           B : in STD_LOGIC_VECTOR (3 downto 0);
           S : out STD_LOGIC_VECTOR (3 downto 0);
           E : in STD_LOGIC;
           M : in STD_LOGIC;
           V : out STD_LOGIC;
           C_IN : in STD_LOGIC;
           S0 : out STD_LOGIC;
           S1 : out STD_LOGIC;
           S2 : out STD_LOGIC;
           S3 : out STD_LOGIC;
           C_OUT3 : out STD_LOGIC;
           C_OUT4 : out STD_LOGIC);
end entity;

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
library UNISIM;
--use UNISIM.VComponents.all;
entity Fourbit_Adder_Substrator is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           S : out STD_LOGIC;
           E : in STD_LOGIC;
           M : in STD_LOGIC;
           V : out STD_LOGIC;
           C_IN : in STD_LOGIC;
           S0 : out STD_LOGIC;
           S1 : out STD_LOGIC;
           S2 : out STD_LOGIC;
           S3 : out STD_LOGIC;
           C_OUT3 : out STD_LOGIC;
           C_OUT4 : out STD_LOGIC);
end entity;

architecture Behavioral of Fourbit_Adder_Substrator is
component RCA_4
    Port ( A0 : in STD_LOGIC;
           A1 : in STD_LOGIC;
           A2 : in STD_LOGIC;
           A3 : in STD_LOGIC;
           B0 : in STD_LOGIC;
           B1 : in STD_LOGIC;
           B2 : in STD_LOGIC;
           B3 : in STD_LOGIC;
           C_IN : in STD_LOGIC;
           S0 : out STD_LOGIC;
           S1 : out STD_LOGIC;
           S2 : out STD_LOGIC;
           S3 : out STD_LOGIC;
           C_OUT3 : out STD_LOGIC;
           C_OUT4 : out STD_LOGIC);
end component;
SIGNAL X,Y :STD_LOGIC;
SIGNAL BB, Temp: STD_LOGIC_VECTOR(3 DOWNTO 0);
begin
begin
    process (B, M)
    begin
        for i in 0 to B'length-1 loop
            BB(i) <= B(i) XOR M;
        end loop;
    end process;
    RCA_4_I : RCA_4
    PORT MAP (
        A0 => A(0),
        A1 => A(1),
        A2 => A(2),
        A3 => A(3),
        B0 => BB(0),
        B1 => BB(1),
        B2 => BB(2),
        B3 => BB(3),
        C_IN => C_IN,
        S0 => S0,
        S1 => S1,
        S2 => S2,
        S3 => S3,
        C_OUT3 => C_OUT3,
        C_OUT4 => C_OUT4
    );
end;

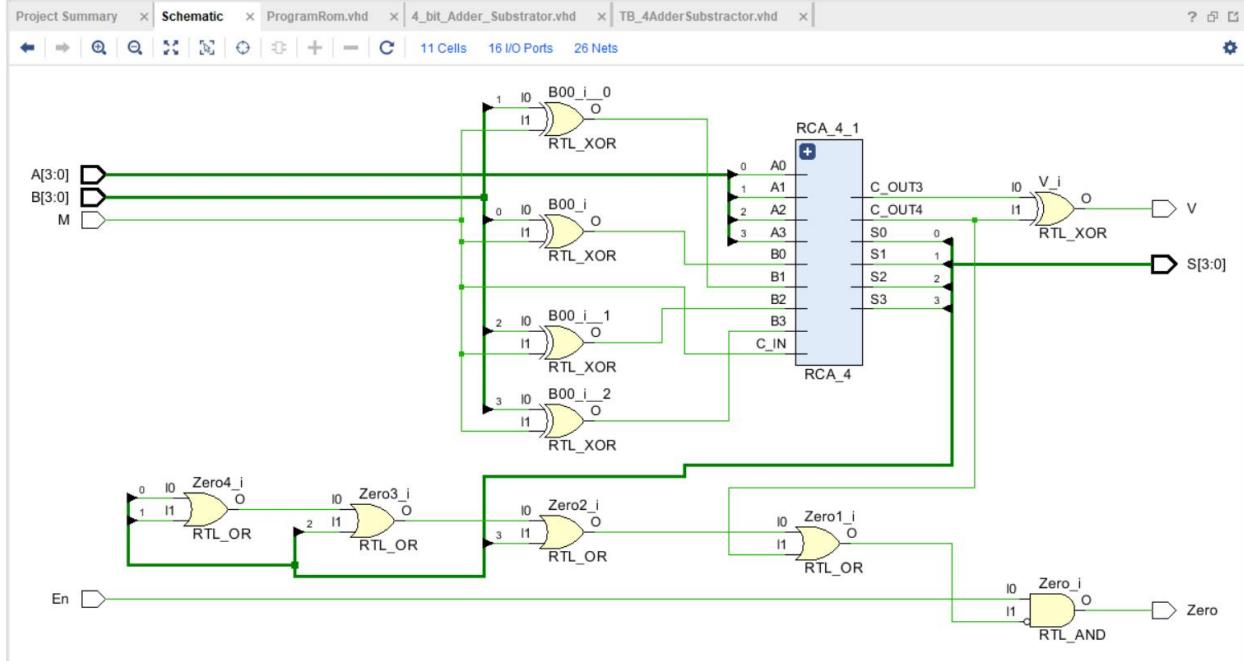
```

```

79 : A0 => A(0),
80 : A1 => A(1),
81 : A2 => A(2),
82 : A3 => A(3),
83 : B0 => BB(0),
84 : B1 => BB(1),
85 : B2 => BB(2),
86 : B3 => BB(3),
87 : C_IN => M,
88 : S0 => Temp(0),
89 : S1 => Temp(1),
90 : S2 => Temp(2),
91 : S3 => Temp(3),
92 : C_OUT3 => X,
93 : C_OUT4 => Y
94 : );
95 :
96 : S <= Temp;
97 : V <= (X XOR Y);
98 : Zero <= En AND NOT(Temp(0) OR Temp(1) OR Temp(2) OR Temp(3) OR Y);
99 :
100 end Behavioral;
101

```

2. Elaborated design schematic



3. Simulation source file

TB_4AdderSubtractor.vhd

```

C:/Users/raleel/OneDrive/Desktop/final/final new/project_1.2/project_1.2/srcs/sim_1/new/TB_4AdderSubtractor.vhd

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
entity TB_AdderSubtractor is
  Port ( );
end TB_AdderSubtractor;
architecture Behavioral of TB_AdderSubtractor is
component Fourbit_Adder_Subtractor
  Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
        B : in STD_LOGIC_VECTOR (3 downto 0);
        S : out STD_LOGIC_VECTOR (3 downto 0);
        M : in STD_LOGIC;
        En : in STD_LOGIC;
        V : out STD_LOGIC;
        Zero : out STD_LOGIC
      );
end component;
begin
  mvt : Fourbit_Adder_Subtractor
  port map (
    A => a,
    B => b,
    En =>en,
    M => m,
    V => v,
    S => s,
    Zero => zero
  );
  process begin
    a <= "0000";
    b <= "0001";
    m <= '0';
    en <= '1';
    wait for 100ns;
    a <= "0110";
    b <= "1001";
    m <= '1';
    en <= '0';
    wait for 100ns;
    a <= "0100";
    b <= "0000";
    end process;
  
```

TB_4AdderSubtractor.vhd

```

C:/Users/raleel/OneDrive/Desktop/final/final new/project_1.2/project_1.2/srcs/sim_1/new/TB_4AdderSubtractor.vhd

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
entity TB_AdderSubtractor is
  Port ( );
end TB_AdderSubtractor;
architecture Behavioral of TB_AdderSubtractor is
component Fourbit_Adder_Subtractor
  Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
        B : in STD_LOGIC_VECTOR (3 downto 0);
        S : out STD_LOGIC_VECTOR (3 downto 0);
        M : in STD_LOGIC;
        En : in STD_LOGIC;
        V : out STD_LOGIC;
        Zero : out STD_LOGIC
      );
end component;
begin
  mvt : Fourbit_Adder_Subtractor
  port map (
    A => a,
    B => b,
    En =>en,
    M => m,
    V => v,
    S => s,
    Zero => zero
  );
  process begin
    a <= "0000";
    b <= "0001";
    m <= '0';
    en <= '1';
    wait for 100ns;
    a <= "0110";
    b <= "1001";
    m <= '1';
    en <= '0';
    wait for 100ns;
    a <= "0100";
    b <= "0000";
    end process;
  
```

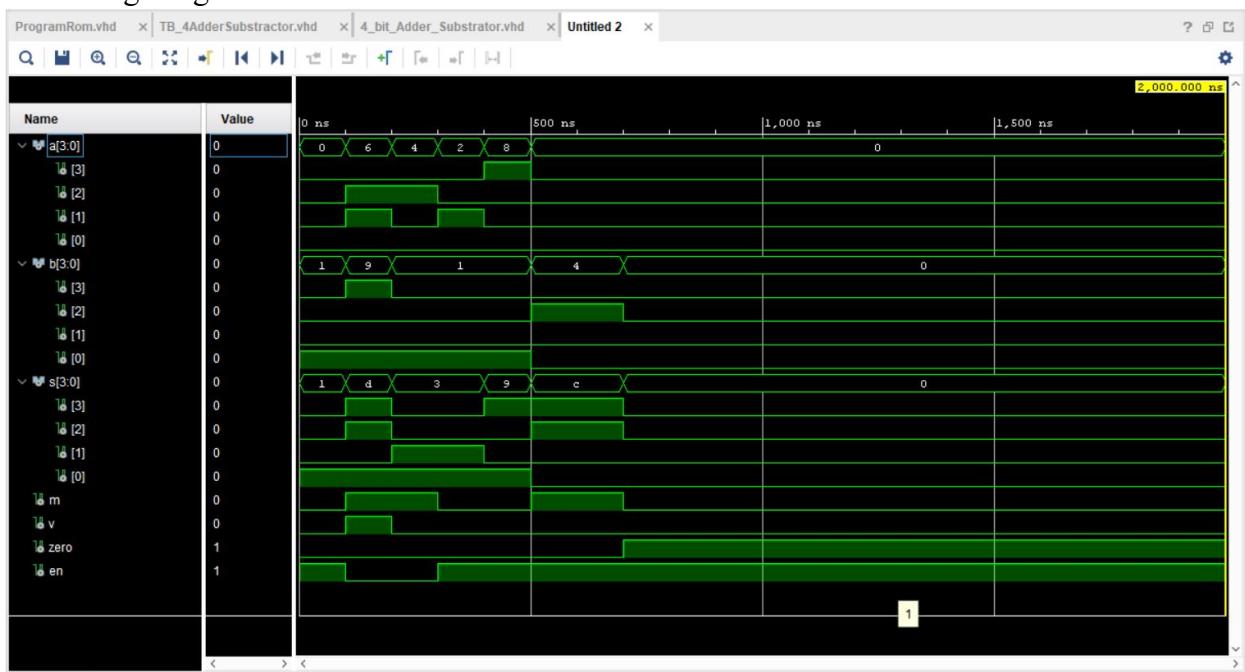
```

TB_4AdderSubtractor.vhd
C:\Users\raees\OneDrive\Desktop\final\final\new\project_1.2\project_1.2.srsc\sim_1\newTB_4AdderSubtractor.vhd

73 :    a <= "0110";
74 :    b <= "1001";
75 :    m <= '1';
76 :    en <= '0';
77 :    wait for 100ns;
78 :    a <= "0100";
79 :    b <= "0001";
80 :    m <= '1';
81 :    wait for 100ns;
82 :    a <= "0010";
83 :    b <= "0001";
84 :    m <= '0';
85 :    en <= '1';
86 :    wait for 100ns;
87 :
88 :    a <= "1000";
89 :    b <= "0001";
90 :    m <= '0';
91 :    wait for 100ns;
92 :    a <= "0000";
93 :    b <= "0100";
94 :    m <= '1';
95 :    wait for 100ns;
96 :
97 :    a <= "0000";
98 :    b <= "0100";
99 :    m <= '1';
100 :   wait for 100ns;
101 :  end process;
102 :  ...
103 :  a <= "0000";
104 :  b <= "0000";
105 :  m <= '0';
106 :  wait for 100ns;
107 :
108 :  wait;
109 : end process;
110 :
111 end Behavioral;
112

```

4. Timing Diagram



Slow Clock

1. Design source file

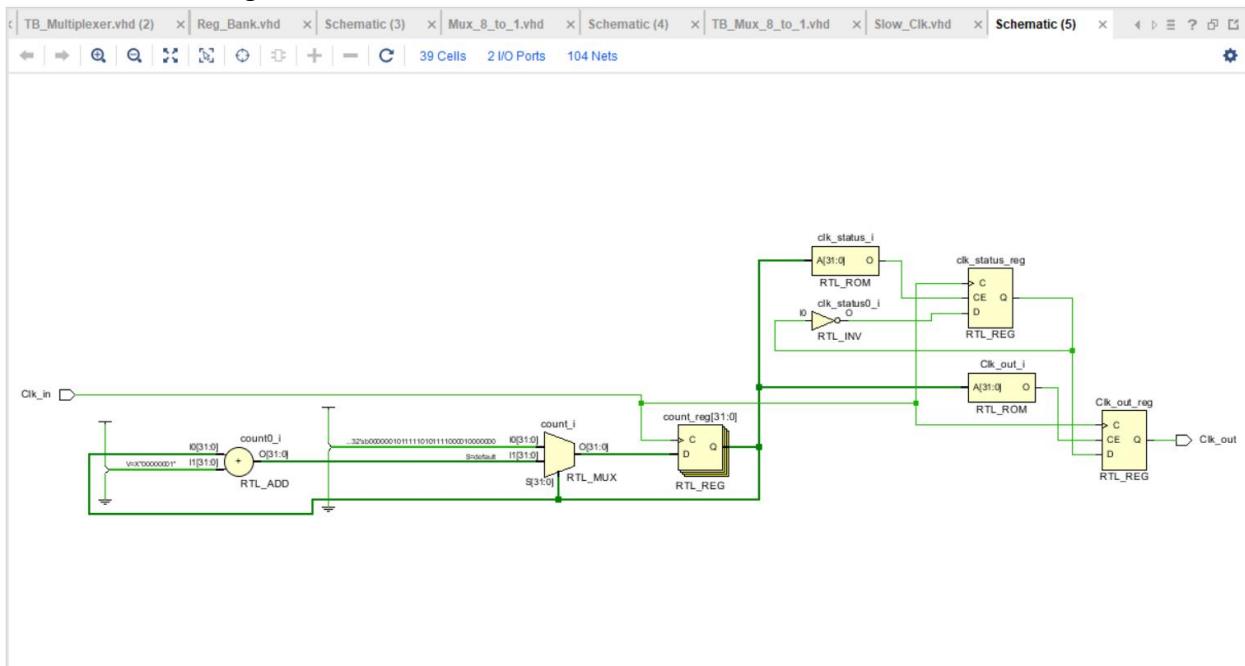
```

Slow_Clk.vhd
C:/Users/rakeel/Desktop/final/project_12/project_12.srcc/sources_1/imports/new/Slow_Clk.vhd

1 -->
2 -- Company:
3 -- Engineer:
4 --
5 -- Create Date: 03/05/2024 10:54:36 PM
6 -- Design Name:
7 -- Module Name: Slow_Clk - Behavioral
8 -- Project Name:
9 -- Target Device:
10 -- Tool Versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19
20
21
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24
25 -- Uncomment the following library declaration if using
26 -- arithmetic functions with Signed or Unsigned values
27 --use IEEE.NUMERIC_STD.ALL;
28
29 -- Uncomment the following library declaration if instantiating
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33
34 entity Slow_Clk is
35     Port ( Clk_in : in STD_LOGIC;
36             Clk_out : out STD_LOGIC);
37 end Slow_Clk;
38
39 architecture Behavioral of Slow_Clk is
40
41 begin
42
43     signal count : integer := 1;
44     signal clk_status : std_logic := '0';
45
46 begin
47
48     process (Clk_in) begin
49         if (rising_edge(Clk_in)) then
50             count <= count +1;
51             if (count = 50000000) then
52                 clk_status <= not clk_status;
53                 Clk_out<= clk_status;
54                 count<=1;
55             end if;
56         end if;
57     end process;
58
59 end Behavioral;
60

```

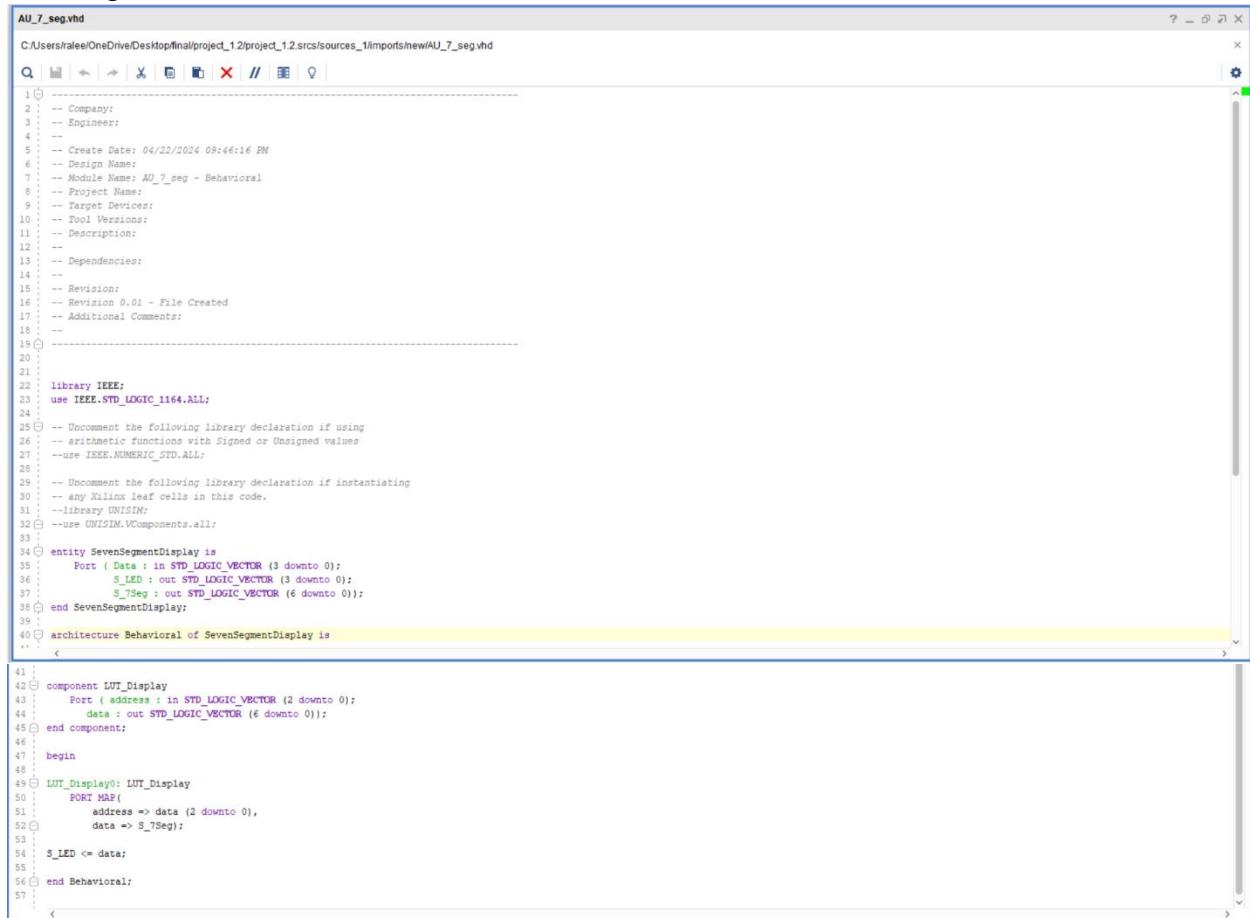
2. Elaborated design schematic



Seven Segment display

This has a 4-bit input and two separate outputs. This uses a component named "LUT_8_7," which acts as a Look-Up Table to translate the 3-bit binary input into a corresponding 7-bit pattern for the 7-segment display. The 4-bit input (Data), is directly mapped to the LED outputs, providing a direct representation of the input value. This module facilitates the visual representation of binary data on both a 7-segment display and individual LEDs simultaneously

1. Design source file



The screenshot shows a VHDL code editor window titled "AU_7_seg.vhd". The code is as follows:

```
AU_7_seg.vhd
C:/Users/rakeel/OneDrive/Desktop/final/project_1.2/project_1.2.srsc/sources_1/imports/new/AU_7_seg.vhd

1 -- Company;
2 -- Engineer;
3 --
4 -- Create Date: 04/23/2024 09:46:16 PM
5 -- Design Name:
6 -- Module Name: AU_7_seg - Behavioral
7 -- Project Name:
8 -- Target Devices:
9 -- Tool Versions:
10 -- Description:
11 -- Dependencies:
12 --
13 -- Revision:
14 -- Revision 0.01 - File Created
15 -- Additional Comments:
16 --
17
18
19
20
21
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24
25 -- Uncomment the following library declaration if using
26 -- arithmetic functions with Signed or Unsigned values
27 --use IEEE.NUMERIC_STD.ALL;
28
29 -- Uncomment the following library declaration if instantiating
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33
34 entity SevenSegmentDisplay is
35     Port ( Data : in STD_LOGIC_VECTOR (3 downto 0);
36             S_LED : out STD_LOGIC_VECTOR (3 downto 0);
37             S_7Seg : out STD_LOGIC_VECTOR (6 downto 0));
38 end SevenSegmentDisplay;
39
40 architecture Behavioral of SevenSegmentDisplay is
41     begin
42         component LUT_Display
43             Port ( address : in STD_LOGIC_VECTOR (2 downto 0);
44                   data : out STD_LOGIC_VECTOR (6 downto 0));
45         end component;
46
47         begin
48             LUT_Display0: LUT_Display
49                 PORT MAP(
50                     address => data (2 downto 0),
51                     data => S_7Seg);
52
53             S_LED <= data;
54
55         end Behavioral;
56
57 
```

2. Lookup table

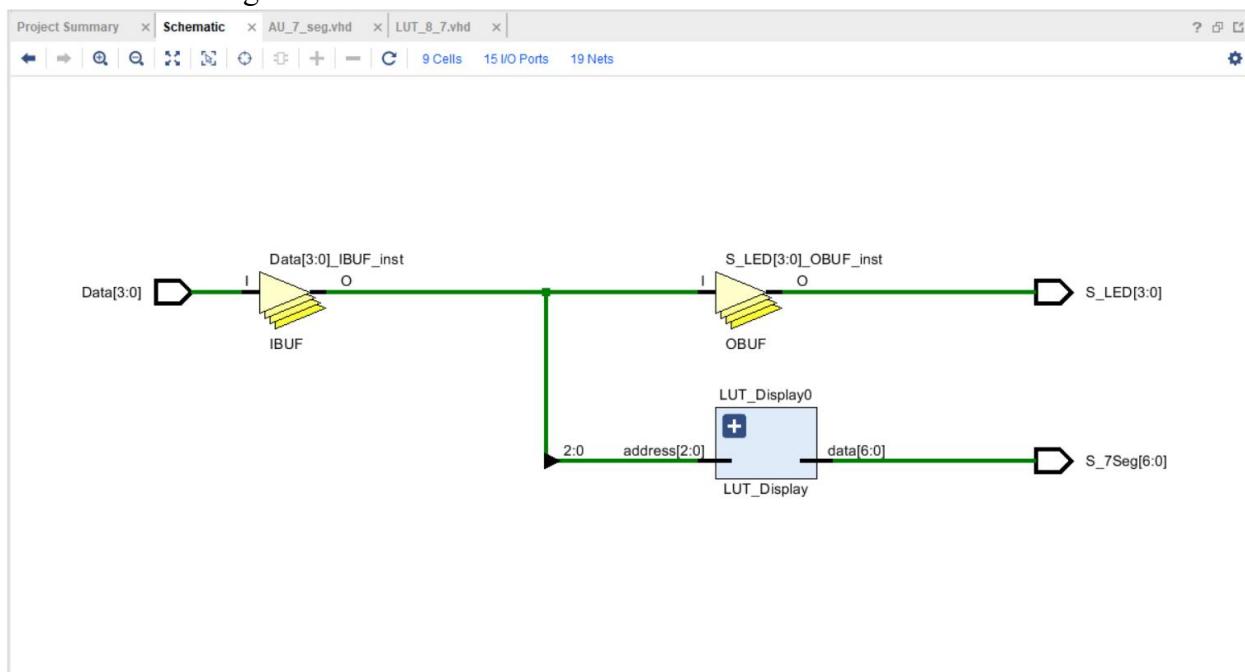
A lookup table is used to find the mapping between inputs S2 - S0 and outputs from 0 to 7. As the mapping will not change with time, we can save it in a ROM. I had to instruct the ROM from which location I need to send the output to 7- segment display. As there are 8 memory locations in ROM, it needs a 3-bits to address each location uniquely.

