

Spring 2021
California State University, Northridge
Department of Electrical & Computer Engineering



Experiment 6B
Sum of Products
April 3, 2021
ECE 526L

Written By: Jose Luis Martinez

Introduction

In this lab we are tasked with creating a Sum of Products with verilog. The inputs of the circuit are to be scalable meaning that we should be able to just change a parameter and have it work for that input size. Our default size for the inputs should be 4 bits wide. The lowest level of the design should just be 3 modules, which are scalable multiplier, register, and adder. At the second level we will have two registers, two multipliers and one adder. For the second level we will have two instantiations of the first level and an adder. **Fig. 1** shows the Sum of Products circuit in diagram form.

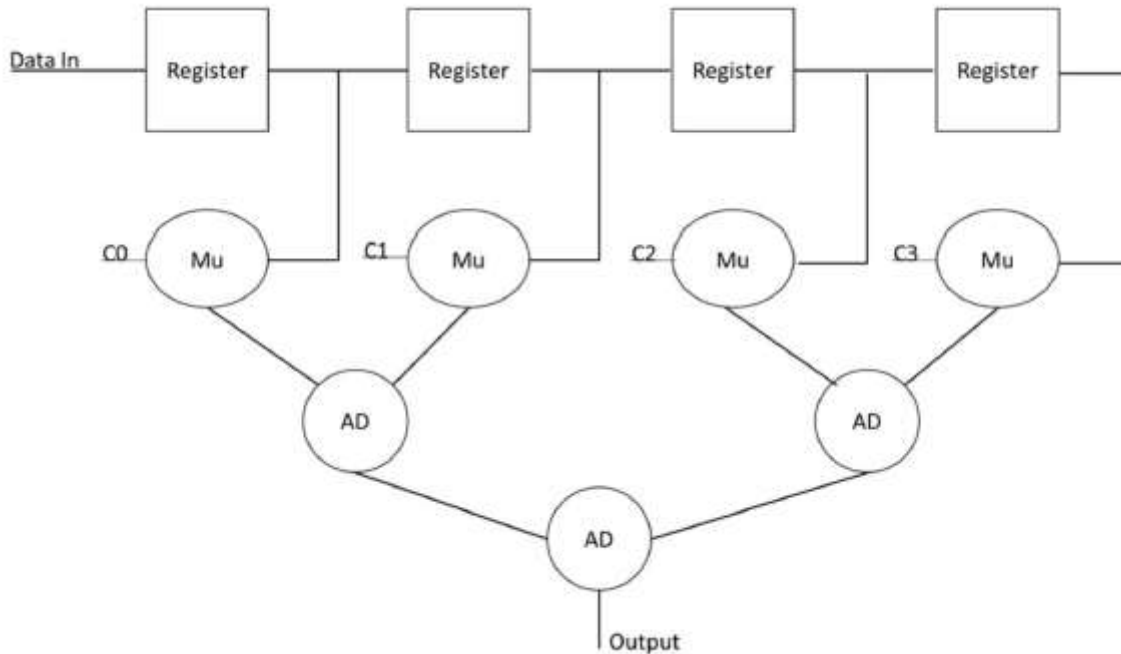


Fig. 1 Sum of Products Diagram

Methodology

I followed these steps to complete the lab:

1. The three lowest level modules were created first.
 - a. Scalable register was made, shown in **Fig. 2**.
 - b. Scalable multiplier was made, shown in **Fig. 3**.
 - c. Scalable adder was made, shown in **Fig. 4**.
2. The second level module was created.
 - a. Second level module was made, shown in **Fig. 5**.
 - i. Was made using 2 scalable registers, 2 scalable multipliers, and 1 scalable adder.
 - ii. The input D_in is connected to the input of the first register.
 - iii. The C0 and C1 inputs are connected to one multiplier each.
 - iv. The output is the result of the adder.
3. The top level module was created.
 - a. Top level module was made, as shown in **Fig. 6**.
 - i. Was made using 2 second level modules and one adder adding the outputs of each second level module.

4. The testbench file was created to test and ensure our top level module was working properly.
 - a. A non exhaustive test bench was created as shown in **Fig. 7**.
 - i. A small test vector of about a dozen elements was tested.
 - b. An exhaustive test bench was created as shown in **Fig. 8**.
 - i. All possible values of D_IN were tested and the answer was compared to a non-hierarchical output in order to ensure that my design is working properly.
 - ii. An `ifdef compiler directive was used to force an error.