```

LUT_8_7.vhd
C:/Users/raleel/OneDrive/Desktop/final/project_12/project_12.srcs/sources_1/imports/new/LUT_8_7.vhd

1 -- Company:
2 -- Engineer:
3 --
4 --
5 -- Design Name: 04/22/2024 09:13:03 PM
6 -- Module Name: LUT_8_7 - Behavioral
7 -- Project Name:
8 -- Target Devices:
9 -- Tool Versions:
10 -- Description:
11 --
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 
20 
21 
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24 use ieee.numeric_std.all;
25 
26 -- Uncomment the following library declaration if using
27 -- arithmetic functions with Signed or Unsigned values
28 --use IEEE.NUMERIC_STD.ALL;
29 
30 -- Uncomment the following library declaration if instantiating
31 -- any Xilinx leaf cells in this code.
32 --library UNISIM;
33 --use UNISIM.VComponents.all;
34 
35 entity LUT_Display is
36     Port ( address : in STD_LOGIC_VECTOR (2 downto 0);
37            data : out STD_LOGIC_VECTOR (6 downto 0));
38 end LUT_Display;
39 
40 architecture Behavioral of LUT_Display is
41 
42 type rom_type is array (0 to 7) of std_logic_vector(6 downto 0);
43 
44 signal sevenSegment_ROM : rom_type := (
45     "1000000", -- 0
46     "1110001", -- 1
47     "0100100", -- 2
48     "0110000", -- 3
49     "0011001", -- 4
50     "0010010", -- 5
51     "0000010", -- 6
52     "1111000"); -- 7
53 
54 begin
55 
56 data <= sevenSegment_ROM(to_integer(unsigned(address)));
57 
58 end Behavioral;
59 
```

3. Elaborated design schematic

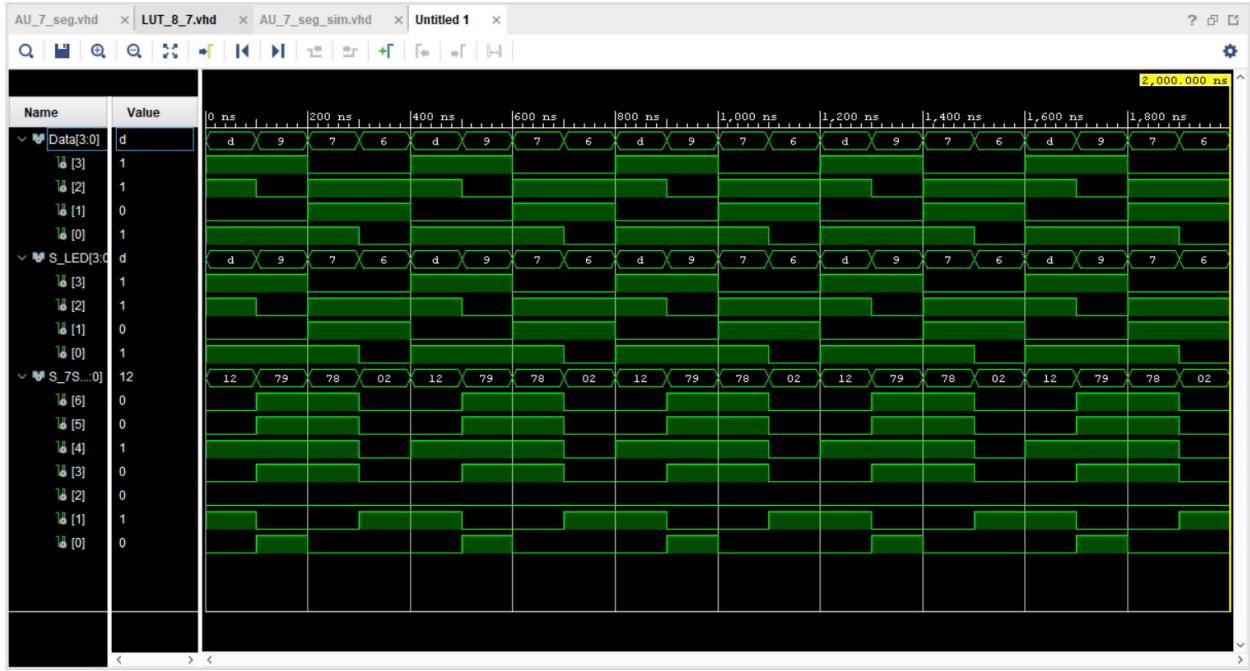


4. Simulation source file

```
AU_7_seg_sim.vhd
C:/Users/alee/Desktop/final/project_1_2/project_1_2.srsc/sim_1/imports/newAU_7_seg_sim.vhd

1 --
2 -- Company:
3 -- Engineer:
4 --
5 -- Create Date: 04/22/2024 09:55:57 PM
6 -- Design Name:
7 -- Module Name: AU_7_seg_sim - Behavioral
8 -- Project Name:
9 -- Target Devices:
10 -- Tool Versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 --
20
21
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24
25 -- Uncomment the following library declaration if using
26 -- arithmetic functions with Signed or Unsigned values
27 --use IEEE.NUMERIC_STD.ALL;
28
29 -- Uncomment the following library declaration if instantiating
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33
34 entity TB_SevenSegmentDisplay is
35   -- Port ( );
36 end TB_SevenSegmentDisplay;
37
38 architecture Behavioral of TB_SevenSegmentDisplay is
39
40 COMPONENT SevenSegmentDisplay is
41   Port ( Data : in STD_LOGIC_VECTOR (3 downto 0);
42         S_LED : out STD_LOGIC_VECTOR (3 downto 0);
43         S_7Seg : out STD_LOGIC_VECTOR (6 downto 0);
44   end COMPONENT;
45
46 SIGNAL Data,S_LED : std_logic_vector (3 downto 0);
47 SIGNAL S_7Seg : std_logic_vector (6 downto 0);
48
49 begin
50
51   UUT: SevenSegmentDisplay PORT MAP(
52     Data => Data,
53     S_LED => S_LED,
54     S_7Seg => S_7Seg
55   );
56
57 process
58 begin
59   Data <= "1101";
60   wait for 100 ns;
61
62   Data <= "1001";
63   wait for 100 ns;
64
65   Data <= "0111";
66   wait for 100 ns;
67
68   Data <= "0110";
69   wait for 100 ns;
70
71   -- WAIT: -- WILL WAIT FOREVER
72 end process;
73
74 end Behavioral;
75
```

5. Timing Diagram



Final Nano processor

Final nano-processor is designed using connecting all these entities. The program ROM is used to store the relevant machine instructions that needed to be run. The LEDs and the seven segment display will output the values in the R7 register. There are two flags used to output the state. One is the zero flag which will become high when the output of the adder subtractor is 0. The other flag is the overflow flag which will become high when the output of the adder subtractor unit is higher than 7 or lower than -8. In the modified version there are three more flags which is connected to the comparator indicating whether a the output of the comparator is “greater than”, “less than” or equal.

1. Design source file

NanoProcessor.vhd

```

C:/Users/raleel/OneDrive/Desktop/final/final new/project_1_2/project_1_2/srcs/sources_1/newNanoProcessor.vhd

1 -- Company:
2 -- Engineer:
3 -- 
4 -- Design Date: 04/11/2024 07:41:34 AM
5 -- Design Name:
6 -- Module Name: NanoProcessor - Behavioral
7 -- Project Name:
8 -- Target Devices:
9 -- Tool Versions:
10 -- Description:
11 -- 
12 -- Dependencies:
13 -- 
14 -- Revision:
15 -- Revision 0.01 - File Created
16 -- Additional Comments:
17 -- 
18 -- 
19 
20 
21 
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24 
25 -- Uncomment the following library declaration if using
26 -- arithmetic functions with Signed or Unsigned values
27 --use IEEE.NUMERIC_STD.ALL;
28 
29 -- Uncomment the following library declaration if instantiating
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33 
34 entity NanoProcessor is
35     Port ( Reset : in STD_LOGIC;
36             Clk : in STD_LOGIC;
37             Overflow : out STD_LOGIC;
38             Zero : out STD_LOGIC;
39             RT_LED : out STD_LOGIC_VECTOR (3 downto 0);
40             RT_Display : out STD_LOGIC_VECTOR(6 downto 0);
41             Anode : out STD_LOGIC_VECTOR(3 downto 0));
42 
43 end NanoProcessor;
44 
45 architecture Behavioral of NanoProcessor is
46 
47 component ProgramROM
48     Port ( address : in STD_LOGIC_VECTOR (2 downto 0);
49             data : out STD_LOGIC_VECTOR (11 downto 0));
50 end component;
51 
52 component ProgramCounter
53     Port ( D : in STD_LOGIC_VECTOR (2 downto 0);
54             Clk : in STD_LOGIC;
55             CLR : in STD_LOGIC;
56             Q : out STD_LOGIC_VECTOR (2 downto 0));
57 end component;
58 
59 component ThreeBitAdder
60     Port ( A : in STD_LOGIC_VECTOR (2 downto 0);
61             S : out STD_LOGIC_VECTOR (2 downto 0));
62 end component;
63 
64 component Slow_CLK
65     Port ( Clk_in : in STD_LOGIC;
66             Clk_out : out STD_LOGIC);
67 end component;
68 
69 component Two_3_bit_MUX
70     Port ( D1 : in STD_LOGIC_VECTOR (2 downto 0);
71             D2 : in STD_LOGIC_VECTOR (2 downto 0);
72             EN : in STD_LOGIC;
73             S : in STD_LOGIC;
74             Y : out STD_LOGIC_VECTOR (2 downto 0));
75 end component;
76 
77 component Instruction_Dec
78     Port ( OpCode: in STD_LOGIC_VECTOR(11 downto 0);
79             JumpRegisterValue: in STD_LOGIC_VECTOR(3 downto 0);
80             );
81 end component;

```

NanoProcessor.vhd

```

C:/Users/raleel/OneDrive/Desktop/final/final new/project_1_2/project_1_2/srcs/sources_1/newNanoProcessor.vhd

1 RT_Display: out STD_LOGIC_VECTOR(6 downto 0);
2 Anode : out STD_LOGIC_VECTOR(3 downto 0));
3 
4 end NanoProcessor;
5 
6 architecture Behavioral of NanoProcessor is
7 
8 component ProgramROM
9     Port ( address : in STD_LOGIC_VECTOR (2 downto 0);
10            data : out STD_LOGIC_VECTOR (11 downto 0));
11 end component;
12 
13 component ProgramCounter
14     Port ( D : in STD_LOGIC_VECTOR (2 downto 0);
15             Clk : in STD_LOGIC;
16             CLR : in STD_LOGIC;
17             Q : out STD_LOGIC_VECTOR (2 downto 0));
18 end component;
19 
20 component ThreeBitAdder
21     Port ( A : in STD_LOGIC_VECTOR (2 downto 0);
22             S : out STD_LOGIC_VECTOR (2 downto 0));
23 end component;
24 
25 component Slow_CLK
26     Port ( Clk_in : in STD_LOGIC;
27             Clk_out : out STD_LOGIC);
28 end component;
29 
30 component Two_3_bit_MUX
31     Port ( D1 : in STD_LOGIC_VECTOR (2 downto 0);
32             D2 : in STD_LOGIC_VECTOR (2 downto 0);
33             EN : in STD_LOGIC;
34             S : in STD_LOGIC;
35             Y : out STD_LOGIC_VECTOR (2 downto 0));
36 end component;
37 
38 component Instruction_Dec
39     Port ( OpCode: in STD_LOGIC_VECTOR(11 downto 0);
40             JumpRegisterValue: in STD_LOGIC_VECTOR(3 downto 0);
41             );
42 end component;

```

NanoProcessor.vhd

```
C:/Users/ralee/OneDrive/Desktop/finalfinal/new/project_1_2/project_1_2/srcs/sources_1/newNanoProcessor.vhd
```

Q | | | | | | | | | | | | |

```

79:     JumpRegisterValue: in STD_LOGIC_VECTOR(3 downto 0);
80:     JumpFlag: out STD_LOGIC;
81:     JumpAddress: Out STD_LOGIC_VECTOR(2 downto 0);
82:     AddOrSubtract : out STD_LOGIC;
83:     MuxEnable1: Out STD_LOGIC;
84:     MuxEnable2: Out STD_LOGIC;
85:     MuxControl1Signal1: Out STD_LOGIC_VECTOR(2 downto 0);
86:     MuxControl1Signal2: Out STD_LOGIC_VECTOR(2 downto 0);
87:     ImmediateValue : Out STD_LOGIC_VECTOR(3 downto 0);
88:     Load: out STD_LOGIC;
89:     RegisterEnable : Out STD_LOGIC_VECTOR(2 downto 0);
90:     AddSubtractEnable : Out STD_LOGIC;
91: end component;
92:
93: component Reg_Bank
94:   Port ( D : in STD_LOGIC_VECTOR (3 downto 0);
95:          I : in STD_LOGIC_VECTOR (2 downto 0);
96:          Clk : in STD_LOGIC;
97:          CLR : in STD_LOGIC;
98:          R0 : out STD_LOGIC_VECTOR (3 downto 0);
99:          R1 : out STD_LOGIC_VECTOR (3 downto 0);
100:         R2 : out STD_LOGIC_VECTOR (3 downto 0);
101:         R3 : out STD_LOGIC_VECTOR (3 downto 0);
102:         R4 : out STD_LOGIC_VECTOR (3 downto 0);
103:         R5 : out STD_LOGIC_VECTOR (3 downto 0);
104:         R6 : out STD_LOGIC_VECTOR (3 downto 0);
105:         R7 : out STD_LOGIC_VECTOR (3 downto 0));
106: end component;
107:
108: component Two_4bit_MUX
109:   Port ( D1 : in STD_LOGIC_VECTOR (3 downto 0);
110:          D2 : in STD_LOGIC_VECTOR (3 downto 0);
111:          S : in STD_LOGIC;
112:          EN : in STD_LOGIC;
113:          Y : out STD_LOGIC_VECTOR (3 downto 0));
114: end component;
115:
116: component Fourbit_Adder_Subtractor
117:   Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
118:          B : in STD_LOGIC_VECTOR (3 downto 0);
119:          C : out STD_LOGIC_VECTOR (3 downto 0));
120:
121:
122:
123:
124:
125:
126:
127: component Mux_8_to_1
128:   Port ( S : in STD_LOGIC_VECTOR (2 downto 0);
129:          D_1 : in STD_LOGIC_VECTOR (3 downto 0);
130:          D_2 : in STD_LOGIC_VECTOR (3 downto 0);
131:          D_3 : in STD_LOGIC_VECTOR (3 downto 0);
132:          D_4 : in STD_LOGIC_VECTOR (3 downto 0);
133:          D_5 : in STD_LOGIC_VECTOR (3 downto 0);
134:          D_6 : in STD_LOGIC_VECTOR (3 downto 0);
135:          D_7 : in STD_LOGIC_VECTOR (3 downto 0);
136:          D_8 : in STD_LOGIC_VECTOR (3 downto 0);
137:          E : in STD_LOGIC;
138:          Y_out : out STD_LOGIC_VECTOR (3 downto 0));
139: end component;
140:
141: component SevenSegmentDisplay
142:   Port ( Data : in STD_LOGIC_VECTOR (3 downto 0);
143:          S_LED : out STD_LOGIC_VECTOR (3 downto 0);
144:          S_7Seg : out STD_LOGIC_VECTOR (6 downto 0));
145: end component;
146:
147: signal PC, J_Address, Address, PC_in, sel1, sel2, enable: STD_LOGIC_VECTOR (2 downto 0);
148: signal Instruction: STD_LOGIC_VECTOR (11 downto 0);
149: signal Slow_Clock, J_Flag, operation, Load_from, Max_En1, Max_En2,AddSubEn: STD_LOGIC;
150: signal register1, register2, R_0, R_1, R_2, R_3, R_4, R_5, R_6, R_7, Value, result, data : STD_LOGIC_VECTOR (3 downto 0);
151:
152:
153: begin
154:   InstructionDecoder: Instruction_Dec
155:     port map (OpCode => Instruction,
156:               JumpRegisterValue =>register2,
157:               JumpFlag => J_Flag,
158:               JumpAddress => J_Address,
159:               ...
160: );
161: end;
```

NanoProcessor.vhd

```

C:/Users/ratee/Desktop/final/final new/project_1_2/project_1_2/srcs/sources_1/new/NanoProcessor.vhd

157 :     JumpAddress => J_Address,
158 :     AddrSubtract => operation,
159 :     NumEnabled1 => Mux_En1,
160 :     NumEnabled2 => Mux_En2,
161 :     NumControlSignals1 => Sel1,
162 :     NumControlSignals2 => Sel2,
163 :     ImmediateValue => Value,
164 :     Load => load_from,
165 :     RegisterIfEnable => enable,
166 :     AddSubtractEnable => AddSubEn);
167 :
168 :
169 : Clock0 : Slow_Clk
170 :     port map(
171 :         Clk_in => CLK,
172 :         Clk_out => Slow_Clock);
173 :
174 : ProgramCounter0 : ProgramCounter
175 :     port map(
176 :         D => PC_in,
177 :         Clk =>Slow_Clock,
178 :         CLR =>Reset,
179 :         Q =>PC);
180 :
181 : Adder0 :ThreeBitAdder
182 :     port map(
183 :         A => PC,
184 :         S => Address);
185 :
186 : ProgramRom : ProgramRom
187 :     port map(
188 :         address =>PC,
189 :         data =>Instruction);
190 :
191 : Two_3_MUX :Two_3_bit_MUX
192 :     port map(
193 :         DI =>Address,
194 :         DO =>J_Address,
195 :         EN =>'1',
196 :         S =>J_Flag,
197 :         Y =>FC_in);
198 :
199 : Two_4_MUX :Two_4bit_MUX
200 :     port map(
201 :         D1 =>result,
202 :         D2 =>Value,
203 :         EN =>'1',
204 :         S =>load_from,
205 :         Y => data);
206 :
207 : RegisterBank: Reg_Bank
208 :     port map(
209 :         I =>data,
210 :         I =>enable,
211 :         Clk =>Slow_Clock,
212 :         CLR =>Reset,
213 :         R0 =>R_1,
214 :         R1 =>R_1,
215 :         R2 =>R_2,
216 :         R3 =>R_3,
217 :         R4 =>R_4,
218 :         R5 =>R_5,
219 :         R6 =>R_6,
220 :         R7 =>R_7);
221 :
222 : Mux_8_to_4_0 :Mux_8_to_1
223 :     port map(
224 :         S =>sel1,
225 :         D_1 =>R_0,
226 :         D_2 =>R_1,
227 :         D_3 =>R_2,
228 :         D_4 =>R_3,
229 :         D_5 =>R_4,
230 :         D_6 =>R_5,
231 :         D_7 =>R_6,
232 :         D_8 =>R_7,
233 :         EN =>Mux_En1,
234 :         Y_out =>register2);
235 :

```

NanoProcessor.vhd

```

C:/Users/ratee/Desktop/final/final new/project_1_2/project_1_2/srcs/sources_1/new/NanoProcessor.vhd

196 :         S =>J_Flag,
197 :         Y =>FC_in;
198 :
199 : Two_4_MUX :Two_4bit_MUX
200 :     port map(
201 :         D1 =>result,
202 :         D2 =>Value,
203 :         EN =>'1',
204 :         S =>load_from,
205 :         Y => data);
206 :
207 : RegisterBank: Reg_Bank
208 :     port map(
209 :         I =>data,
210 :         I =>enable,
211 :         Clk =>Slow_Clock,
212 :         CLR =>Reset,
213 :         R0 =>R_1,
214 :         R1 =>R_1,
215 :         R2 =>R_2,
216 :         R3 =>R_3,
217 :         R4 =>R_4,
218 :         R5 =>R_5,
219 :         R6 =>R_6,
220 :         R7 =>R_7);
221 :
222 : Mux_8_to_4_0 :Mux_8_to_1
223 :     port map(
224 :         S =>sel1,
225 :         D_1 =>R_0,
226 :         D_2 =>R_1,
227 :         D_3 =>R_2,
228 :         D_4 =>R_3,
229 :         D_5 =>R_4,
230 :         D_6 =>R_5,
231 :         D_7 =>R_6,
232 :         D_8 =>R_7,
233 :         EN =>Mux_En1,
234 :         Y_out =>register2);
235 :

```

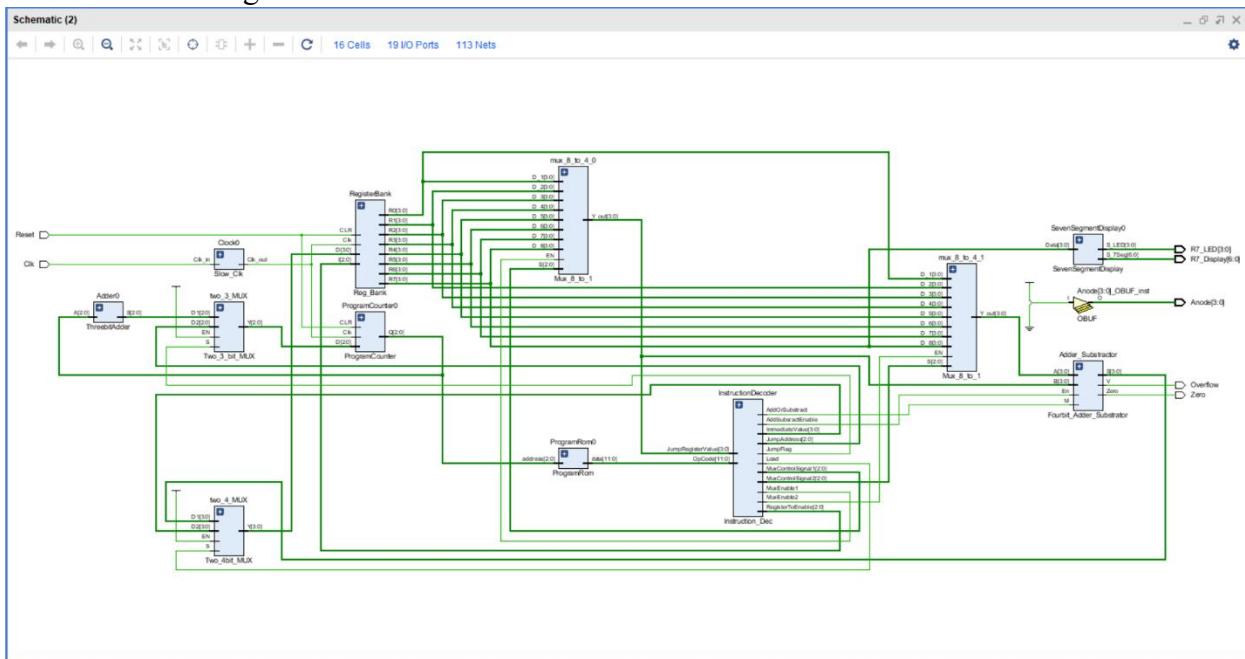
```

NanoProcessor.vhd
C:/Users/rakee/Desktop/final/final/new/project_1.2/project_1.2/srcs/sources_1/newNanoProcessor.vhd

230      D_6    =>R_5,
231      D_7    =>R_6,
232      D_8    =>R_7,
233      EN     =>Mux_En1,
234      Y_out  =>register2);
235
236 mux_8_to_4_1 :Mux_8_to_1
237   port map(
238     S      =>sel12,
239     D_0   =>R_0,
240     D_1   =>R_1,
241     D_2   =>R_2,
242     D_3   =>R_3,
243     D_4   =>R_4,
244     D_5   =>R_5,
245     D_6   =>R_6,
246     D_7   =>R_7,
247     EN    =>Mux_En2,
248     Y_out  =>register1);
249
250 Adder_Subtractor : Fourbit_Adder_Subtractor
251   port map(
252     A      =>register1,
253     B      =>register2,
254     S      =>result,
255     En    =>AddSubEn,
256     M     =>operation,
257     V     =>Overflow,
258     Zero  =>Zero);
259
260 SevenSegmentDisplay0 : SevenSegmentDisplay
261   port map(
262     Data   =>R_7,
263     S_LED  =>R7_LED,
264     S_7Seg =>R7_Display);
265
266 Anode <- "1110";
267
268 end Behavioral;
269

```

2. Elaborated design schematic



3. Simulation source file

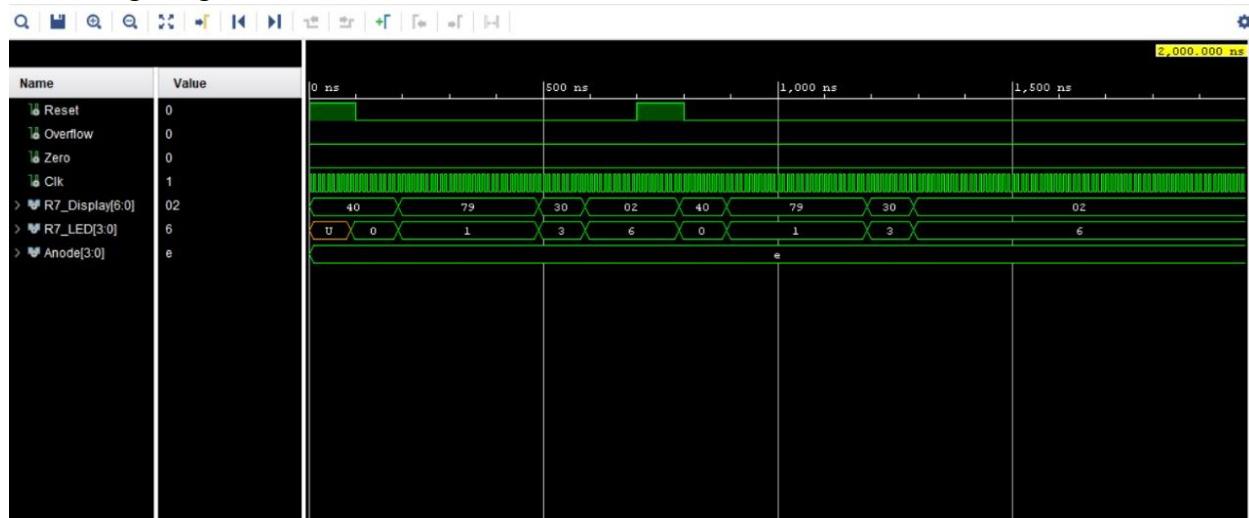
```

TB_NanoProcessor.vhd
C:/Users/raleelOneDrive/Desktop/final/final new/project_1.2/project_1.2/srcs/sim_1/new/TB_NanoProcessor.vhd

1 -- Testbench for NanoProcessor
2
3 library IEEE;
4 use IEEE.STD_LOGIC_1164.ALL;
5
6 entity TB_NanoProcessor is
7 end TB_NanoProcessor;
8
9 architecture testbench of TB_NanoProcessor is
10
11 component NanoProcessor
12     Port ( Reset : in STD_LOGIC;
13             Clk : in STD_LOGIC;
14             Overflow : out STD_LOGIC;
15             Zero : out STD_LOGIC;
16             R7_LED : out STD_LOGIC_VECTOR (3 downto 0);
17             R7_Display: out STD_LOGIC_VECTOR(6 downto 0);
18             Anode: out STD_LOGIC_VECTOR(3 downto 0));
19
20 end component;
21
22 signal Reset, Overflow, Zero : std_logic;
23 signal Clk : STD_LOGIC := '0';
24 signal R7_Display: STD_LOGIC_VECTOR(6 downto 0);
25 signal R7_LED, Anode : STD_LOGIC_VECTOR (3 downto 0);
26
27 begin
28
29     UUT : NanoProcessor
30     port map (
31         Reset => Reset,
32         Clk => Clk,
33         Overflow => Overflow,
34         Zero => Zero,
35         R7_LED => R7_LED,
36         R7_Display => R7_Display,
37         Anode => Anode
38     );
39
40     clk_proc: process
41
42         o
43         o
44         o
45         o
46         o
47         o
48
49         stimuli: process
50             begin
51
52             o
53             o
54             o
55             o
56             o
57             o
58             o
59             o
60             o
61             o
62             o
63
64         end process;
65
66     end testbench;
67

```

4. Timing Diagram



Constraints file

```

Basys3Labs.xdc
C:/Users/ralee/OneDrive/Desktop/nano processor/project_1.2/project_1.2/srcs/constrs_1/newBasys3Labs.xdc

Q | M | ← | → | X | E | F | X | // | B | ♡ |

1: ## This file is a general .xdc for the Basys3 rev B board
2: ## To use it in a project:
3: ## - uncomment the lines corresponding to used pins
4: ## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project
5:
6: ## Clock signal
7: set_property PACKAGE_PIN W5 [get_ports {Clk}]
8:     set_property IOSTANDARD LVCMOS33 [get_ports {Clk}]
9:         create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports {Clk}]
10:
11: ## Switches
12: #set_property PACKAGE_PIN V17 [get_ports {A}]
13:     #set_property IOSTANDARD LVCMOS33 [get_ports {A}]
14: #set_property PACKAGE_PIN V16 [get_ports {B}]
15:     #set_property IOSTANDARD LVCMOS33 [get_ports {B}]
16: #set_property PACKAGE_PIN W16 [get_ports {C}]
17:     #set_property IOSTANDARD LVCMOS33 [get_ports {C}]
18: #set_property PACKAGE_PIN W17 [get_ports {sv[3]}]
19:     #set_property IOSTANDARD LVCMOS33 [get_ports {sv[3]}]
20: #set_property PACKAGE_PIN W15 [get_ports {sv[4]}]
21:     #set_property IOSTANDARD LVCMOS33 [get_ports {sv[4]}]
22: #set_property PACKAGE_PIN V15 [get_ports {sv[5]}]
23:     #set_property IOSTANDARD LVCMOS33 [get_ports {sv[5]}]
24: #set_property PACKAGE_PIN W14 [get_ports {sv[6]}]
25:     #set_property IOSTANDARD LVCMOS33 [get_ports {sv[6]}]
26: #set_property PACKAGE_PIN W13 [get_ports {sv[7]}]
27:     #set_property IOSTANDARD LVCMOS33 [get_ports {sv[7]}]
28: #set_property PACKAGE_PIN V2 [get_ports {sv[8]}]
29:     #set_property IOSTANDARD LVCMOS33 [get_ports {sv[8]}]
30: #set_property PACKAGE_PIN T3 [get_ports {sv[9]}]
31:     #set_property IOSTANDARD LVCMOS33 [get_ports {sv[9]}]
32: #set_property PACKAGE_PIN T2 [get_ports {sv[10]}]
33:     #set_property IOSTANDARD LVCMOS33 [get_ports {sv[10]}]
34: #set_property PACKAGE_PIN R3 [get_ports {sv[11]}]
35:     #set_property IOSTANDARD LVCMOS33 [get_ports {sv[11]}]
36: #set_property PACKAGE_PIN W2 [get_ports {sv[12]}]
37:     #set_property IOSTANDARD LVCMOS33 [get_ports {sv[12]}]
38: #set_property PACKAGE_PIN U1 [get_ports {sv[13]}]
39:     #set_property IOSTANDARD LVCMOS33 [get_ports {sv[13]}]
40: #set_property PACKAGE_PIN T1 [get_ports {sv[14]}]
41:     #set_property IOSTANDARD LVCMOS33 [get_ports {sv[14]}]
42: #set_property PACKAGE_PIN R2 [get_ports {sv[15]}]
43:     #set_property IOSTANDARD LVCMOS33 [get_ports {sv[15]}]
44:
45:
46: ## LEDs
47: set_property PACKAGE_PIN U16 [get_ports {R7_LED[0]}]
48:     set_property IOSTANDARD LVCMOS33 [get_ports {R7_LED[0]}]
49: set_property PACKAGE_PIN E19 [get_ports {R7_LED[1]}]
50:     set_property IOSTANDARD LVCMOS33 [get_ports {R7_LED[1]}]
51: set_property PACKAGE_PIN U19 [get_ports {R7_LED[2]}]
52:     set_property IOSTANDARD LVCMOS33 [get_ports {R7_LED[2]}]
53: set_property PACKAGE_PIN V19 [get_ports {R7_LED[3]}]
54:     set_property IOSTANDARD LVCMOS33 [get_ports {R7_LED[3]}]
55: #set_property PACKAGE_PIN W18 [get_ports {R7_LED[4]}]
56:     #set_property IOSTANDARD LVCMOS33 [get_ports {R7_LED[4]}]
57: #set_property PACKAGE_PIN U18 [get_ports {led[5]}]
58:     #set_property IOSTANDARD LVCMOS33 [get_ports {led[5]}]
59: #set_property PACKAGE_PIN V14 [get_ports {led[6]}]
60:     #set_property IOSTANDARD LVCMOS33 [get_ports {led[6]}]
61: #set_property PACKAGE_PIN V14 [get_ports {led[7]}]
62:     #set_property IOSTANDARD LVCMOS33 [get_ports {led[7]}]
63: #set_property PACKAGE_PIN V13 [get_ports {led[8]}]
64:     #set_property IOSTANDARD LVCMOS33 [get_ports {led[8]}]
65: #set_property PACKAGE_PIN V3 [get_ports {led[9]}]
66:     #set_property IOSTANDARD LVCMOS33 [get_ports {led[9]}]
67: #set_property PACKAGE_PIN W3 [get_ports {led[10]}]
68:     #set_property IOSTANDARD LVCMOS33 [get_ports {led[10]}]
69: #set_property PACKAGE_PIN U9 [get_ports {led[11]}]
70:     #set_property IOSTANDARD LVCMOS33 [get_ports {led[11]}]
71: #set_property PACKAGE_PIN P3 [get_ports {led[12]}]
72:     #set_property IOSTANDARD LVCMOS33 [get_ports {led[12]}]
73: #set_property PACKAGE_PIN H3 [get_ports {led[13]}]
74:     #set_property IOSTANDARD LVCMOS33 [get_ports {led[13]}]
75: set_property PACKAGE_PIN P4 [get_ports {Zero}]
76:     set_property IOSTANDARD LVCMOS33 [get_ports {Zero}]
77: set_property PACKAGE_PIN L1 [get_ports {Overflow}]
78:     set_property IOSTANDARD LVCMOS33 [get_ports {Overflow}]
79:
80:

```

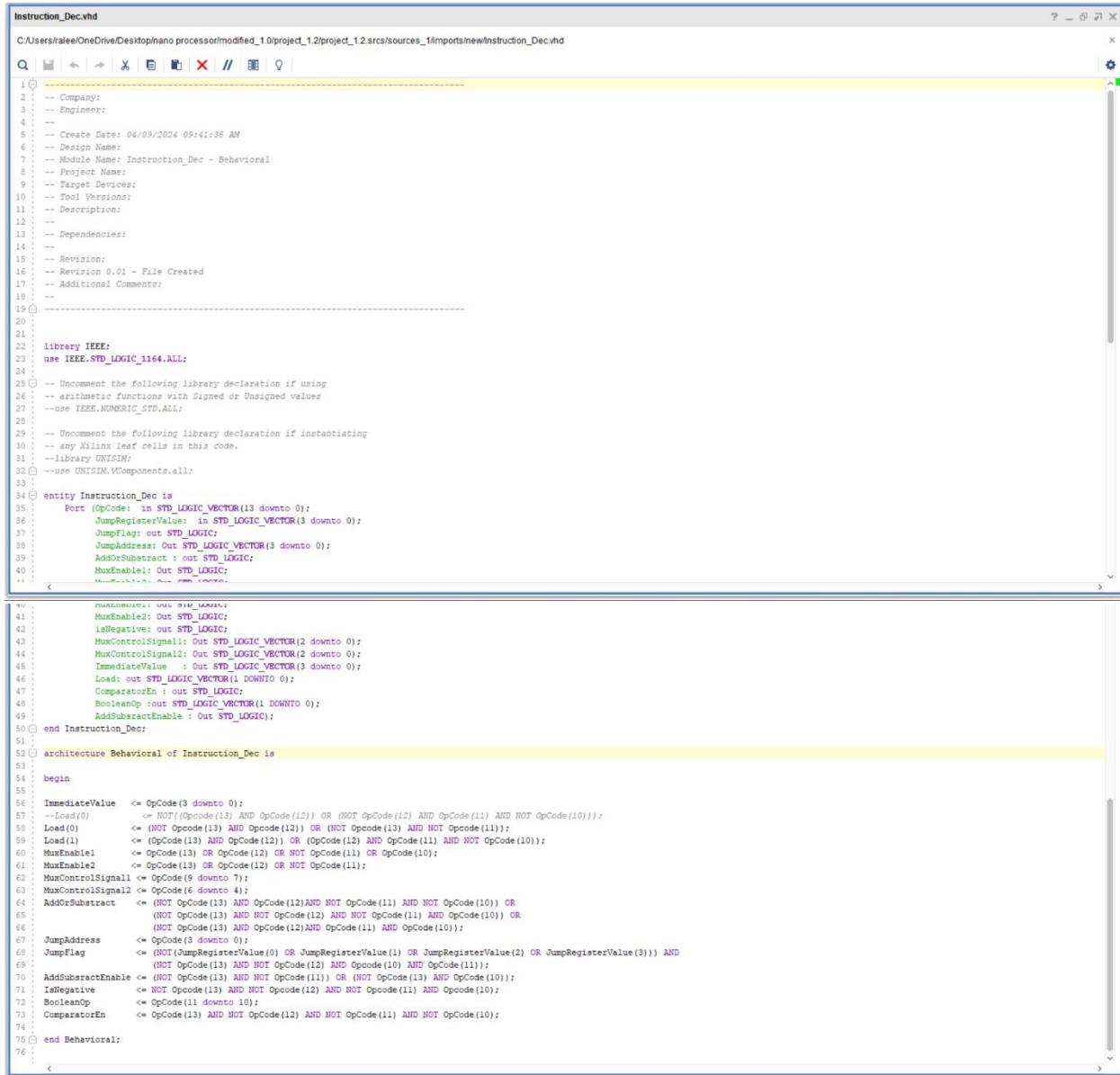
```
79 :  
80 :  
81 :##7 segment display  
82 :    set_property PACKAGE_PIN W7 [get_ports {R7_Display[0]}]  
83 :        set_property IOSTANDARD LVCMOS33 [get_ports {R7_Display[0]}]  
84 :    set_property PACKAGE_PIN W6 [get_ports {R7_Display[1]}]  
85 :        set_property IOSTANDARD LVCMOS33 [get_ports {R7_Display[1]}]  
86 :    set_property PACKAGE_PIN U8 [get_ports {R7_Display[2]}]  
87 :        set_property IOSTANDARD LVCMOS33 [get_ports {R7_Display[2]}]  
88 :    set_property PACKAGE_PIN V8 [get_ports {R7_Display[3]}]  
89 :        set_property IOSTANDARD LVCMOS33 [get_ports {R7_Display[3]}]  
90 :    set_property PACKAGE_PIN U5 [get_ports {R7_Display[4]}]  
91 :        set_property IOSTANDARD LVCMOS33 [get_ports {R7_Display[4]}]  
92 :    set_property PACKAGE_PIN V5 [get_ports {R7_Display[5]}]  
93 :        set_property IOSTANDARD LVCMOS33 [get_ports {R7_Display[5]}]  
94 :    set_property PACKAGE_PIN U7 [get_ports {R7_Display[6]}]  
95 :        set_property IOSTANDARD LVCMOS33 [get_ports {R7_Display[6]}]  
96 :  
97 :##set_property PACKAGE_PIN V7 [get_ports dp]  
98 :    #set_property IOSTANDARD LVCMOS33 [get_ports dp]  
99 :  
100: set_property PACKAGE_PIN U2 [get_ports {Anode[0]}]  
101:     set_property IOSTANDARD LVCMOS33 [get_ports {Anode[0]}]  
102: set_property PACKAGE_PIN U4 [get_ports {Anode[1]}]  
103:     set_property IOSTANDARD LVCMOS33 [get_ports {Anode[1]}]  
104: set_property PACKAGE_PIN V4 [get_ports {Anode[2]}]  
105:     set_property IOSTANDARD LVCMOS33 [get_ports {Anode[2]}]  
106: set_property PACKAGE_PIN W4 [get_ports {Anode[3]}]  
107:     set_property IOSTANDARD LVCMOS33 [get_ports {Anode[3]}]  
108:  
109:  
110: ##Buttons  
111: set_property PACKAGE_PIN U18 [get_ports Reset]  
112:     set_property IOSTANDARD LVCMOS33 [get_ports Reset]  
113: #set_property PACKAGE_PIN T18 [get_ports btnU]  
114:     #set_property IOSTANDARD LVCMOS33 [get_ports btnU]
```

Additional components in the Modified version

Instruction Decoder

This new instruction decoder supports 14 bit instructions and send signals to newly connected entities too.

1. Design source file

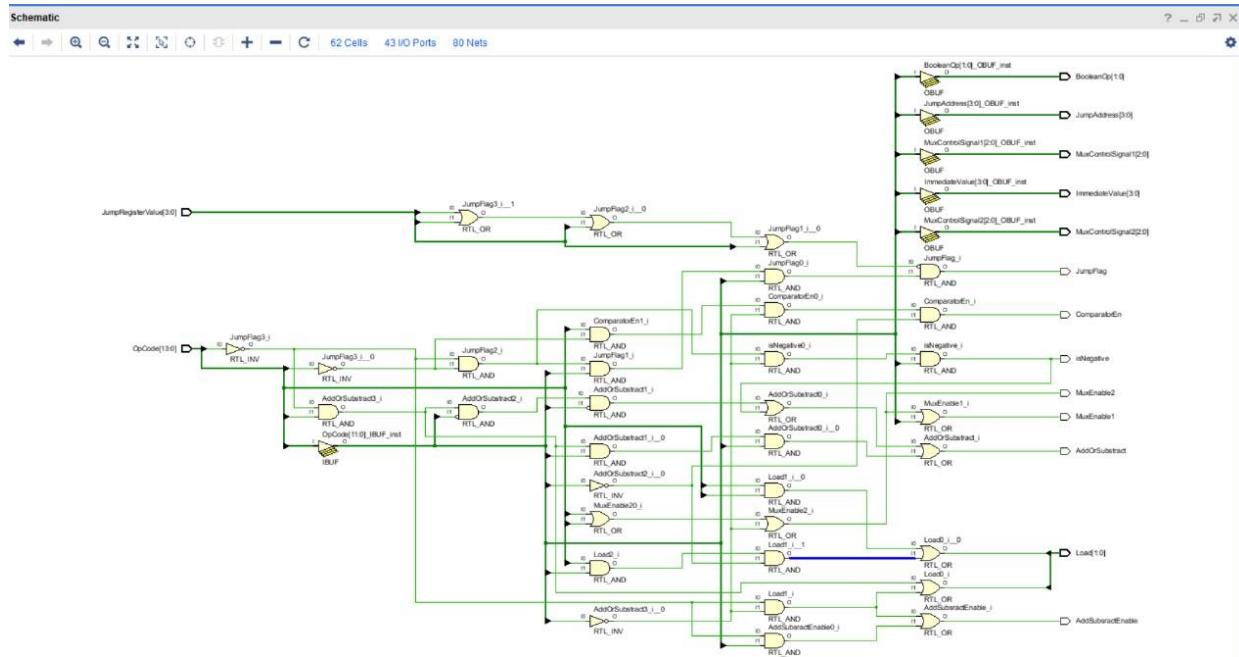


The screenshot shows a VHDL source code editor window titled "Instruction_Dec.vhd". The code is a behavioral model for an instruction decoder. It includes comments, library declarations, port definitions, and an architecture section. The architecture contains several signal assignments using the '`<=`' operator, defining various control signals based on the input Opcode.

```
Instruction_Dec.vhd
C:/Users/rakeel/OneDrive/Desktop/nano processor/modified_1.0/project_1.2/project_1.2.srcc/sources_1/imports/new/Instruction_Dec.vhd

1 -- Company;
2 -- Engineer;
3 --
4 --
5 -- Create Date: 04/09/2024 09:41:35 AM
6 -- Design Name:
7 -- Module Name: Instruction_Dec - Behavioral
8 -- Project Name:
9 -- Target Devices:
10 -- Tool Versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 
20 
21 
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24 
25 -- Uncomment the following library declaration if using
26 -- arithmetic functions with Signed or Unsigned values
27 --use IEEE.NUMERIC_STD.ALL;
28 
29 -- Uncomment the following library declaration if instantiating
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33 
34 entity Instruction_Dec is
35     Port (Opcode: in STD_LOGIC_VECTOR(13 downto 0);
36             JumpRegisterValue: in STD_LOGIC_VECTOR(3 downto 0);
37             JumpFlag: out STD_LOGIC;
38             JumpAddress: Out STD_LOGIC_VECTOR(3 downto 0);
39             AddOrSubtract : out STD_LOGIC;
40             MuxEnable1: Out STD_LOGIC;
41             MuxEnable2: Out STD_LOGIC;
42             IsNegative: out STD_LOGIC;
43             MuxControlSignal1: Out STD_LOGIC_VECTOR(2 downto 0);
44             MuxControlSignal2: Out STD_LOGIC_VECTOR(2 downto 0);
45             ImmediateValue : Out STD_LOGIC_VECTOR(3 downto 0);
46             Load: out STD_LOGIC_VECTOR(1 DOWNTO 0);
47             ComparatorEn : out STD_LOGIC;
48             BooleanOp : out STD_LOGIC_VECTOR(1 DOWNTO 0);
49             AddSubtractEnable : Out STD_LOGIC);
50 end Instruction_Dec;
51 
52 architecture Behavioral of Instruction_Dec is
53 begin
54 
55 
56     ImmediateValue <= OpCode(3 downto 0);
57     --Load(0) <= NOT((OpCode(13) AND OpCode(12)) OR (NOT OpCode(12) AND OpCode(11)) AND NOT OpCode(10));
58     Load(0) <= (NOT OpCode(13) AND OpCode(12)) OR (NOT OpCode(13) AND NOT OpCode(11));
59     Load(1) <= (OpCode(13) AND OpCode(12)) OR (OpCode(12) AND OpCode(11) AND NOT OpCode(10));
60     MuxEnable1 <= OpCode(13) OR OpCode(12) OR NOT OpCode(11) OR OpCode(10);
61     MuxEnable2 <= OpCode(13) OR OpCode(12) OR NOT OpCode(11);
62     MuxControlSignal1 <= OpCode(9 downto 7);
63     MuxControlSignal2 <= OpCode(6 downto 4);
64     AddOrSubtract <= (NOT OpCode(13) AND OpCode(12) AND NOT OpCode(11) AND NOT OpCode(10)) OR
65         ((NOT OpCode(13) AND NOT OpCode(12) AND NOT OpCode(11) AND OpCode(10)) OR
66         ((NOT OpCode(13) AND OpCode(12) AND OpCode(11) AND OpCode(10)));
67     JumpAddress <= OpCode(3 downto 0);
68     JumpFlag <= NOT(JumpRegisterValue(0) OR JumpRegisterValue(1) OR JumpRegisterValue(2) OR JumpRegisterValue(3)) AND
69         ((NOT OpCode(13) AND NOT OpCode(12) AND OpCode(10) AND OpCode(11));
70     AddSubtractEnable <= (NOT OpCode(13) AND NOT OpCode(11)) OR (NOT OpCode(13) AND OpCode(10));
71     IsNegative <= NOT(OpCode(13) AND NOT OpCode(12) AND NOT OpCode(11) AND OpCode(10));
72     BooleanOp <= OpCode(11 downto 10);
73     ComparatorEn <= OpCode(13) AND NOT OpCode(12) AND NOT OpCode(11) AND NOT OpCode(10);
74 
75 end Behavioral;
76 
```

2. Elaborated design schematic



3. Simulation source file

```

Sim_Ins_Dec.vhd
C:/Users/raleel/OneDrive/Desktop/nano processor/modified_1.0/project_1.2/project_1.2.srccs/sim_1imports/new/Sim_Ins_Dec.vhd

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity TB_Instruction_Dec is
end TB_Instruction_Dec;
architecture Behavioral of TB_Instruction_Dec is
COMPONENT Instruction_Dec is
    Port (OpCode: in STD_LOGIC_VECTOR(13 downto 0);
          JumpRegisterValue: in STD_LOGIC_VECTOR(3 downto 0);
          JumpFlag: out STD_LOGIC;
          JumpAddress: out STD_LOGIC_VECTOR(3 downto 0);
          AddOrSubtract: out STD_LOGIC;
          MuxEnable1: out STD_LOGIC;
          MuxEnable2: out STD_LOGIC;
          MuxEnable3: out STD_LOGIC;
          MuxControlSignal1: out STD_LOGIC_VECTOR(2 downto 0);
          MuxControlSignal2: out STD_LOGIC_VECTOR(2 downto 0);
          ImmediateValue: out STD_LOGIC_VECTOR(3 downto 0);
          Load: out STD_LOGIC_VECTOR(1 DOWNTO 0);
          ComparatorEn: out STD_LOGIC;
          BooleanOp: out STD_LOGIC_VECTOR(1 DOWNTO 0);
          AddOrSubtractEnable: out STD_LOGIC);
end COMPONENT;
-- Signals declaration
signal OpCode: STD_LOGIC_VECTOR(13 downto 0);
signal JumpRegisterValue: STD_LOGIC_VECTOR(3 downto 0);
signal JumpFlag: STD_LOGIC;
signal JumpAddress: STD_LOGIC_VECTOR(3 downto 0);
signal AddOrSubtract: STD_LOGIC;
signal MuxEnable1: STD_LOGIC;
signal MuxEnable2: STD_LOGIC;
signal MuxEnable3: STD_LOGIC;
signal MuxControlSignal1: STD_LOGIC_VECTOR(2 downto 0);
signal MuxControlSignal2: STD_LOGIC_VECTOR(2 downto 0);
signal ImmediateValue: STD_LOGIC_VECTOR(3 downto 0);
signal Load: STD_LOGIC_VECTOR(1 DOWNTO 0);
signal ComparatorEn: STD_LOGIC;
signal BooleanOp: STD_LOGIC_VECTOR(1 DOWNTO 0);
signal AddOrSubtractEnable: STD_LOGIC;
-- End of component declarations
begin
end;