Results/Verilog Files

```
1  /*****
2  ***
3  *** ECE 526 L Experiment #6B          Jose Luis Martinez, Spring, 2021 ***
4  ***
5  *** Experiment 6B - Sum of Products ***
6  ***
7  *****/
8  *** Filename: sca_reg.v          Created by Jose Luis Martinez, April 1, 2021 ***
9  ***
10 *****/
11
12 `timescale 1ns/100ps
13
14 module sca_reg(OUT, A, RST, CLK);
15     parameter SIZE = 4;
16     input CLK, RST;
17     input [SIZE-1:0] A;
18     output reg [SIZE-1:0] OUT;
19
20     always@(posedge CLK or negedge RST) begin
21         if(!RST) begin
22             OUT <= 0;
23         end else begin
24             OUT <= A;
25         end
26     end
27 endmodule
```

Fig. 2 sca_reg.v

```

1  /*****
2  ***                                     ***
3  *** ECE 526 L Experiment #6B           Jose Luis Martinez, Spring, 2021 ***
4  ***                                     ***
5  *** Experiment 6B - Sum of Products ***
6  ***                                     ***
7  *****/
8  *** Filename: sca_mult.v             Created by Jose Luis Martinez, April 1, 2021 ***
9  ***                                     ***
10 *****/
11
12 `timescale 1ns/100ps
13
14 module sca_mult(OUT, A, B);
15     parameter SIZE = 4;
16     input [SIZE-1:0] A;
17     input [SIZE-1:0] B;
18     output [(SIZE*2)-1:0] OUT;
19
20     assign OUT = A * B;
21 endmodule

```

Fig. 3 sca_mult.v

```

1  /*****
2  ***                                     ***
3  *** ECE 526 L Experiment #6B           Jose Luis Martinez, Spring, 2021 ***
4  ***                                     ***
5  *** Experiment 6B - Sum of Products ***
6  ***                                     ***
7  *****/
8  *** Filename: sca_add.v             Created by Jose Luis Martinez, April 1, 2021 ***
9  ***                                     ***
10 *****/
11
12 `timescale 1ns/100ps
13
14 module sca_add(OUT, A, B);
15     parameter SIZE = 8;
16     input [SIZE-1:0] A;
17     input [SIZE-1:0] B;
18     output [SIZE:0] OUT;
19
20     assign OUT = A + B;
21 endmodule

```

Fig. 4 sca_add.v

```

1  /**
2  ***
3  *** ECE 526 L Experiment #6B          Jose Luis Martinez, Spring, 2021 ***
4  ***
5  *** Experiment 6B - Sum of Products ***
6  ***
7  ****
8  *** Filename: lvl2.v          Created by Jose Luis Martinez, April 1, 2021 ***
9  ***
10 ****/
11
12 `timescale 1ns/100ps
13
14 module lvl2(SUM_OUT, REG_OUT, A, B, C, CLK, RST);
15     parameter SIZE = 4;
16
17     input CLK, RST;
18     input [SIZE-1:0] A;
19     input [SIZE-1:0] B;
20     input [SIZE-1:0] C;
21     output [SIZE-1:0] REG_OUT;
22     output [SIZE*2:0] SUM_OUT;
23
24     wire [SIZE-1:0] r1r2;
25     wire [(SIZE*2)-1:0] prod1;
26     wire [(SIZE*2)-1:0] prod2;
27
28     sca_reg #(.SIZE(SIZE)) sr1(.OUT(r1r2), .A(A), .RST(RST), .CLK(CLK));
29     sca_reg #(.SIZE(SIZE)) sr2(.OUT(REG_OUT), .A(r1r2), .RST(RST), .CLK(CLK));
30
31     sca_mult #(.SIZE(SIZE)) mult1(.OUT(prod1), .A(r1r2), .B(B));
32     sca_mult #(.SIZE(SIZE)) mult2(.OUT(prod2), .A(REG_OUT), .B(C));
33
34     sca_add #(.SIZE(SIZE*2)) add1(.OUT(SUM_OUT), .A(prod1), .B(prod2));
35 endmodule