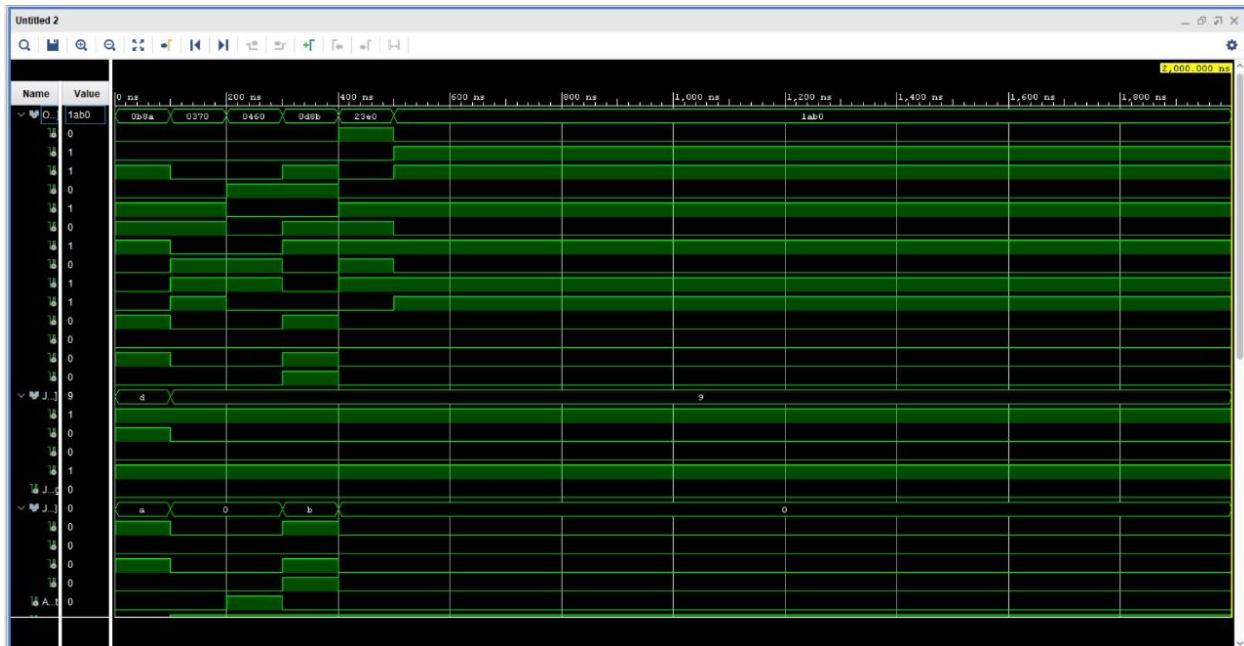
```

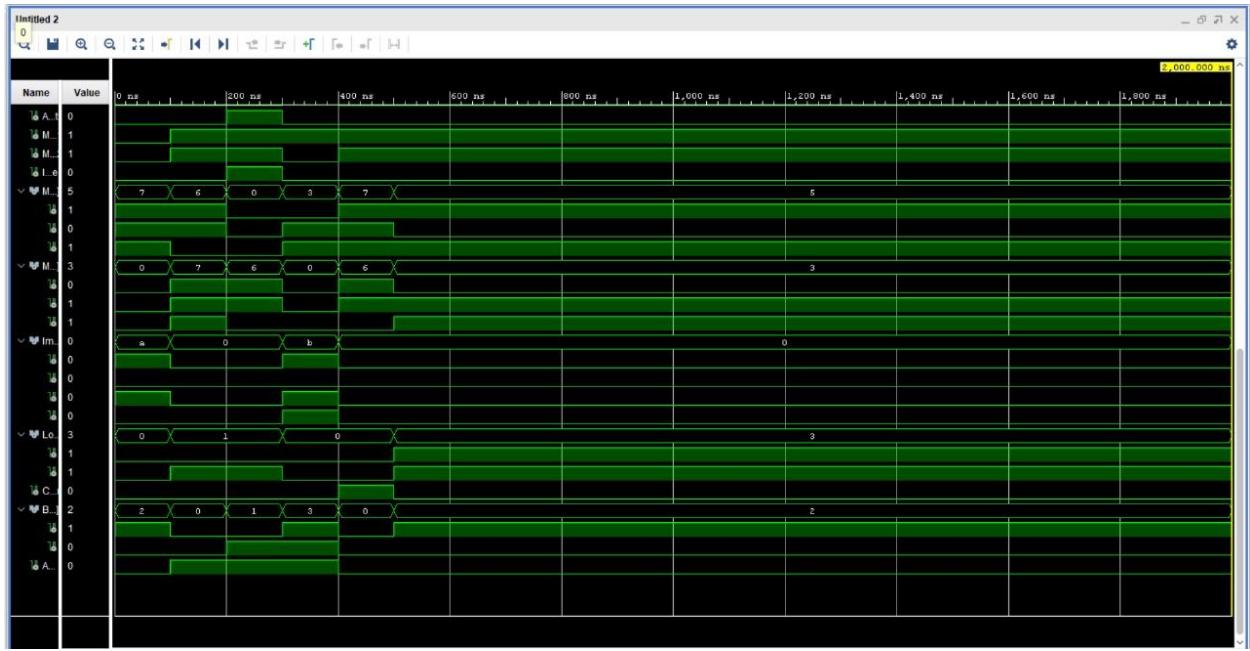
```

40 : signal ComparatorEn    : STD_LOGIC;
41 : signal BooleanOp     : STD_LOGIC_VECTOR(1 DOWNTO 0);
42 : signal AddSubtractEnable: STD_LOGIC;
43 :
44 begin
45
46   UUT: Instruction_Dec port map (
47     OpCode      => OpCode,
48     JumpRegisterValue => JumpRegisterValue,
49     JumpFlag     => JumpFlag,
50     JumpAddress  => JumpAddress,
51     AddrOrSubtract => AddrOrSubtract,
52     MuxEnable1   => MuxEnable1,
53     MuxEnable2   => MuxEnable2,
54     IsNegative   => IsNegative,
55     MuxControlSignal1 => MuxControlSignal1,
56     MuxControlSignal2 => MuxControlSignal2,
57     ImmediateValue => ImmediateValue,
58     Load         => Load,
59     ComparatorEn  => ComparatorEn,
60     BooleanOp    => BooleanOp,
61     AddSubtractEnable=> AddSubtractEnable
62   );
63
64   process
65   begin
66
67     OpCode <= "0010111100001010"; --MOV R7 10
68     JumpRegisterValue <= "1101";
69     wait for 100 ns;
70
71     OpCode <= "00001101110000"; --ADD R6 R7
72     JumpRegisterValue <= "1001";
73     wait for 100 ns;
74
75     OpCode <= "00010001100000"; --NEG R6
76     JumpRegisterValue <= "1001";
77     wait for 100 ns;
78
79     OpCode <= "001101100001011"; --JER R3 1011
80     JumpRegisterValue <= "1001";
81     wait for 100 ns;
82
83     OpCode <= "001101100001011"; --JER R3 1011
84     JumpRegisterValue <= "1001";
85     wait for 100 ns;
86
87     OpCode <= "10001111100000"; --COMP R7 R6
88     JumpRegisterValue <= "1001";
89     wait for 100 ns;
90
91   end process Stimulus_Process;
92
93 end Behavioral;
94

```

4. Timing Diagram





4-bit Adder

1. Design source file

```
fourBitAdder.vhd
C:/Users/ratee/OneDrive/Desktop/nano processor/modified_1.0/project_1.2/projed_1.2.srcc/sources_1/imports/newfourBitAdder.vhd

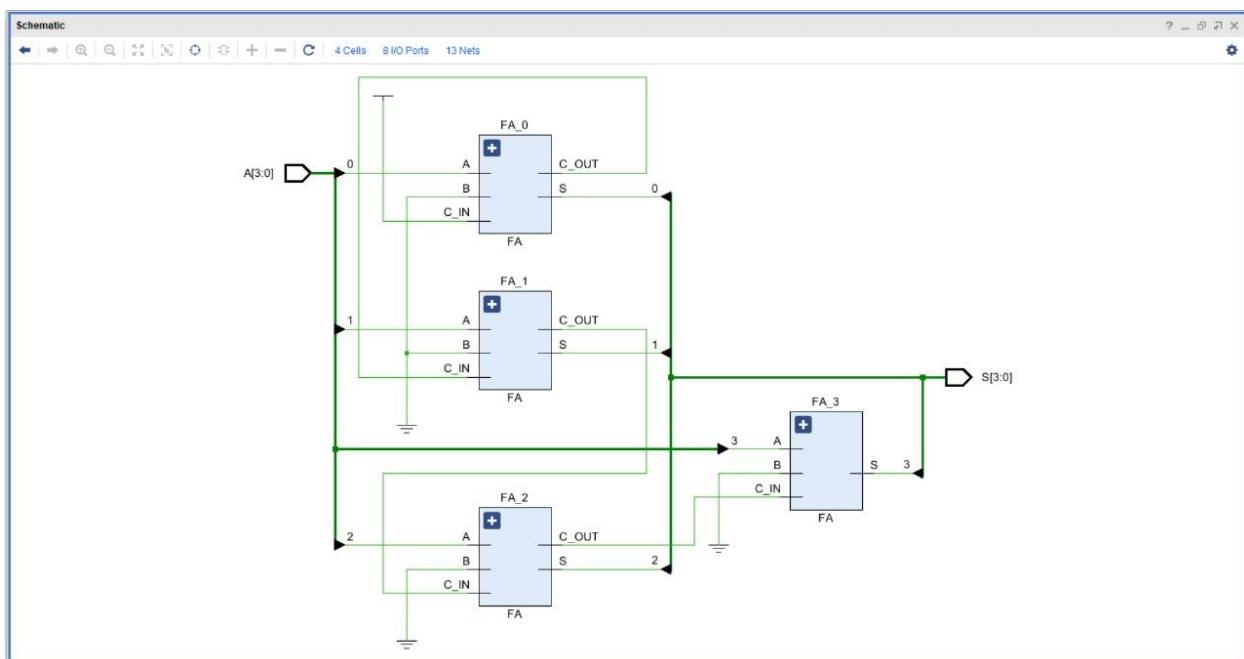
1 -- Company:
2 -- Engineer:
3 --
4 --
5 -- Create Date: 04/25/2024 06:40:33 AM
6 -- Design Name:
7 -- Module Name: fourBitAdder - Behavioral
8 -- Project Name:
9 -- Target Devices:
10 -- Tool Versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 
20 
21 
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24 
25 -- Uncomment the following library declaration if using
26 -- arithmetic functions with Signed or Unsigned values
27 --use IEEE.NUMERIC_STD.ALL;
28 
29 -- Uncomment the following library declaration if instantiating
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33 
34 entity fourBitAdder is
35   Port ( A : in STD.LOGIC_VECTOR (3 downto 0);
36         S : out STD.LOGIC_VECTOR (3 downto 0));
37 end fourBitAdder;
38 
39 architecture Behavioral of fourBitAdder is
40 begin
41   -- Your implementation here
42 end Behavioral;
```

```

41 component FA
42     port (
43         A: in std_logic;
44         B: in std_logic;
45         C_IN: in std_logic;
46         S: out std_logic;
47         C_OUT: out std_logic);
48 end component;
49 SIGNAL FA0_C, FA0_S, FA1_C, FA1_S, FA2_C, FA2_S, FA3_C, FA3_S : std_logic := '0';
50
51 begin
52     FA_0 : FA
53         port map (
54             A => A(0),
55             B => '0',
56             C_IN => '1',
57             S => S(0),
58             C_OUT => FA0_C);
59     FA_1 : FA
60         port map (
61             A => A(1),
62             B => '0',
63             C_IN => FA0_C,
64             S => S(1),
65             C_OUT => FA1_C);
66     FA_2 : FA
67         port map (
68             A => A(2),
69             B => '0',
70             C_IN => FA1_C,
71             S => S(2),
72             C_OUT => FA2_C);
73     FA_3 : FA
74         port map (
75             A => A(3),
76             B => '0',
77             C_IN => FA2_C,
78             S => S(3),
79             C_OUT => FA3_C);
80 end Behavioral;
<

```

2. Elaborated design schematic

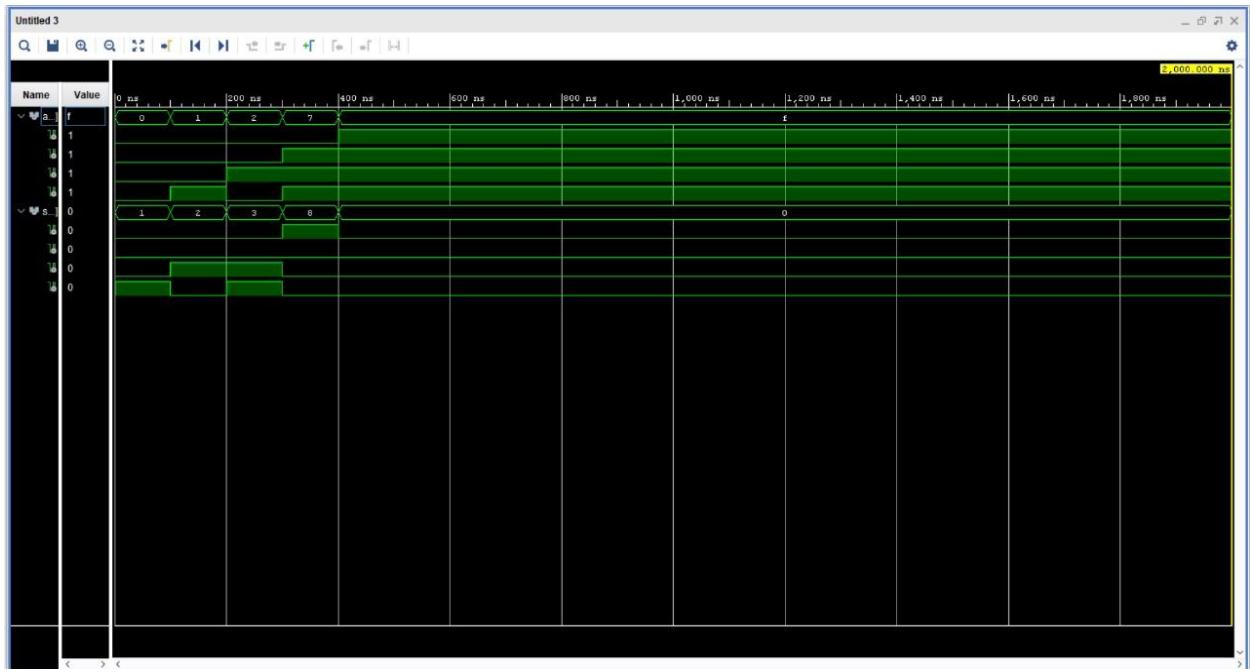


3. Simulation source file

```
TB_fourBitAdder.vhd
C:\Users\alee\OneDrive\Desktop\nano processor\modified_1.0\project_1.2\srcs\sim_1\imports\new\TB_fourBitAdder.vhd

1 -- Company:
2 -- Engineer:
3 --
4 --
5 -- Create Date: 04/26/2024 06:48:17 AM
6 -- Design Name: TB_fourBitAdder - Behavioral
7 -- Module Name: TB_fourBitAdder
8 -- Project Name:
9 -- Target Devices:
10 -- Tool Versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 -----
20
21 library IEEE;
22 use IEEE.STD_LOGIC_1164.ALL;
23
24 -- Uncomment the following library declaration if using
25 -- arithmetic functions with Signed or Unsigned values
26 --use IEEE.NUMERIC_STD.ALL;
27
28 -- Uncomment the following library declaration if instantiating
29 -- any Xilinx leaf cells in this code.
30 --library UNISIM;
31 --use UNISIM.VComponents.all;
32
33 entity TB_fourBitAdder is
34     Port ( );
35 end TB_fourBitAdder;
36
37 architecture Behavioral of TB_fourBitAdder is
38
39 component fourBitAdder
40     Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
41             S : out STD_LOGIC_VECTOR (3 downto 0));
42 end component;
43
44 signal a,s : std_logic_vector (3 downto 0);
45
46 begin
47
48 begin
49
50     uut : fourBitAdder
51         port map (
52             A => a,
53             S => s
54         );
55
56 process
57 begin
58     a <= "0000";
59     wait for 100ns;
60
61     a <= "0001";
62     wait for 100ns;
63
64     a <= "0010";
65     wait for 100ns;
66
67     a <= "0111";
68     wait for 100ns;
69
70     a <= "1111";
71     wait for 100ns;
72
73     wait;
74
75 end process;
76
77
78 end Behavioral;
79
```

4. Timing Diagram



4-bit Adder-Subtractor

1. Design source file

```
4_bit_Adder_Subtractor.vhd
C:/Users/rakeel/Desktop/nano processor/modified_1.0/project_1.2/project_1.2/srcs/sources_1/imports/new4_bit_Adder_Subtractor.vhd

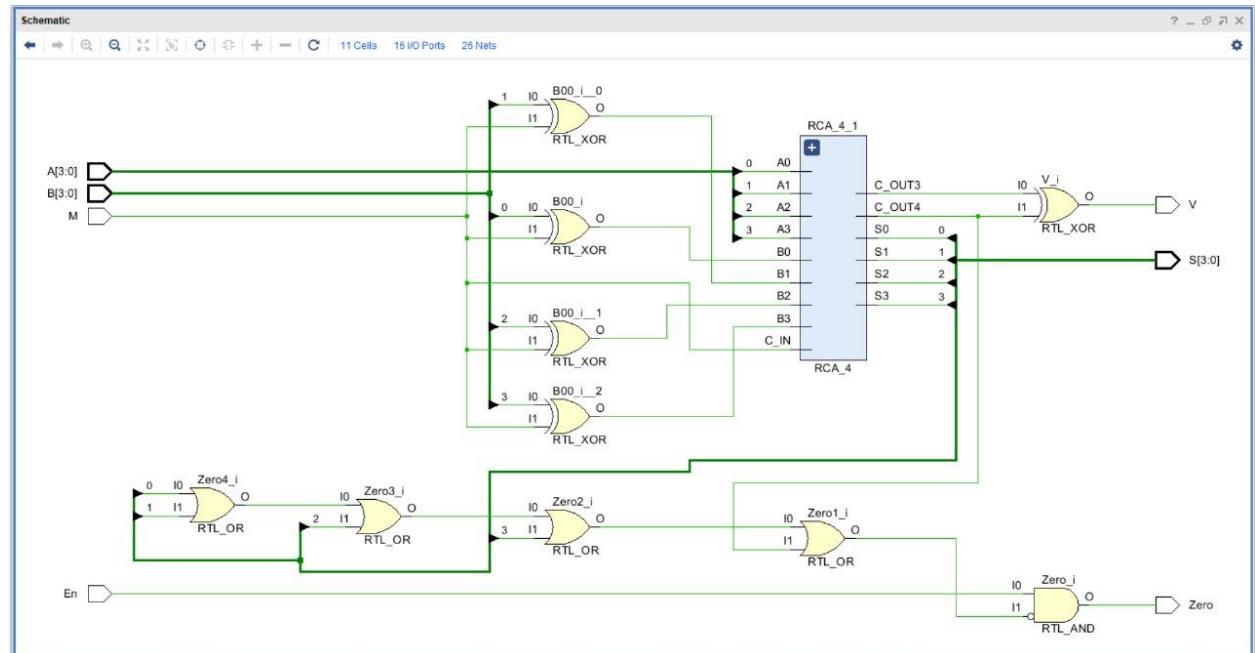
1 -- Company:
2 -- Engineer:
3 --
4 --
5 -- Create Date: 04/02/2024 01:54:02 PM
6 -- Design Name:
7 -- Module Name: 4_bit_Adder_Subtractor - Behavioral
8 -- Project Name:
9 -- Target Devices:
10 -- Tool Versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 --
20 --
21 library IEEE;
22 use IEEE.STD_LOGIC_1164.ALL;
23 --
24 -- Uncomment the following library declaration if using
25 -- arithmetic functions with Signed or Unsigned values
26 --use IEEE.NUMERIC_STD.ALL;
27 --
28 -- Uncomment the following library declaration if instantiating
29 -- any Xilinx leaf cells in this code,
30 --library UNISIM;
31 --use UNISIM.VComponents.all;
32 --
33 entity Fourbit_Adder_Subtractor is
34     Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
35             B : in STD_LOGIC_VECTOR (3 downto 0);
36             S : out STD_LOGIC_VECTOR (3 downto 0);
37             E : in STD_LOGIC;
38             M : in STD_LOGIC;
39             V : out STD_LOGIC;
40             );
41 end;
```

```

40 :         V : out STD_LOGIC;
41 :         Zero : out STD_LOGIC;
42 :     );
43 end Fourbit_Adder_Subtractor;
44
45 architecture Behavioral of Fourbit_Adder_Subtractor is
46 component RCA_4
47     Port ( A0 : in STD_LOGIC;
48             A1 : in STD_LOGIC;
49             A2 : in STD_LOGIC;
50             A3 : in STD_LOGIC;
51             B0 : in STD_LOGIC;
52             B1 : in STD_LOGIC;
53             B2 : in STD_LOGIC;
54             B3 : in STD_LOGIC;
55             C_IN : in STD_LOGIC;
56             S0 : out STD_LOGIC;
57             S1 : out STD_LOGIC;
58             S2 : out STD_LOGIC;
59             S3 : out STD_LOGIC;
60             C_OUT3 : out STD_LOGIC;
61             C_OUT4 : out STD_LOGIC);
62
63 end component;
64
65 SIGNAL X,Y :STD_LOGIC;
66 SIGNAL BB,Temp: STD_LOGIC_VECTOR(3 DOWNTO 0);
67
68 begin
69
70 process (B, M)
71 begin
72 for i in 0 to B'length-1 loop
73 BB(i) <= B(i) XOR M;
74 end loop;
75 end process;
76
77 RCA_4_1 : RCA_4
78 PORT MAP (
79 A0 => A(0),
80 A1 => A(1),
81 A2 => A(2),
82 A3 => A(3),
83 B0 => BB(0),
84 B1 => BB(1),
85 B2 => BB(2),
86 B3 => BB(3),
87 C_IN => M,
88 S0 => Temp(0),
89 S1 => Temp(1),
90 S2 => Temp(2),
91 S3 => Temp(3),
92 C_OUT3 => X,
93 C_OUT4 => Y
94 );
95
96 S <= Temp;
97 V <= (X XOR Y);
98 Zero <= En AND NOT(Temp(0) OR Temp(1) OR Temp(2) OR Temp(3) OR Y);
99
100 end Behavioral;
101

```

2. Elaborated design schematic



3. Simulation source file

```

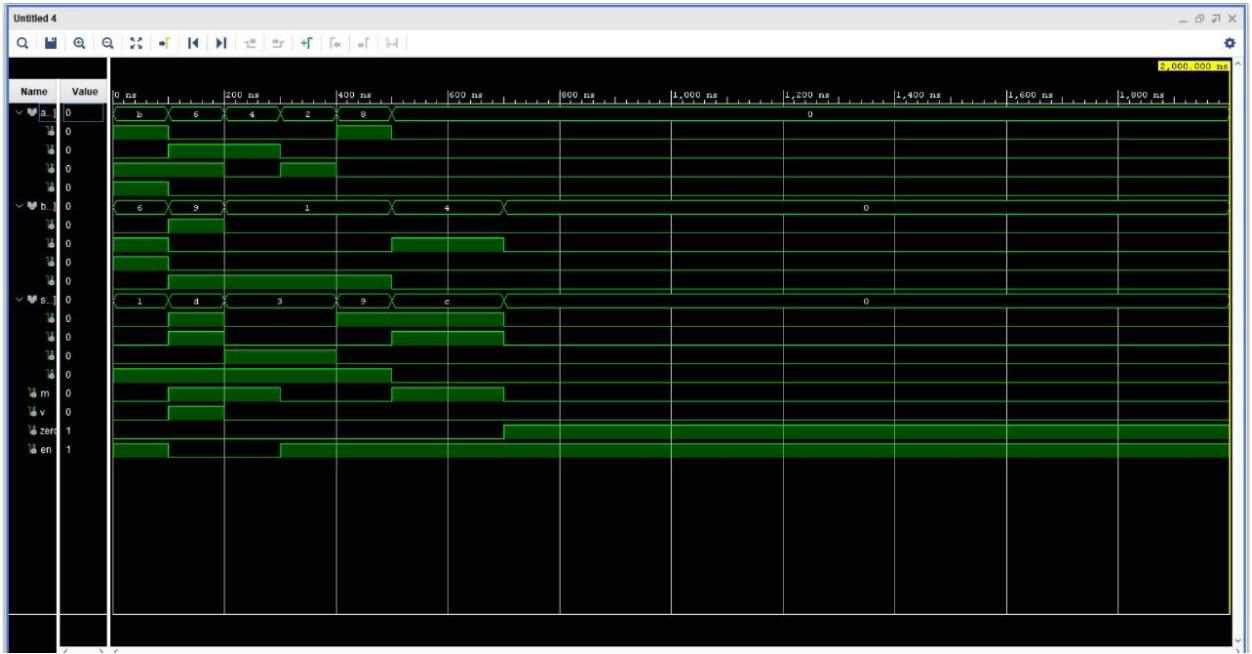
TB_4AdderSubtractor.vhd
C:/Users/raleelOneDrive/Desktop/nano processor/modified_1/0/project_1_2/project_1_2/srcs/sim_1/new/TB_4AdderSubtractor.vhd

Q   I   <   >   X   //   |   ?   X   I   S   E   C   Z   A   F   X   H   O

1 --
2 -- Company:
3 -- Engineer:
4 --
5 -- Create Date: 04/02/2024 03:05:53 PM
6 -- Design Name:
7 -- Module Name: sim_4_bit_adder_subtractor - Behavioral
8 -- Project Name:
9 -- Target Device:
10 -- Tool Versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 --
20
21
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24
25 -- Uncomment the following library declaration if using
26 -- arithmetic functions with Signed or Unsigned values
27 --use IEEE.NUMERIC_STD.ALL;
28
29 -- Uncomment the following library declaration if instantiating
30 -- any Klimo leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33
34 entity TB_AdderSubtractor is
35   -- Port ();
36 end TB_AdderSubtractor;
37
38 architecture Behavioral of TB_AdderSubtractor is
39 component Fourbit_Adder_Substrator
40   Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
41         B : in STD_LOGIC_VECTOR (3 downto 0);
42         S : out STD_LOGIC_VECTOR (3 downto 0);
43         M : in STD_LOGIC;
44         En : in STD_LOGIC;
45         V : out STD_LOGIC;
46         Zero : out STD_LOGIC
47       );
48 end component;
49
50 signal a,b,s :STD_LOGIC_VECTOR (3 downto 0);
51 signal m,v,zero, en : std_logic ;
52
53
54 begin
55  uut : Fourbit_Adder_Substrator
56  port map (
57    A => a,
58    B => b,
59    En =>en,
60    M => m,
61    V => v,
62    S => s,
63    Zero => zero
64  );
65
66 process begin
67
68  a <= "1011";
69  b <= "0110";
70  m <= '0';
71  en <= '1';
72  wait for 100ns;
73
74  a <= "0110";
75  b <= "1001";
76  m <= '1';
77  en <= '0';
78  wait for 100ns;
79  a <= "0100";
80  b <= "0001";
81  m <= '1';
82  wait for 100ns;
83  a <= "0010";
84  b <= "0001";
85  m <= '0';
86  en <= '1';
87  wait for 100ns;
88
89  a <= "1000";
90  b <= "0001";
91  m <= '0';
92  wait for 100ns;
93  a <= "0000";
94  b <= "0100";
95  m <= '1';
96  wait for 100ns;
97
98  a <= "0000";
99  b <= "0100";
100 m <= '1';
101 wait for 100ns;
102
103 a <= "0000";
104 b <= "0000";
105 m <= '0';
106 wait for 100ns;
107
108 wait;
109 end process;
110
111 end Behavioral;
112

```

4. Timing Diagram



Comparator

1. Design source file

```
Comparador.vhd
C:/Users/ralee/OneDrive/Desktop/processor/modified_1.0/project_1.2/project_1.2.srcs/sources_1/import/new/Comparador.vhd

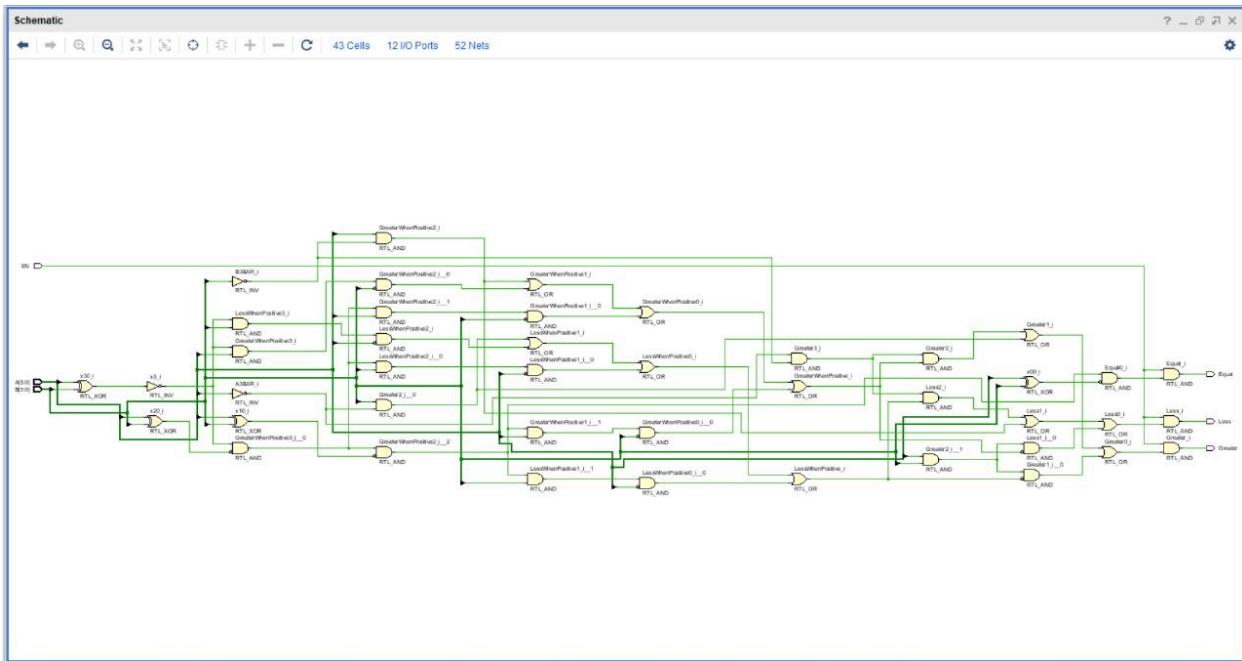
Q | L | ← | → | X | D | F | X | // | E | V | Q |
1 -- Company:
2 -- Engineer:
3 --
4 --
5 -- Create Date: 04/25/2024 02:48:03 PM
6 -- Design Name:
7 -- Module Name: Comparador - Behavioral
8 -- Project Name:
9 -- Target Devices:
10 -- Tool Versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 -----
20
21
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24
25 -- Uncomment the following library declaration if using
26 -- arithmetic functions with Signed or Unsigned values
27 --use IEEE.NUMERIC_STD.ALL;
28
29 -- Uncomment the following library declaration if instantiating
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33
34 entity Comparador is
35     Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
36             B : in STD_LOGIC_VECTOR (3 downto 0);
37             EN : in STD_LOGIC;
38             Greater : out STD_LOGIC;
39             Equal : out STD_LOGIC;
40             Less : out STD_LOGIC);
41 end Comparador;
```

```

40 :           Less : out STD_LOGIC);
41 end Comparator;
42
43 architecture Behavioral of Comparator is
44
45 SIGNAL x0,x1,x2,x3 : std_logic;
46 SIGNAL A0BAR,A1BAR,A2BAR,A3BAR : std_logic;
47 SIGNAL B0BAR,B1BAR,B2BAR,B3BAR : std_logic;
48 SIGNAL GreaterWhenPositive, LessWhenPositive : std_logic;
49
50 begin
51
52 x0 <= NOT(A(0)) XOR B(0);
53 x1 <= NOT(A(1)) XOR B(1);
54 x2 <= NOT(A(2)) XOR B(2);
55 x3 <= NOT(A(3)) XOR B(3);
56
57 A0BAR <= NOT(A(0));
58 A1BAR <= NOT(A(1));
59 A2BAR <= NOT(A(2));
60 A3BAR <= NOT(A(3));
61 B0BAR <= NOT(B(0));
62 B1BAR <= NOT(B(1));
63 B2BAR <= NOT(B(2));
64 B3BAR <= NOT(B(3));
65
66 GreaterWhenPositive <= (A(3) AND B1BAR) OR (X3 AND X2 AND A2BAR) OR (X3 AND X2 AND X1 AND A1) AND B1BAR) OR (X3 AND X2 AND X1 AND A(0) AND B0BAR);
67 LessWhenPositive <= (B(3) AND A3BAR) OR (X3 AND B(2) AND A2BAR) OR (X3 AND X2 AND B(1) AND A1BAR) OR (X3 AND X2 AND X1 AND B(0) AND A0BAR);
68
69 Equal <= EN AND ( X3 AND X2 AND X1 AND X0 );
70 Greater <= EN AND ( (A3BAR AND B3BAR AND GreaterWhenPositive) OR (A3BAR AND B(3)) OR (A(3) AND B(3) AND NOT(LessWhenPositive)) );
71 less <= EN AND ( (A3BAR AND B3BAR AND LessWhenPositive) OR (A(3) AND B3BAR) OR (A(3) AND B(3) AND NOT(GreaterWhenPositive)) );
72
73 end Behavioral;
74

```

2. Elaborated design schematic



3. Simulation source file

TB_Comparator.vhd

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
-- Uncomment the following library declaration if instantiating
-- any Kilns leaf cells in this code
--library UNISIM;
--use UNISIM.VComponents.all;
entity TB_Comparator is
    Port ( );
end TB_Comparator;

architecture Behavioral of TB_Comparator is
COMPONENT Comparator is
    Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
           B : in STD_LOGIC_VECTOR (3 downto 0);
           EN : in STD_LOGIC;
           less : out STD_LOGIC;
           equal : out STD_LOGIC;
           greater : out STD_LOGIC);
end COMPONENT;
begin
    Data_A,Data_B :STD LOGIC VECTOR (3 downto 0);
    signal EN,less,equal,greater : std_logic ;
begin
    begin
        uut : Comparator
        port map (
            A => Data_A,
            B => Data_B,
            En => EN,
            less => less,
            equal => equal,
            greater => greater
        );
        process begin
            Data_A <= "1011";
            Data_B <= "0011";
            EN <= '1';
            wait for 100ns;
            Data_A <= "1011";
            Data_B <= "0011";
            EN <= '1';
            wait for 100ns;
            Data_A <= "0111";
            Data_B <= "0100";
            wait for 100ns;
            Data_A <= "0110";
            Data_B <= "1110";
            wait for 100ns;
            Data_A <= "1101";
            Data_B <= "1101";
            wait for 100ns;
            wait;
        end process;
    end Behavioral;
<

```

TB_Comparator.vhd

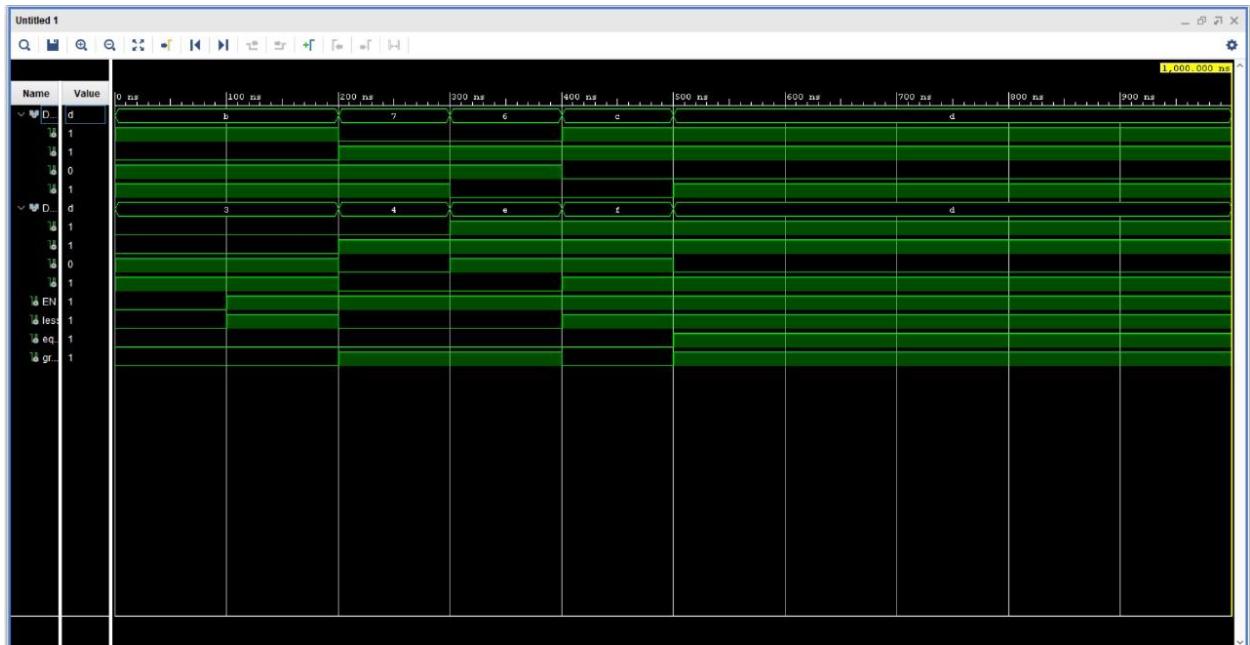
```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
-- Uncomment the following library declaration if instantiating
-- any Kilns leaf cells in this code
--library UNISIM;
--use UNISIM.VComponents.all;
entity TB_Comparator is
    Port ( );
end TB_Comparator;

architecture Behavioral of TB_Comparator is
COMPONENT Comparator is
    Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
           B : in STD_LOGIC_VECTOR (3 downto 0);
           EN : in STD_LOGIC;
           less : out STD_LOGIC;
           equal : out STD_LOGIC;
           greater : out STD_LOGIC);
end COMPONENT;
begin
    Data_A,Data_B :STD LOGIC VECTOR (3 downto 0);
    signal EN,less,equal,greater : std_logic ;
begin
    begin
        uut : Comparator
        port map (
            A => Data_A,
            B => Data_B,
            En => EN,
            less => less,
            equal => equal,
            greater => greater
        );
        process begin
            Data_A <= "1011";
            Data_B <= "0011";
            EN <= '1';
            wait for 100ns;
            Data_A <= "1011";
            Data_B <= "0011";
            EN <= '1';
            wait for 100ns;
            Data_A <= "0111";
            Data_B <= "0100";
            wait for 100ns;
            Data_A <= "0110";
            Data_B <= "1110";
            wait for 100ns;
            Data_A <= "1101";
            Data_B <= "1101";
            wait for 100ns;
            wait;
        end process;
    end Behavioral;
<

```

4. Timing Diagram



Boolean Operator

1. Design source file

```

Boolean_operator.vhd
C:/Users/rakeel/OneDrive/Desktop/nano processor/modified_1.0/project_1.2/project_1.2/srcs/sources_tl/imports/new/Boolean_operator.vhd

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
entity Boolean_operator is
    Port ( Input : in STD_LOGIC_VECTOR (1 downto 0);
           Data_A : in STD_LOGIC_VECTOR (3 downto 0);
           Data_B : in STD_LOGIC_VECTOR (3 downto 0);
           S_out : out STD_LOGIC_VECTOR (3 downto 0));
end Boolean_operator;

```

```

41 architecture Behavioral of Boolean_operator is
42
43 component Decoder_2_to_4
44   port(  

45     I : in STD_LOGIC_VECTOR (1 downto 0);  

46     EN : in STD_LOGIC;  

47     Y : out STD_LOGIC_VECTOR (3 downto 0));  

48 end component;
49
50 component AND_operator is
51   Port (A : in STD_LOGIC_VECTOR (3 downto 0);  

52   B : in STD_LOGIC_VECTOR (3 downto 0);  

53   EN : in STD_LOGIC;  

54   S : out STD_LOGIC_VECTOR (3 downto 0));  

55 end component;
56
57 component OR_operator is
58   Port (A : in STD_LOGIC_VECTOR (3 downto 0);  

59   B : in STD_LOGIC_VECTOR (3 downto 0);  

60   EN : in STD_LOGIC;  

61   S : out STD_LOGIC_VECTOR (3 downto 0));  

62 end component;
63
64 component NOT_operator is
65   Port (A : in STD_LOGIC_VECTOR (3 downto 0);  

66   EN : in STD_LOGIC;  

67   S : out STD_LOGIC_VECTOR (3 downto 0));  

68 end component;
69
70 component XOR_operator is
71   Port (A : in STD_LOGIC_VECTOR (3 downto 0);  

72   B : in STD_LOGIC_VECTOR (3 downto 0);  

73   EN : in STD_LOGIC;  

74   S : out STD_LOGIC_VECTOR (3 downto 0));  

75 end component;
76
77 signal AND_signal, OR_signal, NOT_signal, XOR_signal : STD_LOGIC;
78 signal decoder_output, AND_operator_output, OR_operator_output, NOT_operator_output, XOR_operator_output : STD_LOGIC_VECTOR (3 downto 0);
79
80 begin
81
82   Decoder_2_to_4_0: Decoder_2_to_4
83     port map(  

84       I=> Input,  

85       EN => '1',  

86       Y=> decoder_output);  

87   AND_operator_0: AND_operator
88     PORT MAP(  

89       A => Data_A,  

90       B => Data_B,  

91       EN => decoder_output(0),  

92       S => AND_operator_output);
93
94   OR_operator_0: OR_operator
95     PORT MAP(  

96       A => Data_A,  

97       B => Data_B,  

98       EN => decoder_output(1),  

99       S => OR_operator_output);
100
101  NOR_operator_0: NOT_operator
102    PORT MAP(  

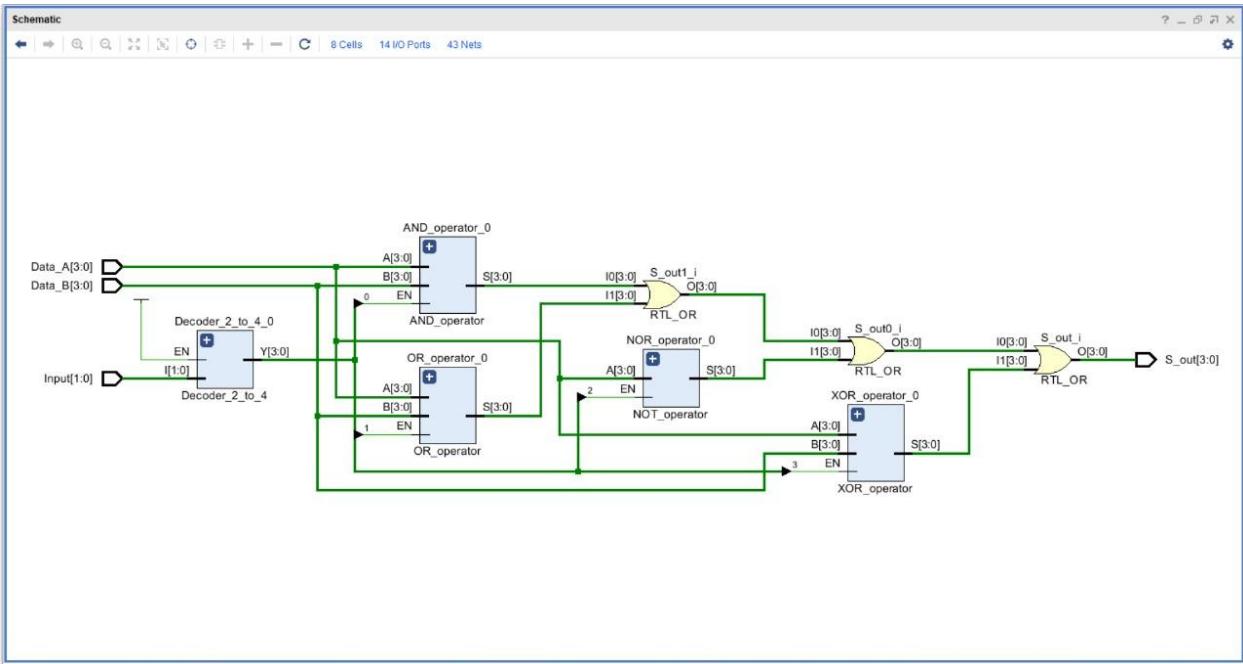
103      A => Data_A,  

104      EN => decoder_output(2),
105      S => NOT_operator_output);
106
107  XOR_operator_0: XOR_operator
108    PORT MAP(  

109      A => Data_A,
110      B => Data_B,
111      EN => decoder_output(3),
112      S => XOR_operator_output);
113
114
115
116
117
118
119   S_out <- AND_operator_output OR OR_operator_output OR NOT_operator_output OR XOR_operator_output;
120
121 end Behavioral;
122

```

2. Elaborated design schematic



3. Simulation source file

The screenshot shows a text editor window with the following details:

- Title Bar:** TB_Boolean_operator.vhd
- Path:** C:/Users/ralee/Desktop/micro processor/modified_1.0/project_1.2/project_1.2.srcs/sim_1/imports/new/TB_Boolean_operator.vhd
- Toolbar:** Includes icons for search, file operations, and other common functions.
- Code Content:** VHDL code for a test bench. The code includes comments for company, engineer, creation date, design name, module type, project name, target devices, tool versions, dependencies, revision, and additional comments. It also includes library declarations for IEEE and STD_LOGIC_1164, and component declarations for TB_Boolean_operator and Boolean_operator.

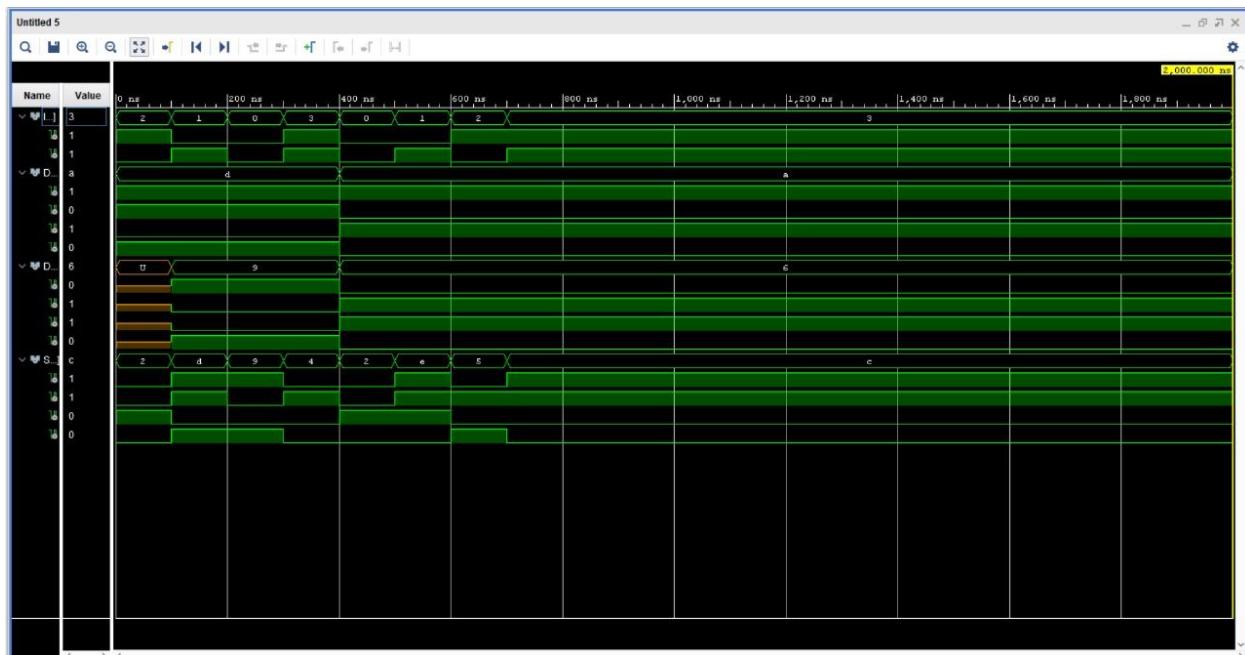
```
1 -- Company:  
2 -- Engineer:  
3 --  
4 -- Create Date: 04/24/2024 07:19:38 PM  
5 -- Design Name:  
6 -- Module Name: TB_Boolean_operator - Behavioral  
7 -- Project Name:  
8 -- Target Devices:  
9 -- Tool Versions:  
10 -- Description:  
11 --  
12 -- Dependencies:  
13 --  
14 -- Revision:  
15 -- Revision 0.01 - File Created  
16 -- Additional Comments:  
17 --  
18 --  
19 -- Uncomment the following library declaration if using  
20 -- arithmetic functions with Signed or Unsigned values  
21 --  
22 library IEEE;  
23 use IEEE.STD_LOGIC_1164.ALL;  
24 --  
25 -- Uncomment the following library declaration if instantiating  
26 -- arithmetic functions with Signed or Unsigned values  
27 --use IEEE.NUMERIC_STD.ALL;  
28 --  
29 -- Uncomment the following library declaration if instantiating  
30 -- any Xilinx leaf cells in this code.  
31 --library UNISIM;  
32 --use UNISIM.VComponents.all;  
33 --  
34 entity TB_Boolean_operator is  
35 -- Port ( );  
36 end TB_Boolean_operator;  
37 --  
38 architecture Behavioral of TB_Boolean_operator is  
39 --  
40 COMPONENT Boolean_operator is  
41 --  
42 --
```

```

40: COMPONENT boolean_operator IS
41:   Port ( Input : in STD_LOGIC_VECTOR (1 downto 0);
42:          Data_A : in STD_LOGIC_VECTOR (3 downto 0);
43:          Data_B : in STD_LOGIC_VECTOR (3 downto 0);
44:          S_out : out STD_LOGIC_VECTOR (3 downto 0));
45: end COMPONENT;
46:
47: signal Input : STD_LOGIC_VECTOR (1 downto 0);
48: signal Data_A,Data_B,S_out : STD_LOGIC_VECTOR (3 downto 0);
49:
50 begin
51
52  UUT: Boolean_operator PORT MAP(
53    Input => Input,
54    Data_A => Data_A,
55    Data_B => Data_B,
56    S_out => S_out
57  );
58
59  process
60  begin
61    Data_A <= "1101";
62    Input <= "10";
63
64    WAIT FOR 100 ns;
65    Data_B <= "1001";
66    Input <= "01";
67
68    WAIT FOR 100 ns;
69    Input <= "00";
70
71    WAIT FOR 100 ns;
72    Input <= "11";
73
74    WAIT FOR 100 ns;
75    Data_A <= "1010";
76    Data_B <= "0110";
77    Input <= "00";
78
79    WAIT FOR 100 ns;
80    -- ...
81
82    WAIT FOR 100 ns;
83    Input <= "10";
84
85    WAIT FOR 100 ns;
86    Input <= "11";
87
88    WAIT;
89  end process;
90
91 end Behavioral;
92

```

4. Timing Diagram



Multiplier

This component is able to multiply 4 bit signed numbers (-8 to 7). This component takes two 4bit vectors and returns an 8bit vector. We have used the unsigned multiplier designed in class and sent data through an interface. Through the interface we will convert signed values to unsigned values and then do the calculation. Then the result will be given as a signed value. If the result is in range between (0 to 7) it will be shown. Other results are incompatible with registers