```

Fig. 5 lvl2.v


```

1  /**
2  ***
3  *** ECE 526 L Experiment #6B Jose Luis Martinez, Spring, 2021 ***
4  ***
5  *** Experiment 6B - Sum of Products ***
6  ***
7  ****
8  *** Filename: top_lv1.v Created by Jose Luis Martinez, April 1, 2021 ***
9  ***
10 ****/
11
12 `timescale 1ns/100ps
13
14 module top_lv1(OUT, D_IN, C0, C1, C2, C3, RST, CLK);
15     parameter SIZE = 4;
16
17     input RST, CLK;
18     input [SIZE-1:0] D_IN;
19     input [SIZE-1:0] C0;
20     input [SIZE-1:0] C1;
21     input [SIZE-1:0] C2;
22     input [SIZE-1:0] C3;
23     output [(SIZE*2)+1:0] OUT;
24
25     wire [SIZE*2:0] sum1;
26     wire [SIZE*2:0] sum2;
27     wire [SIZE-1:0] rout1;
28     wire [SIZE-1:0] rout2;
29
30     lv12_1(.SIZE(SIZE)) lv12_1(.SUM_OUT(sum1), .REG_OUT(rout1),
31     .A(D_IN), .B(C0), .C(C1), .CLK(CLK), .RST(RST));
32     lv12_2(.SIZE(SIZE)) lv12_2(.SUM_OUT(sum2), .REG_OUT(rout2),
33     .A(rout1), .B(C2), .C(C3), .CLK(CLK), .RST(RST));
34
35     sca_add #(.SIZE((SIZE*2)+1)) add1(.OUT(OUT), .A(sum1), .B(sum2));
36
37 endmodule

```

Fig. 6 top_lv1.v

```

1  /*****
2  ***                                     ***
3  *** ECE 526 L Experiment #6B           Jose Luis Martinez, Spring, 2021 ***
4  ***                                     ***
5  *** Experiment 6B - Sum of Products    ***
6  ***                                     ***
7  *****/
8  *** Filename: lab_6b_nonex_tb.vCreated by Jose Luis Martinez, April 1, 2021 ***
9  ***                                     ***
10 *****/
11
12 `timescale 1ns/100ps
13
14 `define CLK_PER 20
15 `define SDISPLAY "%d, D_IN = %h, C0 = %h, C1 = %h, C2 = %h, C3 = %h, RST = %b, CLK = %b| C
16
17 module lab_6b_nonex_tb();
18     parameter SIZE = 4;
19
20     reg [SIZE-1:0] D_IN;
21     reg [SIZE-1:0] C0;
22     reg [SIZE-1:0] C1;
23     reg [SIZE-1:0] C2;
24     reg [SIZE-1:0] C3;
25     reg RST, CLK;
26     wire [(SIZE*2)+1:0] OUT;
27
28     top_lvl #(.SIZE(SIZE)) tp1(.OUT(OUT), .D_IN(D_IN), .C0(C0), .C1(C1), .C2(C2), .C3(C3),
29
30     initial begin
31         CLK <= 0;
32         forever begin
33             #(`CLK_PER/2) CLK <= ~CLK;
34         end
35     end
36
37     initial begin
38         $vcdpluson;
39         D_IN <= 4'h4;
40         RST <= 1'b1;
41         C0 <= 4'h2;
42         C1 <= 4'h3;
43         C2 <= 4'h4;
44         C3 <= 4'h9;
45         $display(`SDISPLAY, $time, D_IN, C0, C1, C2, C3, RST, CLK, OUT);
46
47         #(`CLK_PER)
48         D_IN <= 4'h0;

```

```

49         $display(`SDISPLAY, $time, D_IN, C0, C1, C2, C3, RST, CLK, OUT);
50
51         #(`CLK_PER)
52         D_IN <= 4'h1;
53         $display(`SDISPLAY, $time, D_IN, C0, C1, C2, C3, RST, CLK, OUT);
54
55         #(`CLK_PER)
56         D_IN <= 4'hA;
57         $display(`SDISPLAY, $time, D_IN, C0, C1, C2, C3, RST, CLK, OUT);
58
59         #(`CLK_PER)
60         D_IN <= 4'hB;
61         $display(`SDISPLAY, $time, D_IN, C0, C1, C2, C3, RST, CLK, OUT);
62
63         #(`CLK_PER)

```

```

64     D_IN <= 4'h2;
65     $display(`SDISPLAY, $time, D_IN, C0, C1, C2, C3, RST, CLK, OUT);
66
67     #(`CLK_PER)
68     D_IN <= 4'hF;
69     $display(`SDISPLAY, $time, D_IN, C0, C1, C2, C3, RST, CLK, OUT);
70
71     #(`CLK_PER)
72     D_IN <= 4'h8;
73     $display(`SDISPLAY, $time, D_IN, C0, C1, C2, C3, RST, CLK, OUT);
74
75     #(`CLK_PER)
76     D_IN <= 4'h0;
77     $display(`SDISPLAY, $time, D_IN, C0, C1, C2, C3, RST, CLK, OUT);
78
79     #(`CLK_PER)
80     D_IN <= 4'hD;
81     $display(`SDISPLAY, $time, D_IN, C0, C1, C2, C3, RST, CLK, OUT);
82
83     #(`CLK_PER)
84     $display(`SDISPLAY, $time, D_IN, C0, C1, C2, C3, RST, CLK, OUT);
85     $finish;
86 end
87
88 endmodule

```

Fig. 7 lab_6b_nonex_tb.v


```

1  /*******
2  ***
3  *** ECE 526 L Experiment #6B           Jose Luis Martinez, Spring, 2021 ***
4  ***
5  *** Experiment 6B - Sum of Products ***
6  ***
7  *****/
8  *** Filename: lab_6b_ex_tb.v   Created by Jose Luis Martinez, April 1, 2021 ***
9  ***
10 *****/
11
12 `timescale 1ns/100ps
13
14 `define CLK_PER 20
15 `define SDISPLAY "%d, D_IN = %d, C0 = %d, C1 = %d, C2 = %d, C3 = %d, RST = %b, CLK = %b| C
16
17 module lab_6b_ex_tb();
18     parameter SIZE = 4;
19
20     reg [SIZE-1:0] D_IN;
21     reg [SIZE-1:0] C0;
22     reg [SIZE-1:0] C1;
23     reg [SIZE-1:0] C2;
24     reg [SIZE-1:0] C3;
25     reg [(SIZE*2)+1:0] E_OUT;
26     reg [3:0] D_ARR [SIZE-1:0];
27     reg RST, CLK;
28     wire [(SIZE*2)+1:0] OUT;
29     integer i;
30
31     top_lvl #(.SIZE(SIZE)) tp1(.OUT(OUT), .D_IN(D_IN), .C0(C0), .C1(C1), .C2(C2), .C3(C3),
32
33     initial begin
34         CLK <= 0;
35         forever begin
36             #(`CLK_PER/2) CLK <= ~CLK;
37         end
38     end
39
40     initial $monitorb(`SDISPLAY, $time, D_IN, C0, C1, C2, C3, RST, CLK, OUT, E_OUT);
41
42     initial begin
43         $vcdpluson;
44         RST <= 1'b1;
45         D_IN <= 4'h0;
46         D_ARR[0] <= 4'h0;
47         C0 <= 4'h2;
48         C1 <= 4'h3;

```

```

49         C2 <= 4'h4;
50         C3 <= 4'h9;
51
52         #(`CLK_PER*6)
53         $display("\t\t\tFirst set of test vectors.\n\n");
54         $monitoroff;
55         #(`CLK_PER*4)
56         $monitoron;
57         #(`CLK_PER*6)
58         $display("\t\t\tLast set of test vectors.\n\n");
59         $finish;
60     end

```

```

61
62     always@(posedge CLK) begin
63         D_IN <= D_IN + 1'b1;
64
65         E_OUT <= C0*D_ARR[0] + C1*D_ARR[1] + C2*D_ARR[2] + C3*D_ARR[3];
66
67         #(1)
68         D_ARR[3] <= D_ARR[2];
69         D_ARR[2] <= D_ARR[1];
70         D_ARR[1] <= D_ARR[0];
71         D_ARR[0] <= D_IN;
72
73         if(E_OUT != OUT) begin
74             $display("\t\t\tError! Outputs do not match!\n");
75             $finish;
76         end
77     end
78
79     `ifdef ERROR
80     initial begin
81         #(`CLK_PER*9)
82         $monitoron;
83         #(`CLK_PER*6) force OUT = 4'b1x0x;
84     end
85     `endif
86
87 endmodule
88