1. Design source file

```
Top_Multiplier.vhd
C:/Users/raleel/OneDrive/Desktop/nano processor/modified_1.0/project_1.2/project_1.2.srsc/sources_1/imports/new/Top_Multiplier.vhd

library IEEE;
use ieee.numeric_std.all;
use IEEE.STD_LOGIC_1164.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
entity Top_Multiplier is
    Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
           B : in STD_LOGIC_VECTOR (3 downto 0);
           Y : out STD_LOGIC_VECTOR (7 downto 0));
end Top_Multiplier;
+<

Top_Multiplier.vhd
C:/Users/raleel/OneDrive/Desktop/nano processor/modified_1.0/project_1.2/project_1.2.srsc/sources_1/imports/new/Top_Multiplier.vhd

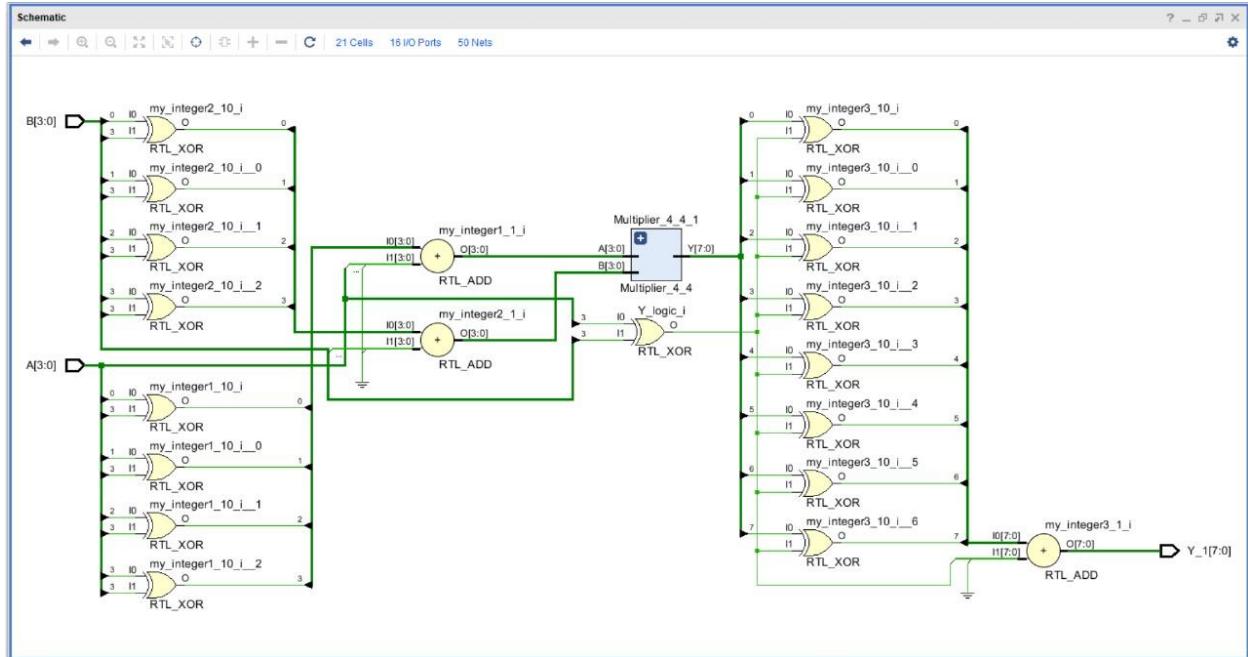
end Top_Multiplier;
architecture Behavioral of Top_Multiplier is
component Multiplier_4_4
    Port (
        A : in STD_LOGIC_VECTOR (3 downto 0);
        B : in STD_LOGIC_VECTOR (3 downto 0);
        Y : out STD_LOGIC_VECTOR (7 downto 0));
end component;
signal AA,AAA,BB,BBB : STD_LOGIC_VECTOR(3 DOWNTO 0);
signal my_integer1,my_integer2,my_integer3,my_integer4,my_integer5,my_integer6,my_integer7,A3,B3 : integer;
signal result_vector : std_logic_vector(3 DOWNTO 0);
signal Y,YY,YYY : STD_LOGIC_VECTOR (7 downto 0);
signal Y_logic : STD_LOGIC;
begin
function std_logic_to_integer(signal_value : std_logic) return integer is
begin
    if signal_value = '1' then
        return 1;
    else
        return 0;
    end if;
end function;
begin
process (AA,BB,A,B)
begin
    for i in 0 to 3 loop
        AA(i) <= A(i) XOR A(3);
        BB(i) <= B(i) XOR B(3);
    end loop;
end process;
A3 <- std_logic_to_integer(A(3));
B3 <- std_logic_to_integer(B(3));
my_integer1 <- to_integer(unsigned(AA));
my_integer1 <- my_integer1 + A3;
+<
```

```

79 : my_integer1_1 <= my_integer1 + A3;
80 : AAA <= std_logic_vector(to_unsigned(my_integer1_1, AA'length));
81 :
82 : my_integer2 <= to_integer(unsigned(BB));
83 : my_integer2_1 <= my_integer2 + B3;
84 : BBB <= std_logic_vector(to_unsigned(my_integer2_1, BB'length));
85 :
86 : Multiplexer_4_4_1 : Multiplier_4_4
87 : port map (
88 :   A => AAA,
89 :   B => BBB,
90 :   Y => YY
91 : );
92 :
93 : Y_logic <= A(3) XOR B(3);
94 : Y_int <= std_logic_to_integer(Y_logic);
95 :
96 : process (YY,Y)
97 : begin
98 :   for i in 0 to 7 loop
99 :     Y(i) <= YY(i) XOR Y_int;
100 :    end loop;
101 : end process;
102 :
103 : my_integer3 <= to_integer(unsigned(Y));
104 : my_integer3_1 <= my_integer3 + Y_int;
105 : YYY <= std_logic_vector(to_unsigned(my_integer3_1, Y'length));
106 :
107 : Y_1 <= YYY;
108 : end Behavioral;
109 :

```

2. Elaborated design schematic



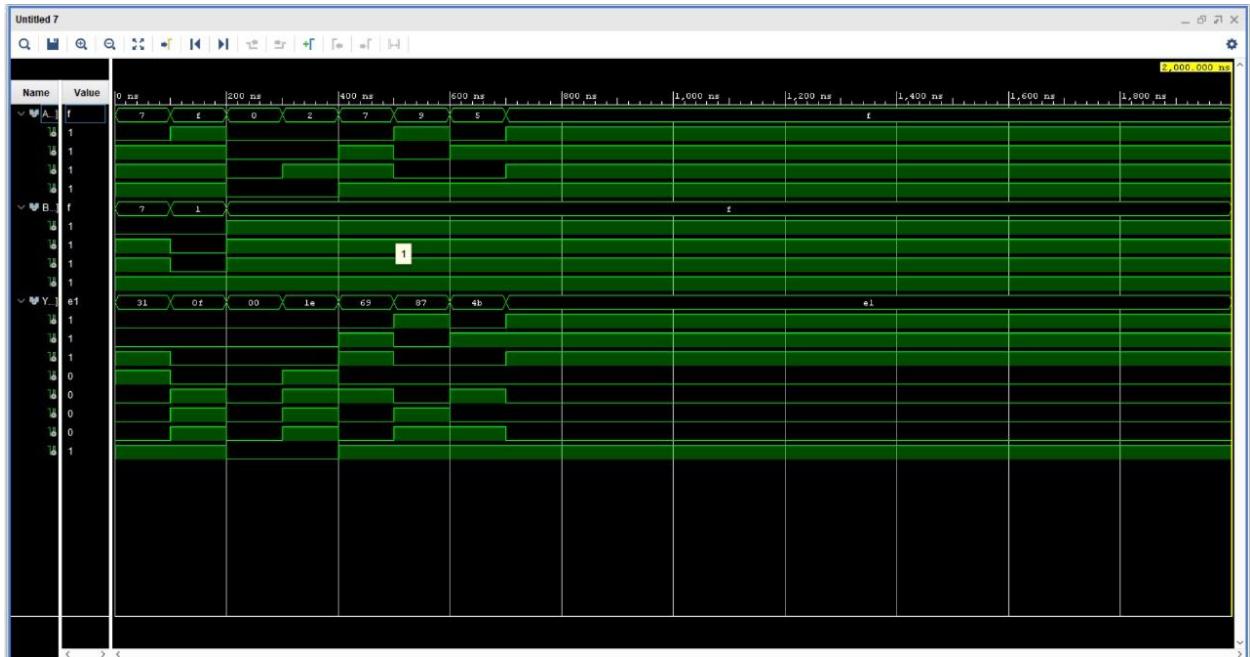
3. Simulation source file

TB_Multiplier_4.vhd

C:/Users/rakee/OneDrive/Desktop/nano processor/modified_1.0/project_1.2/project_1.2/srcs/sim_1/imports/new_TB_Multiplier_4.vhd

```
1 --> 
2 -- Company: 
3 -- Engineer: 
4 -- 
5 -- Create Date: 02/27/2024 09:33:56 PM 
6 -- Design Name: 
7 -- Module Name: TB_Multiplier_4 - Behavioral 
8 -- Project Name: 
9 -- Target Devices: 
10 -- Tool Versions: 
11 -- Description: 
12 -- 
13 -- Dependencies: 
14 -- 
15 -- Revision: 
16 -- Revision 0.01 - File Created 
17 -- Additional Comments: 
18 -- 
19 library IEEE; 
20 use IEEE.STD_LOGIC_1164.ALL; 
21 
22 -- 
23 -- Uncomment the following library declaration if using 
24 -- arithmetic functions with Signed or Unsigned values 
25 --use IEEE.NUMERIC_STD.ALL; 
26 
27 -- Uncomment the following library declaration if instantiating 
28 -- any Wishbone leaf cells in this code. 
29 --library UNISIM; 
30 --use UNISIM.VComponents.all; 
31 
32 entity TB_Multiplier_4 is 
33   -- Port () 
34 end TB_Multiplier_4; 
35 
36 architecture Behavioral of TB_Multiplier_4 is 
37   COMPONENT Multiplier_4_4 
38     Port ( A : in STD.LOGIC_VECTOR (3 downto 0); 
39            B : in STD.LOGIC_VECTOR (3 downto 0); 
40            Y : out STD.LOGIC_VECTOR (7 downto 0)); 
41   end COMPONENT; 
42 
43 begin 
44   UUT: Multiplier_4_4 PORT MAP( 
45     A => A, 
46     B => B, 
47     Y => Y 
48   ); 
49 
50   process 
51   begin 
52     process begin 
53       A <= "0111"; 
54       B <= "0111"; 
55       wait for 100ns; 
56       A <= "1111"; 
57       B <= "0001"; 
58       wait for 100ns; 
59       A <= "0000"; 
60       B <= "1111"; 
61       wait for 100ns; 
62       A <= "0010"; 
63       B <= "1111"; 
64       wait for 100ns; 
65       A <= "0111"; 
66       B <= "1111"; 
67       wait for 100ns; 
68       A <= "1001"; 
69       B <= "1111"; 
70       wait for 100ns; 
71       A <= "0101"; 
72       B <= "1111"; 
73       wait for 100ns; 
74       A <= "1111"; 
75       B <= "1111"; 
76 
77       wait; 
78   end process; 
79 end Behavioral;
```

4. Timing Diagram



4-1 four-bit Multiplexer

1. Design source file

```
Mux_4_to_1.vhd
C:/Users/raleel/Desktop/nano processor/modified_10/project_12/project_12/srcs/sources_1/imports/new/Mux_4_to_1.vhd

1 --> Company;
2 --> Engineer;
3 --> 
4 --> 
5 --> Create Date: 04/25/2024 03:58:54 PM
6 --> Design Name;
7 --> Module Name: Mux_4_to_1 - Behavioral
8 --> Project Name;
9 --> Target Devices;
10 --> Tool Versions;
11 --> Description;
12 --> 
13 --> Dependencies;
14 --> 
15 --> Revision;
16 --> Revision 0.01 - File Created
17 --> Additional Comments;
18 --> 

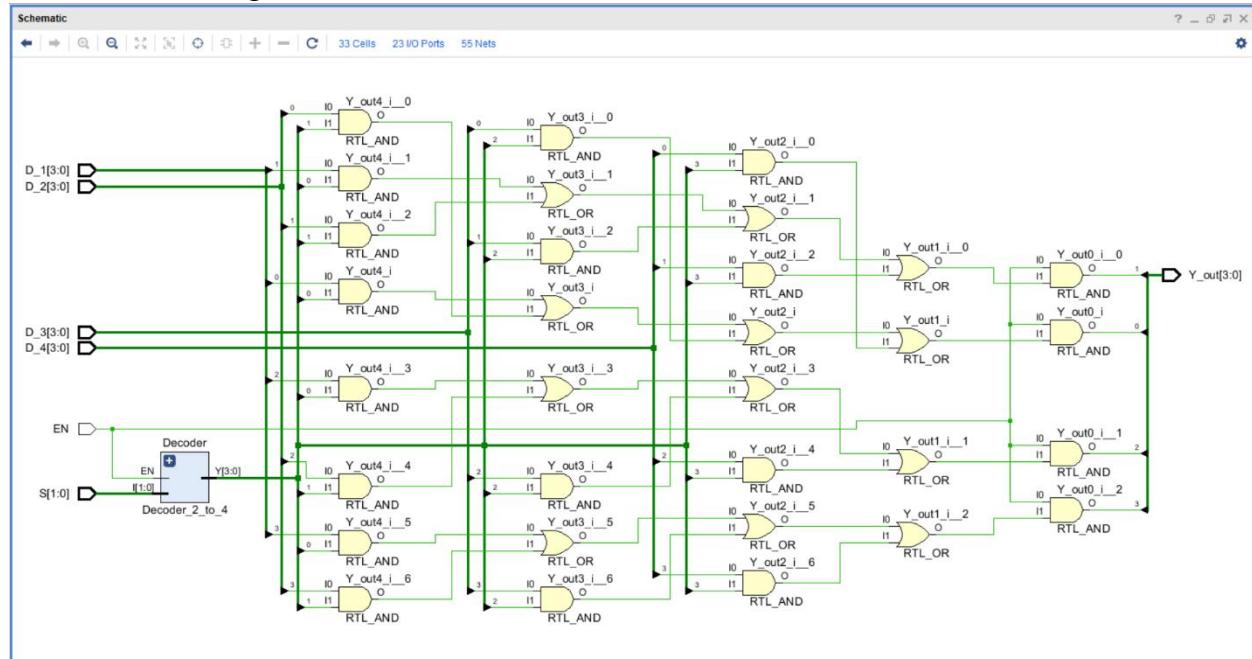
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
entity Mux_4_to_1 is
  Port ( S : in STD.LOGIC_VECTOR (1 downto 0);
         D_1 : in STD.LOGIC_VECTOR (3 downto 0);
         D_2 : in STD.LOGIC_VECTOR (3 downto 0);
         D_3 : in STD.LOGIC_VECTOR (3 downto 0);
         D_4 : in STD.LOGIC_VECTOR (3 downto 0);
         EN : in STD.LOGIC;
```

```

41 :     Y_out : out STD_LOGIC_VECTOR (3 downto 0);
42 : end Mux_4_to_1;
43 :
44 architecture Behavioral of Mux_4_to_1 is
45 :
46 component Decoder_2_to_4
47   port(
48     I : in STD_LOGIC_VECTOR(1 downto 0);
49     EN : in STD_LOGIC;
50     Y : out STD_LOGIC_VECTOR(3 downto 0) );
51 end component;
52 :
53 signal Decoder_Output : STD_LOGIC_VECTOR(3 downto 0);
54 :
55 begin
56
57 Decoder : Decoder_2_to_4
58 port map (
59   I => S,
60   EN => EN,
61   Y => Decoder_Output
62 );
63 Y_out(0) <= EN AND ((D_1(0) AND Decoder_Output(0)) OR (D_2(0) AND Decoder_Output(1)) OR (D_3(0) AND Decoder_Output(2)) OR (D_4(0) AND Decoder_Output(3)));
64 Y_out(1) <= EN AND ((D_1(1) AND Decoder_Output(0)) OR (D_2(1) AND Decoder_Output(1)) OR (D_3(1) AND Decoder_Output(2)) OR (D_4(1) AND Decoder_Output(3)));
65 Y_out(2) <= EN AND ((D_1(2) AND Decoder_Output(0)) OR (D_2(2) AND Decoder_Output(1)) OR (D_3(2) AND Decoder_Output(2)) OR (D_4(2) AND Decoder_Output(3)));
66 Y_out(3) <= EN AND ((D_1(3) AND Decoder_Output(0)) OR (D_2(3) AND Decoder_Output(1)) OR (D_3(3) AND Decoder_Output(2)) OR (D_4(3) AND Decoder_Output(3)));
67
68 end Behavioral;
69

```

2. Elaborated design schematic



3. Simulation source file

```

TB_Mux_4_to_1.vhd
C:/Users/raleel/OneDrive/Desktop/nano processor/New folder/4-way-4-bit-multiplexer/4-way-4-bit-multiplexer.srcs/sim_1/new/TB_Mux_4_to_1.vhd

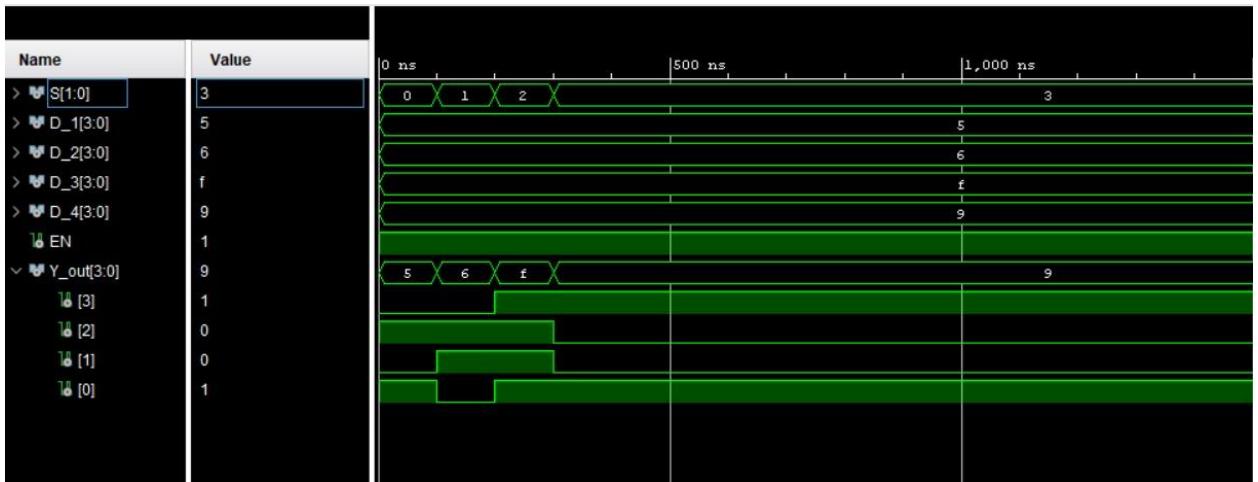
1 --> Company;
2 --> Engineer;
3 --> 
4 --> 
5 --> Create Date: 04/25/2024 04:09:47 PM
6 --> Design Name;
7 --> Module Name: TB_Mux_4_to_1 - Behavioral
8 --> Project Name;
9 --> Target Devices;
10 --> Tool Versions;
11 --> Description;
12 --> 
13 --> Dependencies;
14 --> 
15 --> Revision;
16 --> Revision 0.01 - File Created
17 --> Additional Comments;
18 --> 

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
library UNISIM;
--use UNISIM.VComponents.all;
entity TB_Mux_4_to_1 is
-- Port ();
end TB_Mux_4_to_1;

architecture Behavioral of TB_Mux_4_to_1 is
component Mux_4_to_1 is
    Port ( S : in STD_LOGIC_VECTOR (1 downto 0);
           D_1 : in STD_LOGIC_VECTOR (3 downto 0);
           D_2 : in STD_LOGIC_VECTOR (3 downto 0);
           D_3 : in STD_LOGIC_VECTOR (3 downto 0);
           D_4 : in STD_LOGIC_VECTOR (3 downto 0);
           EN : in STD_LOGIC;
           Y_out : out STD_LOGIC_VECTOR (3 downto 0));
end component;
begin
    signal S : STD_LOGIC_VECTOR (1 downto 0);
    signal D_1 : STD_LOGIC_VECTOR (3 downto 0);
    signal D_2 : STD_LOGIC_VECTOR (3 downto 0);
    signal D_3 : STD_LOGIC_VECTOR (3 downto 0);
    signal D_4 : STD_LOGIC_VECTOR (3 downto 0);
    signal EN : STD_LOGIC;
    signal Y_out : STD_LOGIC_VECTOR (3 downto 0);
begin
    UUT : Mux_4_to_1
        PORT MAP(
            S => S,
            D_1 => D_1,
            D_2 => D_2,
            D_3 => D_3,
            D_4 => D_4,
            EN => EN,
            Y_out => Y_out
        );
    process
    begin
        if EN = '1' then
            D_1 <= "0101";
            D_2 <= "0110";
            D_3 <= "1111";
            D_4 <= "1001";
            S <= "00";
            WAIT FOR 100 ns;
            S <= "01";
            WAIT FOR 100 ns;
            S <= "10";
            WAIT FOR 100 ns;
            S <= "11";
            WAIT;
        end if;
    end process;
end Behavioral;

```

4. Timing Diagram



Final Nanoprocessor

1. Design source file

```

NanoProcessor.vhd
C:/Users/radee/Desktop/nano processor/modified_10/project_12/project_12.srcs/sources_1/newNanoProcessor.vhd

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
entity NanoProcessor is
    Port ( Reset : in STD_LOGIC;
           Clk : in STD_LOGIC;
           Overflow : out STD_LOGIC;
           Zero : out STD_LOGIC;
           OUT1 : out STD_LOGIC;
           RT_LED : out STD_LOGIC_VECTOR (3 downto 0);

```

```

40 :     R7_LED : out STD_LOGIC_VECTOR (3 downto 0);
41 :     R7_Display: out STD_LOGIC_VECTOR(6 downto 0);
42 :     ComparatorOut : out STD_LOGIC_VECTOR (2 downto 0);
43 :     Anode : out STD_LOGIC_VECTOR(3 downto 0));
44 : end NanoProcessor;
45 
46 architecture Behavioral of NanoProcessor is
47 
48 component ProgramRom
49 
50     Port ( address : in STD_LOGIC_VECTOR (3 downto 0);
51             data : out STD_LOGIC_VECTOR (13 downto 0));
52 end component;
53 
54 component ProgramCounter
55 
56     Port ( D : in STD_LOGIC_VECTOR (3 downto 0);
57             Clk : in STD_LOGIC;
58             CLR : in STD_LOGIC;
59             Q : out STD_LOGIC_VECTOR (3 downto 0));
60 end component;
61 
62 component fourBitAdder
63 
64     Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
65             S : out STD_LOGIC_VECTOR (3 downto 0));
66 end component;
67 
68 component Slow_CLK
69 
70     Port ( Clk_in : in STD_LOGIC;
71             Clk_out : out STD_LOGIC);
72 end component;
73 
74 component Instruction_Dec
75 
76     Port ( OpCode: in STD_LOGIC_VECTOR(13 downto 0);
77             JumpRegisterValue; in STD_LOGIC_VECTOR(3 downto 0);
78             JumpFlag: out STD_LOGIC;
79             ComparatorFn : out STD_LOGIC;
80             JumpAddress: Out STD_LOGIC_VECTOR(3 downto 0);
81             AddOrSubtract : out STD_LOGIC;
82             MuxEnable1; Out STD_LOGIC;
83             MuxEnable2; Out STD_LOGIC;
84             MuxEnable2: Out STD_LOGIC;
85             iSnegative: out STD_LOGIC;
86             BooleanOp : out STD_LOGIC;
87             MuxControlSignal1: Out STD_LOGIC_VECTOR(2 downto 0);
88             MuxControlSignal2: Out STD_LOGIC_VECTOR(2 downto 0);
89             ImmediateValue : Out STD_LOGIC_VECTOR(3 downto 0);
90             Load: out STD_LOGIC_VECTOR(1 downto 0);
91             AddOrSubtractEnable : Out STD_LOGIC;
92             );
93 end component;
94 
95 component Reg_Bank
96 
97     Port ( D : in STD_LOGIC_VECTOR (3 downto 0);
98             I : in STD_LOGIC_VECTOR (2 downto 0);
99             Clk : in STD_LOGIC;
100            CLR : in STD_LOGIC;
101            R0 : out STD_LOGIC_VECTOR (3 downto 0);
102            R1 : out STD_LOGIC_VECTOR (3 downto 0);
103            R2 : out STD_LOGIC_VECTOR (3 downto 0);
104            R3 : out STD_LOGIC_VECTOR (3 downto 0);
105            R4 : out STD_LOGIC_VECTOR (3 downto 0);
106            R5 : out STD_LOGIC_VECTOR (3 downto 0);
107            R6 : out STD_LOGIC_VECTOR (3 downto 0);
108            R7 : out STD_LOGIC_VECTOR (3 downto 0));
109 end component;
110 
111 component Two_bit_MUX
112 
113     Port ( D1 : in STD_LOGIC_VECTOR (3 downto 0);
114             D2 : in STD_LOGIC_VECTOR (3 downto 0);
115             S : in STD_LOGIC;
116             EN : in STD_LOGIC;
117             Y : out STD_LOGIC_VECTOR (3 downto 0));
118 end component;
119 
120 component Mux_4_to_1
121 
122     Port ( S : in STD_LOGIC_VECTOR (1 downto 0);
123             D_1 : in STD_LOGIC_VECTOR (3 downto 0);
124             D_2 : in STD_LOGIC_VECTOR (3 downto 0);
125             D_3 : in STD_LOGIC_VECTOR (3 downto 0);
126             D_4 : in STD_LOGIC_VECTOR (3 downto 0);
127             EN : in STD_LOGIC;
128             );
129 end component;