```

Fig. 8 lab_6b_ex_tb.v

```
dcd151.ecs.csun.edu
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

Quick connect...
3 dcd151.ecs.csun.edu

$ simv -l Lab6b_ex.log
Chronologic VCS simulator copyright 1991-2017
Contains Synopsys proprietary information.
Compiler version N-2017.12-SP2-2_Full64; Runtime version N-2017.12-SP2-2_Full64; Apr 3 19:32 2021
VCD+ Writer N-2017.12-SP2-2_Full64 Copyright (c) 1991-2017 by Synopsys Inc.

0, D_IN = 0, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = x Expected OUT = x
10, D_IN = 1, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = x Expected OUT = x
20, D_IN = 1, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = x Expected OUT = x
30, D_IN = 2, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = x Expected OUT = x
40, D_IN = 2, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = x Expected OUT = x
50, D_IN = 3, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = x Expected OUT = x
60, D_IN = 3, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = x Expected OUT = x
70, D_IN = 4, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = 16 Expected OUT = 16
80, D_IN = 4, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = 16 Expected OUT = 16
90, D_IN = 5, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = 34 Expected OUT = 34
100, D_IN = 5, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = 34 Expected OUT = 34
110, D_IN = 6, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = 52 Expected OUT = 52
First set of test vectors.

200, D_IN = 10, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = 124 Expected OUT = 124
210, D_IN = 11, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = 142 Expected OUT = 142
220, D_IN = 11, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = 142 Expected OUT = 142
230, D_IN = 12, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = 160 Expected OUT = 160
240, D_IN = 12, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = 160 Expected OUT = 160
250, D_IN = 13, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = 178 Expected OUT = 178
260, D_IN = 13, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = 178 Expected OUT = 178
270, D_IN = 14, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = 196 Expected OUT = 196
280, D_IN = 14, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = 196 Expected OUT = 196
290, D_IN = 15, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = 214 Expected OUT = 214
300, D_IN = 15, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = 214 Expected OUT = 214
310, D_IN = 0, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = 232 Expected OUT = 232
Last set of test vectors.

$finish called from file "lab_6b_ex_tb.v", line 59.
$finish at simulation time 3200
VCS Simulation Report
Time: 320000 ps
CPU Time: 0.200 seconds; Data structure size: 0.0Mb
Sat Apr 3 19:32:37 2021
$
```

Fig. 9 exhaustive simv output

```
dcd151.ecs.csun.edu
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

Quick connect...
3 dcd151.ecs.csun.edu

Compiler version N-2017.12-SP2-2_Full64; Runtime version N-2017.12-SP2-2_Full64; Apr 3 19:39 2021
VCD+ Writer N-2017.12-SP2-2_Full64 Copyright (c) 1991-2017 by Synopsys Inc.

0, D_IN = 0, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = x Expected OUT = x
10, D_IN = 1, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = x Expected OUT = x
20, D_IN = 1, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = x Expected OUT = x
30, D_IN = 2, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = x Expected OUT = x
40, D_IN = 2, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = x Expected OUT = x
50, D_IN = 3, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = x Expected OUT = x
60, D_IN = 3, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = x Expected OUT = x
70, D_IN = 4, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = 16 Expected OUT = 16
80, D_IN = 4, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = 16 Expected OUT = 16
90, D_IN = 5, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = 34 Expected OUT = 34
100, D_IN = 5, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = 34 Expected OUT = 34
110, D_IN = 6, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = 52 Expected OUT = 52
First set of test vectors.

180, D_IN = 9, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = 106 Expected OUT = 106
190, D_IN = 10, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = 124 Expected OUT = 124
200, D_IN = 10, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = 124 Expected OUT = 124
210, D_IN = 11, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = 142 Expected OUT = 142
220, D_IN = 11, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = 142 Expected OUT = 142
230, D_IN = 12, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = 160 Expected OUT = 160
240, D_IN = 12, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = 160 Expected OUT = 160
250, D_IN = 13, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = 178 Expected OUT = 178
260, D_IN = 13, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = 178 Expected OUT = 178
270, D_IN = 14, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = 196 Expected OUT = 196
280, D_IN = 14, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = 196 Expected OUT = 196
290, D_IN = 15, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = 214 Expected OUT = 214
300, D_IN = 15, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 0 OUT = 214 Expected OUT = 214
310, D_IN = 0, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1 OUT = 232 Expected OUT = 232
Error! Outputs do not match!

$finish called from file "lab_6b_ex_tb.v", line 75.
$finish at simulation time 3110
VCS Simulation Report
Time: 311000 ps
CPU Time: 0.200 seconds; Data structure size: 0.0Mb
Sat Apr 3 19:39:13 2021
$ dve -full64 &
[1] 26816
$
```