```

```

118 :     EN : in STD_LOGIC;
119 :     Y_out : out STD_LOGIC_VECTOR (3 downto 0);
120 : end component;
121 :
122 : component Fourbit_Adder_Substrator
123 :     Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
124 :            B : in STD_LOGIC_VECTOR (3 downto 0);
125 :            S : out STD_LOGIC_VECTOR (3 downto 0);
126 :            En : in STD_LOGIC;
127 :            M : in STD_LOGIC;
128 :            V : out STD_LOGIC;
129 :            Zero : out STD_LOGIC;
130 :        );
131 : end component;
132 :
133 : component Mux_8_to_1
134 :     Port ( S : in STD_LOGIC_VECTOR (2 downto 0);
135 :            D_1 : in STD_LOGIC_VECTOR (3 downto 0);
136 :            D_2 : in STD_LOGIC_VECTOR (3 downto 0);
137 :            D_3 : in STD_LOGIC_VECTOR (3 downto 0);
138 :            D_4 : in STD_LOGIC_VECTOR (3 downto 0);
139 :            D_5 : in STD_LOGIC_VECTOR (3 downto 0);
140 :            D_6 : in STD_LOGIC_VECTOR (3 downto 0);
141 :            D_7 : in STD_LOGIC_VECTOR (3 downto 0);
142 :            D_8 : in STD_LOGIC_VECTOR (3 downto 0);
143 :            En : in STD_LOGIC;
144 :            Y_out : out STD_LOGIC_VECTOR (3 downto 0));
145 : end component;
146 :
147 : COMPONENT Two_3_bit_MUX
148 :     Port ( D1 : in STD_LOGIC_VECTOR (2 downto 0);
149 :            D2 : in STD_LOGIC_VECTOR (2 downto 0);
150 :            En : in STD_LOGIC;
151 :            S : in STD_LOGIC;
152 :            Y : OUT STD_LOGIC_VECTOR (2 downto 0));
153 : end COMPONENT;
154 :
155 : component SevenSegmentDisplay
156 :     Port ( Data : in STD_LOGIC_VECTOR (3 downto 0);
157 :            S_LED : out STD_LOGIC_VECTOR (3 downto 0);
158 :            S_Digit : out STD_LOGIC_VECTOR (3 downto 0);
159 :            S_Seg : out STD_LOGIC_VECTOR (6 downto 0));
160 :
161 : component Comparator
162 :     Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
163 :            B : in STD_LOGIC_VECTOR (3 downto 0);
164 :            En : in STD_LOGIC;
165 :            Greater : out STD_LOGIC;
166 :            Equal : out STD_LOGIC;
167 :            Less : out STD_LOGIC);
168 : end component;
169 :
170 : COMPONENT Top_Multiplier
171 :     Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
172 :            B : in STD_LOGIC_VECTOR (3 downto 0);
173 :            Y_1 : out STD_LOGIC_VECTOR (7 downto 0);
174 :        );
175 : end component;
176 :
177 : component Boolean_operator
178 :     Port ( Input_I : in STD_LOGIC_VECTOR (1 downto 0);
179 :            Data_A : in STD_LOGIC_VECTOR (3 downto 0);
180 :            Data_B : in STD_LOGIC_VECTOR (3 downto 0);
181 :            S_out : out STD_LOGIC_VECTOR (3 downto 0));
182 : end component;
183 :
184 : signal sel1, sel2, enable, compare: STD_LOGIC_VECTOR (2 downto 0);
185 : signal Load_from, bool: STD_LOGIC_VECTOR(1 DOWNTO 0);
186 : signal Instruction: STD_LOGIC_VECTOR (19 downto 0);
187 : signal reg, Slow_Clock, J_flag, operation, Mux_En1, Mux_En2, AddSubEn, CompEnable: STD_LOGIC;
188 : signal PC_in, J_Address, Address, PC, register1, register2, R_0, R_1, R_2, R_3, R_4, R_5, R_6, R_7, Value, result, data , LogicalBus : STD_LOGIC_VECTOR (3 downto 0);
189 : signal Helpbus: STD_LOGIC_VECTOR (7 downto 0);
190 :
191 : begin
192 :     InstructionDecoder: Instruction_Dec
193 :         port map (OpCode => Instruction,
194 :                   JumpRegisterValue => register1,
195 :                   JumpFlag => J_flag,
196 :                   JumpAddress => J_Address,
197 :                   ...
198 : );
199 : end;

```

```

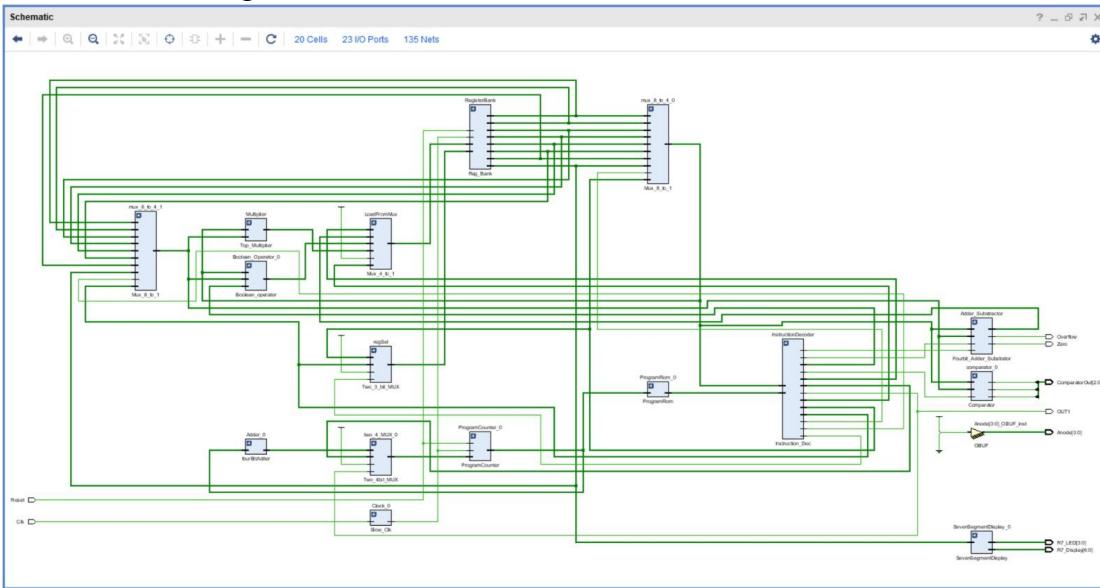
196 :         JumpAddress => J_Address,
197 :         AddrOrSubtract => operation,
198 :         MuxEnable1 => Mux_En1,
199 :         MuxEnable2 => Mux_En2,
200 :         MuxControlSignal1 => Sel1,
201 :         MuxControlSignal2 => Sel2,
202 :         ImmediateValue => Value,
203 :         Load => load_from,
204 :         BooleanOp => bool,
205 :         Isnegative => neg,
206 :         ComparatorEn => CompEnable,
207 :         AddSubtractEnable => AddSubEn);
208 :
209 : regSel :Two_3_bit_MUX
210 :     port map(
211 :         D1 =>Sel1,
212 :         D2 =>Sel2,
213 :         En =>'1',
214 :         S =>neg,
215 :         Y =>enable);
216 :
217 : Multiplier : Top_Multiplier
218 :     port map(
219 :         A =>register1,
220 :         B =>register2,
221 :         Y_1 =>MultipBus);
222 :
223 : comparator_0: Comparator
224 :     port map(
225 :         A =>register1,
226 :         B =>register2,
227 :         En =>CompEnable,
228 :         Greater =>compare(0),
229 :         Equal =>compare(1),
230 :         Less =>compare(2));
231 :
232 : Clock_0 : Slow_Clk
233 :     port map(
234 :         Clk_in => Clk,
235 :         Clk_out => Slow_Clock);
236 :
237 : Boolean_Operator_0 : Boolean_Operator
238 :     port map(
239 :         Input =>bool,
240 :         Data_A =>register1,
241 :         Data_B =>register2,
242 :         S_out =>LogicalBus);
243 :
244 : ProgramCounter_0 : ProgramCounter
245 :     port map(
246 :         D =>PC_in,
247 :         Clk =>Slow_Clock,
248 :         CLR =>Reset,
249 :         Q =>PC);
250 :
251 : Adder_0 :fourBitAdder
252 :     port map(
253 :         A => PC,
254 :         S => Address);
255 :
256 : ProgramRom_0 : ProgramRom
257 :     port map(
258 :         address =>PC,
259 :         data =>Instruction);
260 :
261 : two_4_MUX_0 :Two_4bit_MUX
262 :     port map(
263 :         D1 =>Address,
264 :         D2 =>J_Address,
265 :         EN =>'1',
266 :         S =>J_Flag,
267 :         Y =>PC_in);
268 :
269 : LoadFromMem :Mux_4_to_1
270 :     port map(
271 :         S => load_from,
272 :         D_1 =>Value,
273 :         D_2 =>result,
274 :         D_3 =>LogicalBus,
275 :         D_4 =>MultipBus(3 downto 0),
276 :         EN =>'1',
277 :         Y_out => data);
278 :
279 : RegisterBank: Reg_Bank
280 :     port map(
281 :         D =>data,
282 :         I =>enable,
283 :         Clk =>Slow_Clock,
284 :         CLR =>Reset,
285 :         R0 =>R_0,
286 :         R1 =>R_1,
287 :         R2 =>R_2,
288 :         R3 =>R_3,
289 :         R4 =>R_4,
290 :         R5 =>R_5,
291 :         R6 =>R_6,
292 :         R7 =>R_7);
293 :
294 : mux_8_to_4_0 :Mux_8_to_1
295 :     port map(
296 :         S =>sel1,
297 :         D_1 =>R_0,
298 :         D_2 =>R_1,
299 :         D_3 =>R_2,
300 :         D_4 =>R_3,
301 :         D_5 =>R_4,
302 :         D_6 =>R_5,
303 :         D_7 =>R_6,
304 :         D_8 =>R_7,
305 :         EN =>Mux_En1,
306 :         Y_out =>register1);
307 :
308 : mux_8_to_4_1 :Mux_8_to_1
309 :     port map(
310 :         S =>sel2,
311 :         D_1 =>R_0,
312 :         D_2 =>R_1,
313 :         D_3 =>R_2,
314 :         D_4 =>R_3,
315 :         D_5 =>R_4,
316 :         D_6 =>R_5,
317 :         D_7 =>R_6,
318 :         D_8 =>R_7,
319 :         EN =>Mux_En2,
320 :         Y_out =>register2);
321 :
322 : 
```

```

314 :      D_4    =>R_3,
315 :      D_5    =>R_4,
316 :      D_6    =>R_5,
317 :      D_7    =>R_6,
318 :      D_8    =>R_7,
319 :      EN     =>Mux_En2,
320 :      Y_out  =>register2);
321 :
322 :  Adder_Subtractor : Fourbit_Adder_Subtractor
323 :    port map(
324 :      A      =>register1,
325 :      B      =>register2,
326 :      S      =>result,
327 :      En     =>AddSubEn,
328 :      M      =>operation,
329 :      V      =>Overflow,
330 :      Zero   =>Zero);
331 :
332 :  SevenSegmentDisplay_0 : SevenSegmentDisplay
333 :    port map(
334 :      Data   =>R_7,
335 :      S_LED  =>R7_LED,
336 :      S_Deg  =>R7_Display);
337 :
338 :  GUT1 <=D_flag;
339 :  Anode <="1110";
340 :  ComparatorOut <= compare;  -->GREATERTHAN =>COMPARE(0)
341 :                                --EQUAL =>COMPARE(1)
342 :                                --LESSTHAN =>COMPARE(2)
343 : end Behavioral;

```

2. Elaborated design schematic



3. Simulation source file

```

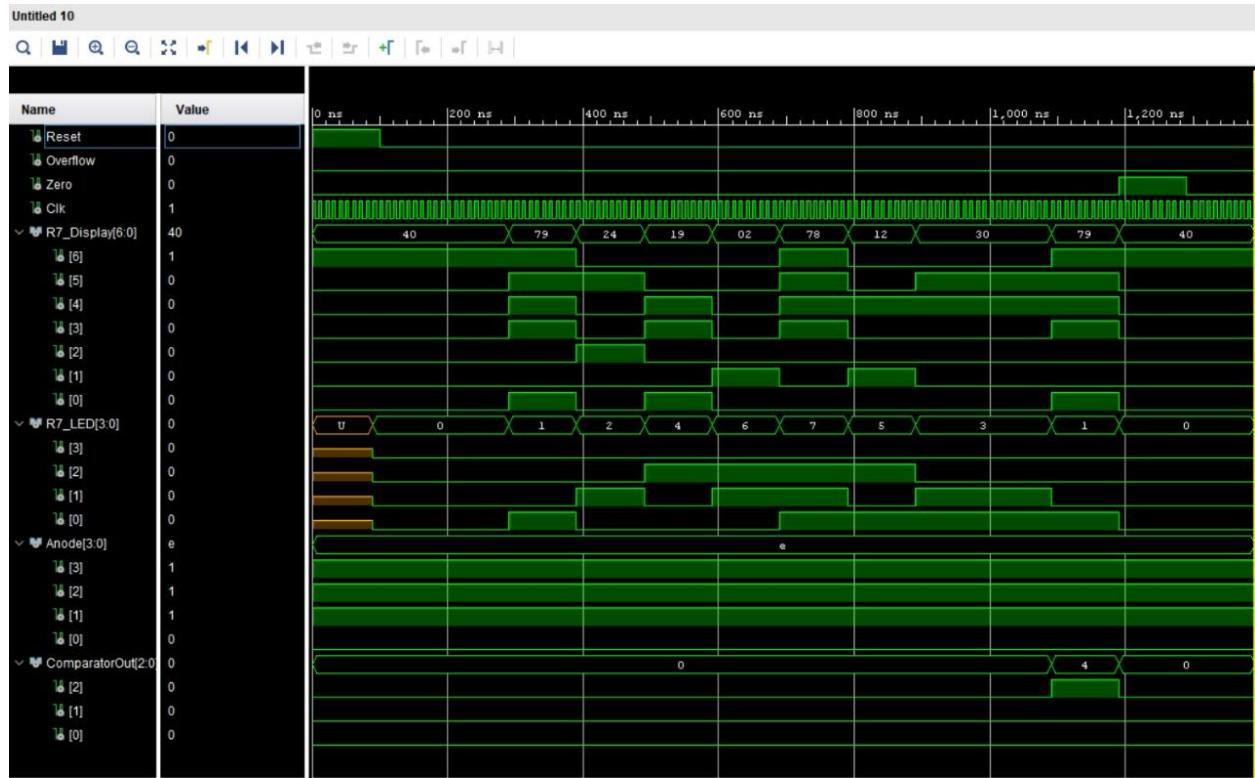
TB_NanoProcessor.vhd
C:/Users/radeeOneDrive/Desktop/nano processor/modified_1.0/project_1.2/src/sim_1/new/TB_NanoProcessor.vhd

Q | ━ | ← | → | X | // | ━ | Q |

1 -- Testbench for NanoProcessor
2
3 library IEEE;
4 use IEEE.STD_LOGIC_1164.ALL;
5
6 entity TB_NanoProcessor is
7 end TB_NanoProcessor;
8
9 architecture testbench of TB_NanoProcessor is
10
11 component NanoProcessor
12     Port ( Reset : in STD_LOGIC;
13             Clk : in STD_LOGIC;
14             Overflow : out STD_LOGIC;
15             Zero : out STD_LOGIC;
16             ComparatorOut : out STD_LOGIC_VECTOR (2 downto 0);
17             R7_LED : out STD_LOGIC_VECTOR (3 downto 0);
18             R7_Display: out STD_LOGIC_VECTOR(6 downto 0);
19             Anode: out STD_LOGIC_VECTOR(3 downto 0));
20
21 end component;
22
23 signal Reset, Overflow, Zero : std_logic;
24 signal Clk : STD_LOGIC := '0';
25 signal R7_Display: STD_LOGIC_VECTOR(6 downto 0);
26 signal R7_LED, Anode : STD_LOGIC_VECTOR (3 downto 0);
27 signal ComparatorOut : STD_LOGIC_VECTOR (2 downto 0);
28
29 begin
30
31     UUT : NanoProcessor
32         port map (
33             Reset => Reset,
34             Clk => Clk,
35             Overflow => Overflow,
36             Zero => Zero,
37             R7_LED => R7_LED,
38             R7_Display => R7_Display,
39             ComparatorOut => ComparatorOut,
40             Anode => Anode
41         );
42
43     clk_proc: process
44     begin
45         while now < 3000 ns loop
46             Clk <= not Clk;
47             wait for 5 ns;
48         end loop;
49         wait;
50     end process;
51
52     stimuli: process
53     begin
54
55         Reset <= '1';
56         wait for 100 ns;
57
58         Reset <='0';
59         wait for 400 ns;
60
61         --Reset <='1';
62         --wait for 100 ns;
63
64         Reset <= '0';
65         wait;
66
67     end process;
68
69 end testbench;
70

```

4. Timing Diagram



Constraints file – Modified version

```

Basys3Labs.xdc
C:/Users/alee/OneDrive/Desktop/nano processor/modified_1.0/project_1.2/project_1.2.srsc/constrs_1/new/Basys3Labs.xdc

Q | L | A | P | X | E | D | F | T | W | M | S | Q |
1 ## This file is a general .xdc for the Basys3 rev B board
2 ## To use it in a project:
3 ## - uncomment the lines corresponding to used pins
4 ## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project
5
6 ## Clock signal
7 set_property PACKAGE_PIN W5 [get_ports {Clk}]
8 set_property IOSTANDARD LVCMOS33 [get_ports {Clk}]
9 create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports {Clk}]
10
11 ## Switches
12 set_property PACKAGE_PIN V17 [get_ports {A}]
13 #set_property IOSTANDARD LVCMOS33 [get_ports {A}]
14 set_property PACKAGE_PIN V16 [get_ports {B}]
15 #set_property IOSTANDARD LVCMOS33 [get_ports {B}]
16 set_property PACKAGE_PIN W16 [get_ports {C}]
17 #set_property IOSTANDARD LVCMOS33 [get_ports {C}]
18 set_property PACKAGE_PIN W17 [get_ports {sv[3]}]
19 #set_property IOSTANDARD LVCMOS33 [get_ports {sv[3]}]
20 set_property PACKAGE_PIN W18 [get_ports {sv[4]}]
21 #set_property IOSTANDARD LVCMOS33 [get_ports {sv[4]}]
22 set_property PACKAGE_PIN V18 [get_ports {sv[5]}]
23 #set_property IOSTANDARD LVCMOS33 [get_ports {sv[5]}]
24 set_property PACKAGE_PIN W14 [get_ports {sv[6]}]
25 #set_property IOSTANDARD LVCMOS33 [get_ports {sv[6]}]
26 set_property PACKAGE_PIN W13 [get_ports {sv[7]}]
27 #set_property IOSTANDARD LVCMOS33 [get_ports {sv[7]}]
28 set_property PACKAGE_PIN V2 [get_ports {sv[8]}]
29 #set_property IOSTANDARD LVCMOS33 [get_ports {sv[8]}]
30 set_property PACKAGE_PIN T3 [get_ports {sv[9]}]
31 #set_property IOSTANDARD LVCMOS33 [get_ports {sv[9]}]
32 set_property PACKAGE_PIN T2 [get_ports {sv[10]}]
33 #set_property IOSTANDARD LVCMOS33 [get_ports {sv[10]}]
34 set_property PACKAGE_PIN R3 [get_ports {sv[11]}]
35 #set_property IOSTANDARD LVCMOS33 [get_ports {sv[11]}]
36 set_property PACKAGE_PIN W2 [get_ports {sv[12]}]
37 #set_property IOSTANDARD LVCMOS33 [get_ports {sv[12]}]
38 set_property PACKAGE_PIN V1 [get_ports {sv[13]}]
39 #set_property IOSTANDARD LVCMOS33 [get_ports {sv[13]}]
40 set_property PACKAGE_PIN T1 [get_ports {sv[14]}]
<----- EXPRESSIONS ----->

```

```

40 : #set_property PACKAGE_PIN T1 [get_ports {sv[14]}]
41 :     #set_property IOSTANDARD LVCMS33 [get_ports {sv[14]}]
42 : #set_property PACKAGE_PIN R2 [get_ports {sv[15]}]
43 :     #set_property IOSTANDARD LVCMS33 [get_ports {sv[15]}]
44 :
45 :
46 : ## LEDs
47 : #set_property PACKAGE_PIN U16 [get_ports {R7_LED[0]}]
48 :     #set_property IOSTANDARD LVCMS33 [get_ports {R7_LED[0]}]
49 : #set_property PACKAGE_PIN K19 [get_ports {R7_LED[1]}]
50 :     #set_property IOSTANDARD LVCMS33 [get_ports {R7_LED[1]}]
51 : #set_property PACKAGE_PIN U19 [get_ports {R7_LED[2]}]
52 :     #set_property IOSTANDARD LVCMS33 [get_ports {R7_LED[2]}]
53 : #set_property PACKAGE_PIN V19 [get_ports {R7_LED[3]}]
54 :     #set_property IOSTANDARD LVCMS33 [get_ports {R7_LED[3]}]
55 : #set_property PACKAGE_PIN W18 [get_ports {R7_LED[4]}]
56 :     #set_property IOSTANDARD LVCMS33 [get_ports {R7_LED[4]}]
57 : #set_property PACKAGE_PIN U15 [get_ports {led[5]}]
58 :     #set_property IOSTANDARD LVCMS33 [get_ports {led[5]}]
59 : #set_property PACKAGE_PIN U14 [get_ports {led[6]}]
60 :     #set_property IOSTANDARD LVCMS33 [get_ports {led[6]}]
61 : #set_property PACKAGE_PIN V14 [get_ports {led[7]}]
62 :     #set_property IOSTANDARD LVCMS33 [get_ports {led[7]}]
63 : #set_property PACKAGE_PIN V13 [get_ports {led[8]}]
64 :     #set_property IOSTANDARD LVCMS33 [get_ports {led[8]}]
65 : #set_property PACKAGE_PIN V3 [get_ports {ComparatorOut[0]}]
66 :     #set_property IOSTANDARD LVCMS33 [get_ports {ComparatorOut[0]}]
67 : #set_property PACKAGE_PIN W3 [get_ports {ComparatorOut[1]}]
68 :     #set_property IOSTANDARD LVCMS33 [get_ports {ComparatorOut[1]}]
69 : #set_property PACKAGE_PIN U3 [get_ports {ComparatorOut[2]}]
70 :     #set_property IOSTANDARD LVCMS33 [get_ports {ComparatorOut[2]}]
71 : #set_property PACKAGE_PIN P3 [get_ports {led[12]}]
72 :     #set_property IOSTANDARD LVCMS33 [get_ports {led[12]}]
73 : #set_property PACKAGE_PIN N3 [get_ports {led[13]}]
74 :     #set_property IOSTANDARD LVCMS33 [get_ports {led[13]}]
75 : #set_property PACKAGE_PIN P1 [get_ports {Zero}]
76 :     #set_property IOSTANDARD LVCMS33 [get_ports {Zero}]
77 : #set_property PACKAGE_PIN L1 [get_ports {Overflow}]
78 :     #set_property IOSTANDARD LVCMS33 [get_ports {Overflow}]
79 :
80 :
81 : ## segment display
82 : #set_property PACKAGE_PIN W7 [get_ports {R7_Display[0]}]
83 :     #set_property IOSTANDARD LVCMS33 [get_ports {R7_Display[0]}]
84 : #set_property PACKAGE_PIN W6 [get_ports {R7_Display[1]}]
85 :     #set_property IOSTANDARD LVCMS33 [get_ports {R7_Display[1]}]
86 : #set_property PACKAGE_PIN U8 [get_ports {R7_Display[2]}]
87 :     #set_property IOSTANDARD LVCMS33 [get_ports {R7_Display[2]}]
88 : #set_property PACKAGE_PIN V8 [get_ports {R7_Display[3]}]
89 :     #set_property IOSTANDARD LVCMS33 [get_ports {R7_Display[3]}]
90 : #set_property PACKAGE_PIN U5 [get_ports {R7_Display[4]}]
91 :     #set_property IOSTANDARD LVCMS33 [get_ports {R7_Display[4]}]
92 : #set_property PACKAGE_PIN V5 [get_ports {R7_Display[5]}]
93 :     #set_property IOSTANDARD LVCMS33 [get_ports {R7_Display[5]}]
94 : #set_property PACKAGE_PIN U7 [get_ports {R7_Display[6]}]
95 :     #set_property IOSTANDARD LVCMS33 [get_ports {R7_Display[6]}]
96 :
97 : #set_property PACKAGE_PIN V7 [get_ports dp]
98 :     #set_property IOSTANDARD LVCMS33 [get_ports dp]
99 :
100 : #set_property PACKAGE_PIN U2 [get_ports {Anode[0]}]
101 :     #set_property IOSTANDARD LVCMS33 [get_ports {Anode[0]}]
102 : #set_property PACKAGE_PIN U4 [get_ports {Anode[1]}]
103 :     #set_property IOSTANDARD LVCMS33 [get_ports {Anode[1]}]
104 : #set_property PACKAGE_PIN V4 [get_ports {Anode[2]}]
105 :     #set_property IOSTANDARD LVCMS33 [get_ports {Anode[2]}]
106 : #set_property PACKAGE_PIN W4 [get_ports {Anode[3]}]
107 :     #set_property IOSTANDARD LVCMS33 [get_ports {Anode[3]}]
108 :
109 :
110 : ##Buttons
111 : #set_property PACKAGE_PIN U10 [get_ports Reset]
112 :     #set_property IOSTANDARD LVCMS33 [get_ports Reset]
113 : #set_property PACKAGE_PIN T18 [get_ports btnD]
114 :     #set_property IOSTANDARD LVCMS33 [get_ports btnD]
115 : #set_property PACKAGE_PIN W19 [get_ports btnL]
116 :     #set_property IOSTANDARD LVCMS33 [get_ports btnL]
117 : #set_property PACKAGE_PIN T17 [get_ports btnR]
118 :     #set_property IOSTANDARD LVCMS33 [get_ports btnR]

```

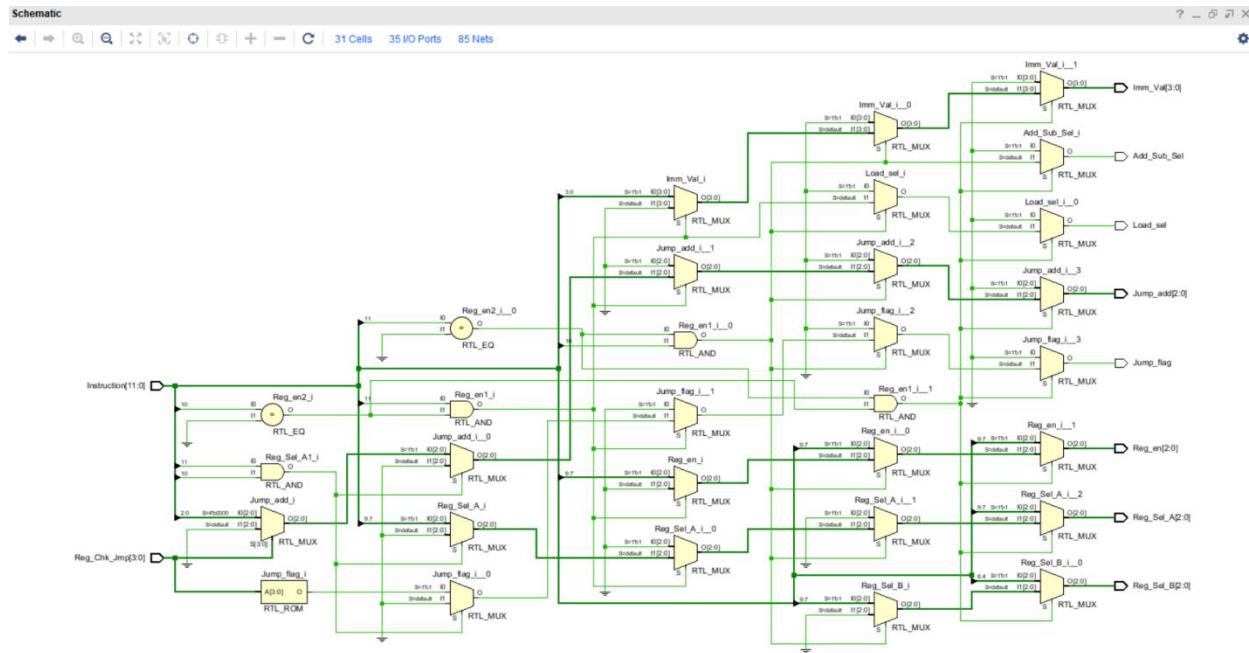
Resource Utilization and Optimization

● Multiplexers

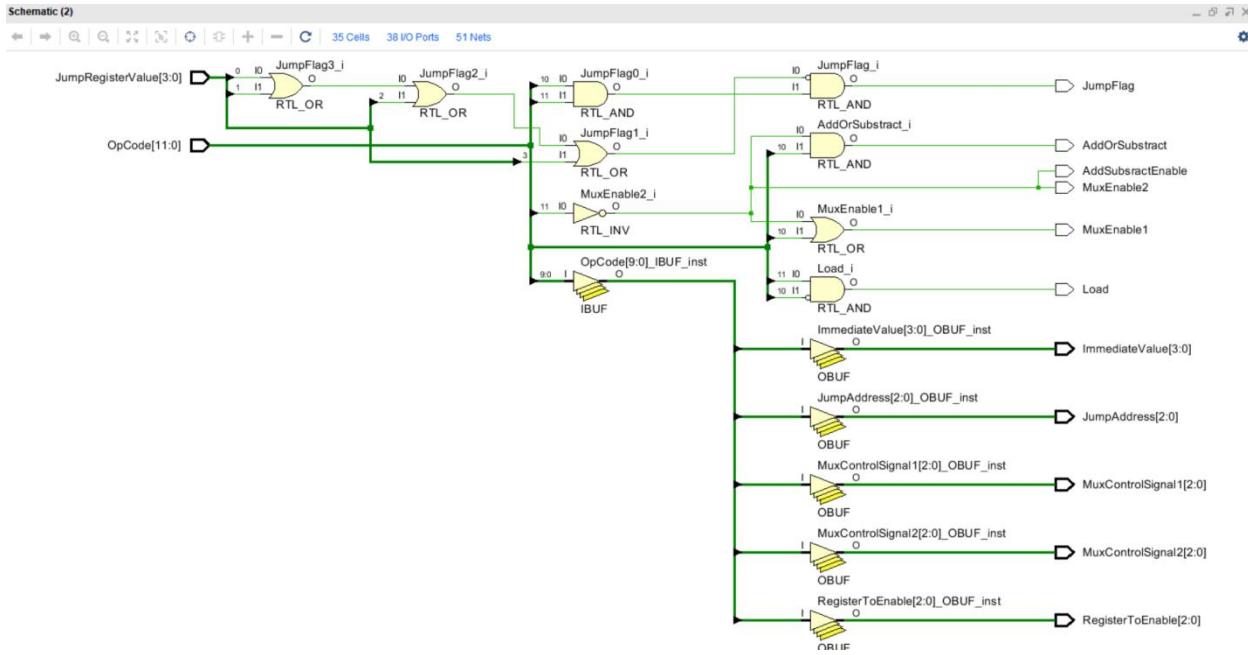
- To design the multiplexer, we used a separate decoder component. This provides a modular approach, and modularity promotes reusability and simplifies design maintenance.
- Boolean expressions are used to implement the multiplexer functionality. This is more resource-efficient compared to using if-else statements.
- The multiplexer can be turned on or off based on the enable signal (EN). When it's turned off, it doesn't use up resources unnecessarily.

Instruction Decoder

- Instruction decoder is designed using basic logic gates. We first designed it using if-else statements which is supported in vhdl and found out that it would use too much resources. It uses a significant amount of multiplexers and other resources. (For you too compare the RTL designs of each design approach is given below)
- Instruction decoder only enables the necessary components needed to execute a command. For an example adder subtractor unit will be enabled only in ADD and NEG instructions in the base version of the nano processor. This helps to reduce power consumption of the nano processor.
- By using Karnaugh maps the the logic related to various instructions were simplified to minimize the number of logic gates used.



The instruction decoder designed using if-else statements



The instruction decoder designed using basic logic gates.

- Seven Segment

- We used a LUT to store the configurations for the seven-segment display. This approach is resource-efficient because LUTs store precomputed outputs for given inputs. Therefore, when we need them, the LUT simply retrieves the output based on the input address.
- Seven segment display is designed with modularity. This promotes code readability and maintainability.

- Boolean operator

- I used a LUT to store the configurations for the seven-segment display. This approach is resource-efficient because LUTs store precomputed outputs for given inputs. Therefore, when we need them, the LUT simply retrieves the output based on the input address.
- Seven segment display is designed with modularity. This promotes code readability and maintainability.

- Comparator

- The outputs (Greater, Equal, Less) are conditioned on the enable signal (EN), ensuring that the comparison results are only generated when needed. This prevents unnecessary resource consumption

- The design employs optimized Boolean logic expressions to compute the comparison results efficiently. This minimize unnecessary logic gates and resources.

- Nano Processor

- Apart from this main entities we have used the following.
- We have used two lookup tables. One in the program ROM and the other in seven segment display.
- In the base version two 8-1 four bit MUXs, one 2-1 three bit and four bit MUXs are used.
- One 2-4 decoder and and 3-8 decoder is used in 8-1 MUX.
- One 3-8 decoder is used in the register bank.
- WE have used D flip-flops in the register bank and in the program counter.
- The adder subtractor unit is made using full adders, which are made using half adders.

Additional features and Improvements

Implemented Improvements

- We replaced the three bit adder with a four bit adder so that the processor with support 16 instructions in the ROM rather than 8. To support this functionality we had to increase the capability of storage of program counter to 4 bits and make the two way three bit mux into a two way four bit mux.
- We added three new entities call comparator, multiplier and boolean operator. Comparaor will compare two values in two registers and output whether one is greater than, lesser than or equal with respect to a another. The multiplier added the function of multiplication to the processor. The boolean operator entity supports 4 bit wise operators.
- With the boolean operator the processor can get the bit wise AND, OR, XOR of the values in two registers and the NOT value of one register at a time.
- We have increased the size of a machine code instruction to 14 bits and upgraded the instruction decoder accordingly to support these new functionality and bit size. This would help us to increase the functionality 16 instructions.
- With these new and improved entities, we have extended the number of supported machine instructions to 13.

Improvements that can be made but not implemented

- We can increase the size of the registers to support representation of more numbers. But that would require to change the sizes of MUXs, data buses and arithmetic operation units.

Problems encountered and solutions

- Implementing the instruction decoder using if else statements consume large amount of resources. So we decided to make as much as using basic logic gates.
- Buses were connected in a wrong way at first. So the processor didn't work as expected at first. We could correct them using RTL schematic.
- Some logic equations were wrong. We found the error by simulation and corrected them using karnaugh maps.
- There were errors in the constraints file because of using wrong type brackets. We could correct them from looking up in the internet.

Characterization details

Non-modified version

- Resource Utilization

Name	1	Slice LUTs (20800)	Slice Registers (41600)	Slice (815 0)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	Bonded IOB (106)	BUFGCTRL (32)
▽ NanoProcessor		46	59	26	46	8	19	1
└ Clock0 (Slow_Clk)		11	34	17	11	3	0	0
└ ProgramCounter0 (Pro...		31	3	11	31	2	0	0
└ D_FF_0 (D_FF_23)		18	1	10	18	1	0	0
└ D_FF_1 (D_FF_24)		10	1	6	10	0	0	0
└ D_FF_2 (D_FF_25)		3	1	3	3	0	0	0
└ RegisterBank (Reg_B...		4	22	10	4	0	0	0
> Reg_2 (Reg)		0	4	2	0	0	0	0
> Reg_3 (Reg_0)		0	4	1	0	0	0	0
> Reg_5 (Reg_1)		0	4	2	0	0	0	0
> Reg_6 (Reg_2)		0	4	3	0	0	0	0
> Reg_7 (Reg_3)		4	6	4	4	0	0	0

Start RTL Component Statistics

Detailed RTL Component Info :

+---Adders :

2 Input 32 Bit Adders := 1

+---XORs:
 2 Input 1 Bit XORs:= 33
+---Registers:
 32 Bit Registers:= 1
 1 Bit Registers:= 33
+---Muxes :
 2 Input 32 Bit Muxes := 1
 7 Input 12 Bit Muxes := 1
 9 Input 7 Bit Muxes := 1
 2 Input 1 Bit Muxes := 1

Finished RTL Component Statistics

Start RTL Hierarchical Component Statistics

Hierarchical RTL Component report

Module Slow_Clk

Detailed RTL Component Info :

+---Adders :

 2 Input 32 Bit Adders := 1

+---Registers :

 32 Bit Registers := 1

 1 Bit Registers := 2

+---Muxes :

 2 Input 32 Bit Muxes := 1

 2 Input 1 Bit Muxes := 1

Module D_FF

Detailed RTL Component Info :

+---Registers :

 1 Bit Registers := 1

Module HA

Detailed RTL Component Info :

+---XORs :

 2 Input 1 Bit XORs := 1

Module FA

Detailed RTL Component Info :

+---XORs :

 2 Input 1 Bit XORs := 2

Module ProgramRom

Detailed RTL Component Info :

+---Muxes :

 7 Input 12 Bit Muxes := 1

Module Fourbit_Adder_Substrator

Detailed RTL Component Info :

+---XORs :

 2 Input 1 Bit XORs := 5

Module LUT_Display

Detailed RTL Component Info :

+---Muxes :

 9 Input 7 Bit Muxes := 1

Finished RTL Hierarchical Component Statistics

● Power Consumption Report

```
| Tool Version   : Vivado v.2018.1 (win64) Build 2188600 Wed Apr 4 18:40:38 MDT 2018
| Date          : Sun May 5 10:52:06 2024
| Host          : Maduni running 64-bit major release (build 9200)
| Command       : report_power -file NanoProcessor_power_routed.rpt -pb
NanoProcessor_power_summary_routed.pb -rpx NanoProcessor_power_routed.rpx
| Design        : NanoProcessor
| Device        : xc7a35tcpg236-1
| Design State  : routed
| Grade         : commercial
| Process       : typical
| Characterization : Production
```

Power Report

Table of Contents

1. Summary

1.1 On-Chip Components

1.2 Power Supply Summary

1.3 Confidence Level

2. Settings

2.1 Environment

2.2 Clock Constraints

3. Detailed Reports

3.1 By Hierarchy

1. Summary

Total On-Chip Power (W)	0.076
-------------------------	-------

Design Power Budget (W)	Unspecified*
Power Budget Margin (W)	NA
Dynamic (W)	0.004
Device Static (W)	0.072
Effective TJA (C/W)	5.0
Max Ambient (C)	84.6
Junction Temperature (C)	25.4
Confidence Level	Low
Setting File	----
Simulation Activity File	----
Design Nets Matched	NA

* Specify Design Power Budget using, set_operating_conditions -design_power_budget <value> in Watts>

1.1 On-Chip Components

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	<0.001	3	----	----
Slice Logic	<0.001	135	----	----
LUT as Logic	<0.001	46	20800	0.22
CARRY4	<0.001	8	8150	0.10
Register	<0.001	59	41600	0.14
Others	0.000	10	----	----
Signals	<0.001	119	----	----
I/O	0.002	19	106	17.92
Static Power	0.072			
Total	0.076			

1.2 Power Supply Summary

Source	Voltage (V)	Total (A)	Dynamic (A)	Static (A)
Vccint	1.000	0.011	0.002	0.010
Vceaux	1.800	0.013	0.000	0.013
Vcco33	3.300	0.002	0.001	0.001
Vcco25	2.500	0.000	0.000	0.000
Vcco18	1.800	0.000	0.000	0.000
Vcco15	1.500	0.000	0.000	0.000
Vcco135	1.350	0.000	0.000	0.000
Vcco12	1.200	0.000	0.000	0.000
Vceaux_io	1.800	0.000	0.000	0.000
Vccbram	1.000	0.000	0.000	0.000
MGTAVcc	1.000	0.000	0.000	0.000
MGTAVtt	1.200	0.000	0.000	0.000
Vccadc	1.800	0.020	0.000	0.020

1.3 Confidence Level

User Input Data	Confidence	Details	Action
Design implementation state	High	Design is routed	
Clock nodes activity	Low	User specified less than 75% of clocks	Provide missing clock activity with a constraint file, simulation results or by editing the "By Clock Domain" view
I/O nodes activity	Medium	More than 5% of inputs are missing user specification	Provide missing input activity with simulation results or by editing the "By Resource Type -> I/Os" view

Internal nodes activity	Medium	User specified less than 25% of internal nodes	Provide missing internal nodes activity with simulation results or by editing the "By Resource Type" views
Device models	High	Device models are Production	
Overall confidence level	Low		

2. Settings

2.1 Environment

Ambient Temp (C)	25.0
ThetaJA (C/W)	5.0
Airflow (LFM)	250
Heat Sink	medium (Medium Profile)
ThetaSA (C/W)	4.6
Board Selection	medium (10"x10")
# of Board Layers	12to15 (12 to 15 Layers)
Board Temperature (C)	25.0

2.2 Clock Constraints

Clock	Domain	Constraint (ns)
sys_clk_pin	Clk	10.0

3. Detailed Reports

3.1 By Hierarchy

Name	Power (W)
NanoProcessor	0.004
Clock0	<0.001
	<0.001
ProgramCounter0	
D_FF_0	<0.001
D_FF_1	<0.001
D_FF_2	<0.001
RegisterBank	0.000
Reg_2	0.000
D_FF_0	0.000
D_FF_1	0.000
D_FF_2	0.000
D_FF_3	0.000
Reg_3	0.000
D_FF_0	0.000
D_FF_1	0.000
D_FF_2	0.000
D_FF_3	0.000
Reg_5	0.000
D_FF_0	0.000
D_FF_1	0.000
D_FF_2	0.000
D_FF_3	0.000
Reg_6	0.000
D_FF_0	0.000
D_FF_1	0.000
D_FF_2	0.000

D_FF_3	0.000
Reg_7	0.000
D_FF_0	0.000
D_FF_1	0.000
D_FF_2	0.000
D_FF_3	0.000

Modified version

- Resource Utilization

Name	1	Slice LUTs (20800)	Slice Registers (41600)	Slice (8150)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	Bonded IOB (106)	BUFGCTRL (32)
▼ N NanoProcessor		74	56	34	74	9	22	1
└ Clock_0 (Slow_Clk)		11	34	17	11	2	0	0
└ ProgramCounter_0 (Pr...		59	4	17	59	2	0	0
└ D_FF_0 (D_FF_18)		5	1	5	5	0	0	0
└ D_FF_1 (D_FF_19)		46	1	16	46	1	0	0
└ D_FF_2 (D_FF_20)		1	1	1	1	0	0	0
└ D_FF_3 (D_FF_21)		7	1	5	7	0	0	0
└ RegisterBank (Reg_B...		4	18	7	4	0	0	0
> Reg_2 (Reg)		0	4	1	0	0	0	0
> Reg_3 (Reg_0)		0	4	1	0	0	0	0
> Reg_6 (Reg_1)		0	4	1	0	0	0	0
> Reg_7 (Reg_2)		4	6	4	4	0	0	0

Start RTL Component Statistics

Detailed RTL Component Info :

+---Adders :

```
2 Input 32 Bit Adders := 1
2 Input 8 Bit Adders := 1
2 Input 4 Bit Adders := 2
```

+---XORs :

```
2 Input 1 Bit XORs := 162
```

+---Registers :

```
32 Bit Registers := 1
1 Bit Registers := 30
```

+---Muxes :

2 Input	32 Bit	Muxes := 1
13 Input	14 Bit	Muxes := 1
9 Input	7 Bit	Muxes := 1
2 Input	1 Bit	Muxes := 1

Finished RTL Component Statistics

Start RTL Hierarchical Component Statistics

Hierarchical RTL Component report

Module HA

Detailed RTL Component Info :

+---XORs :

2 Input	1 Bit	XORs := 1
---------	-------	-----------

Module FA

Detailed RTL Component Info :

+---XORs :

2 Input	1 Bit	XORs := 2
---------	-------	-----------

Module Top_Multiplier

Detailed RTL Component Info :

+---Adders :

2 Input	8 Bit	Adders := 1
---------	-------	-------------

2 Input	4 Bit	Adders := 2
---------	-------	-------------

+---XORs :

2 Input	1 Bit	XORs := 17
---------	-------	------------

Module Comparator

Detailed RTL Component Info :

+---XORs :

2 Input	1 Bit	XORs := 4
---------	-------	-----------

Module Slow_Clk

Detailed RTL Component Info :

+---Adders :

2 Input	32 Bit	Adders := 1
---------	--------	-------------

+---Registers :

32 Bit	Registers := 1
--------	----------------

1 Bit	Registers := 2
-------	----------------

+---Muxes :

2 Input	32 Bit	Muxes := 1
---------	--------	------------

2 Input	1 Bit	Muxes := 1
---------	-------	------------

Module D_FF

Detailed RTL Component Info :

+---Registers :

1 Bit	Registers := 1
-------	----------------

Module ProgramRom

Detailed RTL Component Info :

+---Muxes :

13 Input 14 Bit Muxes := 1

Module Fourbit_Adder_Substrator

Detailed RTL Component Info :

+---XORs :

2 Input 1 Bit XORs := 5

Module LUT_Display

Detailed RTL Component Info :

+---Muxes :

9 Input 7 Bit Muxes := 1

Finished RTL Hierarchical Component Statistics

● Power Consumption Report

| Tool Version : Vivado v.2018.1 (win64) Build 2188600 Wed Apr 4 18:40:38 MDT 2018

| Date : Sun May 5 12:32:27 2024

| Host : Maduni running 64-bit major release (build 9200)

| Command : report_power -file NanoProcessor_power_routed.rpt -pb

NanoProcessor_power_summary_routed.pb -rpx NanoProcessor_power_routed.rpx

| Design : NanoProcessor

| Device : xc7a35tcpg236-1

| Design State : routed

| Grade : commercial

| Process : typical

| Characterization : Production

Power Report

Table of Contents

1. Summary

1.1 On-Chip Components

1.2 Power Supply Summary

1.3 Confidence Level

2. Settings

2.1 Environment

2.2 Clock Constraints

3. Detailed Reports

3.1 By Hierarchy

1. Summary

Total On-Chip Power (W)	0.075
Design Power Budget (W)	Unspecified*
Power Budget Margin (W)	NA
Dynamic (W)	0.003
Device Static (W)	0.072
Effective TJA (C/W)	5.0
Max Ambient (C)	84.6
Junction Temperature (C)	25.4
Confidence Level	Low
Setting File	----
Simulation Activity File	----
Design Nets Matched	NA

* Specify Design Power Budget using, `set_operating_conditions -design_power_budget <value>` in Watts>

1.1 On-Chip Components

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	<0.001	3	---	---
Slice Logic	<0.001	168	---	---
LUT as Logic	<0.001	74	20800	0.36
CARRY4	<0.001	8	8150	0.10
Register	<0.001	56	41600	0.13
Others	0.000	8	---	---
Signals	<0.001	155	---	---
I/O	0.002	22	106	20.75
Static Power	0.072			
Total	0.075			

1.2 Power Supply Summary

Source	Voltage (V)	Total (A)	Dynamic (A)	Static (A)
Vccint	1.000	0.011	0.001	0.010
Vccaux	1.800	0.013	0.000	0.013
Vcco33	3.300	0.002	0.001	0.001
Vcco25	2.500	0.000	0.000	0.000
Vcco18	1.800	0.000	0.000	0.000
Vcco15	1.500	0.000	0.000	0.000
Vcco135	1.350	0.000	0.000	0.000
Vcco12	1.200	0.000	0.000	0.000
Vccaux_io	1.800	0.000	0.000	0.000
Vccbram	1.000	0.000	0.000	0.000
MGTAVcc	1.000	0.000	0.000	0.000
MGTAVtt	1.200	0.000	0.000	0.000
Vccadc	1.800	0.020	0.000	0.020

1.3 Confidence Level

User Input Data	Confidence	Details	Action
Design implementation state Clock nodes activity	High Low	Design is routed User specified less than 75% of clocks	Provide missing clock activity with a constraint file, simulation results or by editing the "By Clock Domain" view
I/O nodes activity	Medium	More than 5% of inputs are missing user specification	Provide missing input activity with simulation results or by editing the "By Resource Type -> I/Os" view
Internal nodes activity	Medium	User specified less than 25% of internal nodes	Provide missing internal nodes activity with simulation results or by editing the "By Resource Type" views
Device models	High	Device models are Production	
Overall confidence level	Low		

2. Settings

2.1 Environment

Ambient Temp (C)	25.0
ThetaJA (C/W)	5.0
Airflow (LFM)	250
Heat Sink	medium (Medium Profile)
ThetaSA (C/W)	4.6
Board Selection	medium (10"x10")
# of Board Layers	12to15 (12 to 15 Layers)
Board Temperature (C)	25.0

2.2 Clock Constraints

Clock	Domain	Constraint (ns)
sys_clk_pin	Clk	10.0

3. Detailed Reports

3.1 By Hierarchy

Name	Power (W)
NanoProcessor	0.003
Clock_0	<0.001
ProgramCounter_0	<0.001
D_FF_0	<0.001
D_FF_1	<0.001
D_FF_2	<0.001
D_FF_3	<0.001

RegisterBank	0.000
Reg_2	0.000
D_FF_0	0.000
D_FF_1	0.000
D_FF_2	0.000
D_FF_3	0.000
Reg_3	0.000
D_FF_0	0.000
D_FF_1	0.000
D_FF_2	0.000
D_FF_3	0.000
Reg_6	0.000
D_FF_0	0.000
D_FF_1	0.000
D_FF_2	0.000
D_FF_3	0.000
Reg_7	0.000
D_FF_0	0.000
D_FF_1	0.000
D_FF_2	0.000
D_FF_3	0.000

Conclusion

This lab gave us hands on experience on using hardware descriptive languages for programming FPGA boards and testing the designs via simulations and FPGA boards. And in this project, we faced many problems and solving them gave us invaluable experience. Also creating these

entities made us more informed about the entities used in this design. This project not only provided hands-on experience in digital design but also broaden our knowledge of the complexities involved in creating efficient and reliable processor architectures. By overcoming various challenges throughout the design and implementation phases, we gained valuable lessons into troubleshooting and optimizing digital systems.

Contribution of each team member

220322K – Kodisinghe R.R.

- Created the VHDL file for the instruction decoder with 12 bits.
- Created the VHDL file for the modified version of the instruction decoder with 14 bits.
- Combined all the components in the nano processor for the both versions.

220440U - Panagoda P.L.A.S.

- Created the VHDL file for the adder-subtractor
- Created the VHDL file for the 3-bit adder.
- Created the VHDL file for the
- multiplier in the modified version.

220309D – Karunaratne T.K.M.C.

- Created the VHDL files for 8 way 4-bit multiplexer, 2 way 4-bit multiplexer and 2 way 3-bit multiplexer
- Created the VHDL file for the seven-segment display.
- Created the VHDL file for the Boolean operator in the modified version.
- Created the VHDL file for the 4 way 4-bit multiplexer in the modified version.
- Created the VHDL file for the comparator in the modified version.

220469P – Perera M.R.S.

- Created the VHDL file for the register bank
- Created the VHDL file for the Program ROM
- Created the VHDL file for the 4-bit adder in the modified version.
- Created the VHDL file for the 4-bit program counter in the modified version.