Fig. 10 exhaustive with error simv output


```

Parsing design file 'lvl2.v'
Parsing design file 'top_lvl.v'
Parsing design file 'lab_6b_nonex_tb.v'
Top Level Modules:
lab_6b_nonex_tb
TimeScale is 1 ns / 100 ps
Starting vcs inline pass...
1 module and 0 UDP read.
However, due to incremental compilation, no re-compilation is necessary.
rm -f _csrc*.so pre_vcsobj*.so share_vcsobj*.so
if [ -x ../simv ]; then chmod -x ../simv; fi
g++ -o ../simv -Wl,-rpath-link=/ -Wl,-rpath=$ORIGIN/simv.daidir/ -Wl,-rpath=/simv.daidir/ -Wl,-rpath=$ORIGIN/simv.daidir
//scsim.db.dir -dynamic -Wl,-rpath=/opt/synopsys/vcs-mx/N-2017.12-SP2-2/linux64/lib -L/opt/synopsys/vcs-mx/N-2017.12-SP2-2/linux
x64/lib -lcmag -obj/armcqw.d.o _22469_archive_1.so prev_archive_1.so SIM_1.o rmapats_mop.o rmapats.o rmar.o rmar_nd.o
rmar_llvm.o rmar_llvm_0.o -lzerosoft -lstub5 -lvirsim -lerrorinf -lspmsmalloc -lvifs -lvcsnew -lsmprofile -lucl
native /opt/synopsys/vcs-mx/N-2017.12-SP2-2/linux64/lib/vcs_tls.o -Wl,-whole-archive -lvcsucli -Wl,-no-whole-archive
/opt/synopsys/vcs-mx/N-2017.12-SP2-2/linux64/lib/vcs_save_restore_new.o -ldl -lc -lm -lpthread -ldl
../simv up to date
CPU time: .302 seconds to compile + .422 seconds to elab + .284 seconds to link
$ simv
Chronologic VCS simulator copyright 1991-2017
Contains Synopsys proprietary information.
Compiler version N-2017.12-SP2-2_Full64; Runtime version N-2017.12-SP2-2_Full64; Apr 3 19:18 2021
VCD+ Writer N-2017.12-SP2-2_Full64 Copyright (c) 1991-2017 by Synopsys Inc.
0, D_IN = x, C0 = x, C1 = x, C2 = x, C3 = x, RST = x, CLK = x| OUT = x
20, D_IN = 4, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1| OUT = x
40, D_IN = 0, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1| OUT = x
60, D_IN = 1, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1| OUT = x
80, D_IN = a, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1| OUT = 59
100, D_IN = b, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1| OUT = 56
120, D_IN = 2, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1| OUT = 86
140, D_IN = f, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1| OUT = 170
160, D_IN = 8, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1| OUT = 168
180, D_IN = 0, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1| OUT = 102
200, D_IN = d, C0 = 2, C1 = 3, C2 = 4, C3 = 9, RST = 1, CLK = 1| OUT = 193
$finish called from file "lab_6b_nonex_tb.v", line 85.
$finish at simulation time 2000
VCS Simulation Report
Time: 200000 ps
CPU Time: 0.200 seconds; Data structure size: 0.0Mb
Sat Apr 3 19:18:47 2021
$

```

Fig. 11 non-exhaustive simv output

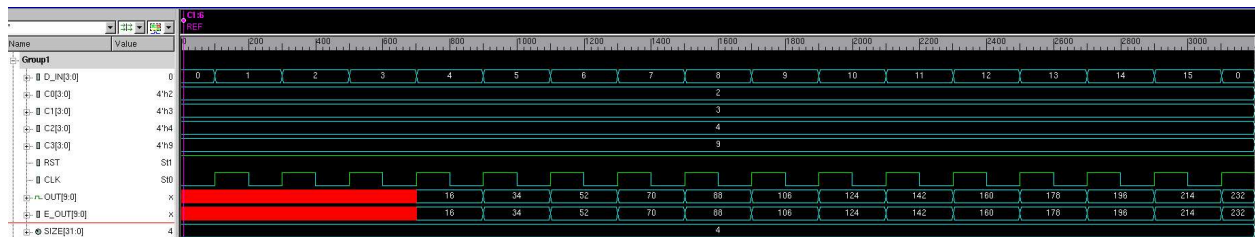


Fig. 12 exhaustive waveform

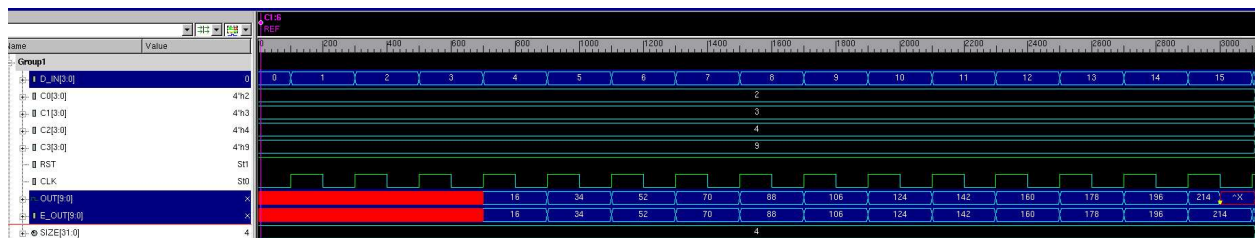


Fig. 13 exhaustive with error waveform

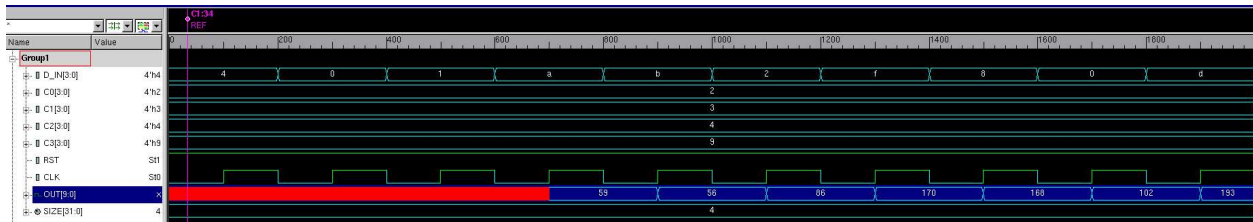


Fig. 14 non-exhaustive waveform

Analysis

For our non-exhaustive test we can see our results in **Fig. 11** and **Fig. 14**. Our OUTPUT is unknown for the first four clock cycles because our registers have not been initialized yet. So at the fourth cycle all registers will have a value and our OUT will not be X. This also applies to the other testbench file.

To verify that its working properly:

1. $OUT = 4 * C3 + 0 * C2 + 1 * C1 + 10 * C0 = 4 * 9 + 0 * 4 + 1 * 3 + 10 * 2 = 59$
2. $OUT = B * C3 + 2 * C2 + F * C1 + 8 * C0 = B * 9 + 2 * 4 + F * 3 + 8 * 2 = 168$

These both match the waveform and simv outputs so we are able to verify that our design is working as designed.

For our exhaustive test, I tested all possible values of D_IN and we can see the results of this testbench in **Fig. 9, 10, 12, and 13**. In this testbench we also included a ``ifdef` compiler directive that will only compile that portion of the code if we specify the flag during compilation. For this testbench the ``ifdef` compiler directive forces our OUT to 4'b1x0x after 15 clock periods. In **Fig. 10 & 13** we compiled with defining ERROR so we were able to force our error. And as we can see in our simv output when our OUT is different from the EX_OUT our program prompts that there was an error and will terminate.

Conclusion

In conclusion, I created a hierarchical design for a Sum of Products digital filter. The design consisted of 3 modules on the lowest level being scalable register, multiplier, and adder. Two multipliers, two registers, and one adder from the lowest level were used to create the second level module. The top level module used two second level modules and one adder to complete the design. Two testbench files were created to test our design and as we can see from my results we are able to verify that the design is working as expected. For this lab I was able to complete all the requirements and have learned about the ``ifdef` compiler directive and I see myself using it more in the future.

Academic Dishonesty

Submitting any report that is not entirely your own work is a form of academic dishonesty and will not be tolerated. Each and every lab report must include the following statement, signed and dated by the student. Lab reports without the statement will be summarily rejected.

I hereby attest that this lab report is entirely my own work. I have not copied either code or text from anyone, nor have I allowed or will I allow anyone to copy my work.

Name (printed) Jose L Martinez

Name (signed) Jose Martinez Date 4/03/2021