

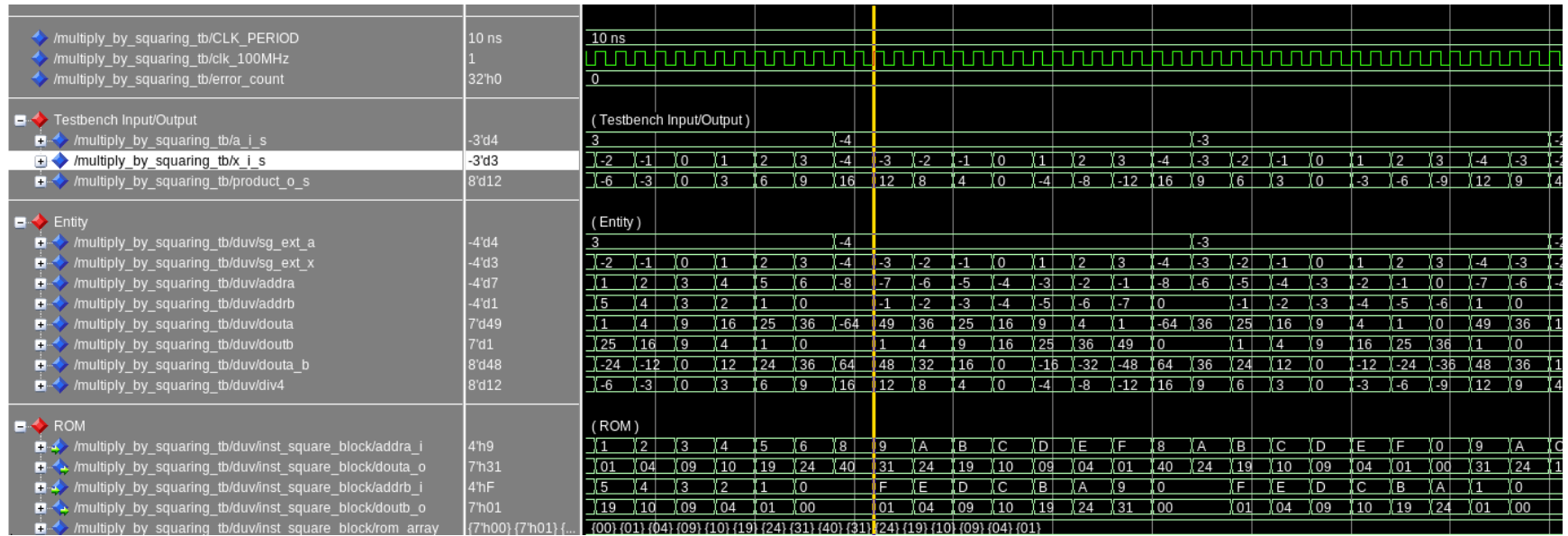
Test Case : When input $a = 001$ and $a = 010$

	Value	Hex	Dec
/multiply_by_squaring_tb/error_count	32'h0	0	0
Testbench Input/Output	(Testbench Input/Output)		
/multiply_by_squaring_tb/a_i_s	3	1	2
/multiply_by_squaring_tb/x_i_s	3	1 2 3 -4 -3 -2 -1 0 1 2 3 -4 -3 -2 -1 0 1 2	
/multiply_by_squaring_tb/product_o_s	9	1 2 3 -4 -3 -2 -1 0 1 4 6 -8 -6 -4 -2 0 2 4	
Entity	(Entity)		
/multiply_by_squaring_tb/duv/sg_ext_a	4'd3	1	2
/multiply_by_squaring_tb/duv/sg_ext_x	4'd3	1 2 3 -4 -3 -2 -1 0 1 2 3 -4 -3 -2 -1 0 1 2	
/multiply_by_squaring_tb/duv/addra	4'd6	2 3 4 -3 -2 -1 0 1 2 4 5 -2 -1 0 1 2 3 4	
/multiply_by_squaring_tb/duv/addrb	4'd0	0 -1 -2 5 4 3 2 1 0 -1 6 5 4 3 2 1 0	
/multiply_by_squaring_tb/duv/douta	7'd36	4 9 16 9 4 1 0 1 4 16 25 4 1 0 1 4 9 16	
/multiply_by_squaring_tb/duv/doutb	7'd0	0 1 4 25 16 9 4 1 0 1 36 25 16 9 4 1 0	
/multiply_by_squaring_tb/duv/douta_b	8'd36	4 8 12 -16 -12 -8 -4 0 4 16 24 -32 -24 -16 -8 0 8 16	
/multiply_by_squaring_tb/duv/div4	8'd9	1 2 3 -4 -3 -2 -1 0 1 4 6 -8 -6 -4 -2 0 2 4	
ROM	(ROM)		
/multiply_by_squaring_tb/duv/inst_square_block[0]	4'h6	2 3 4 D E F 0 1 2 4 5 E F 0 1 2 3 4	
/multiply_by_squaring_tb/duv/inst_square_block[1]	7'h24	04 09 10 09 04 01 00 01 04 10 19 04 01 00 01 04 09 10	
/multiply_by_squaring_tb/duv/inst_square_block[2]	4'h0	0 F E 5 4 3 2 1 0 F 6 5 4 3 2 1 0	
/multiply_by_squaring_tb/duv/inst_square_block[3]	7'h00	00 01 04 19 10 09 04 01 00 01 24 19 10 09 04 01 00	
/multiply_by_squaring_tb/duv/inst_square_block[4]	{7'h00} {7'h01} {...}	{00}{01}{04}{09}{10}{19}{24}{31}{40}{31}{24}{19}{10}{09}{04}{01}	
(0)	7'b0000000	0000000	
(1)	7'b0000001	0000001	
(2)	7'b0000100	0000100	
(3)	7'b0001001	0001001	
(4)	7'b0010000	0010000	
(5)	7'b0011001	0011001	
(6)	7'b0100100	0100100	
(7)	7'b0110001	0110001	
(8)	7'b1000000	1000000	
(9)	7'b0110001	0110001	
(10)	7'b0100100	0100100	
(11)	7'b0011001	0011001	
(12)	7'b0010000	0010000	
(13)	7'b0001001	0001001	
(14)	7'b0000100	0000100	
(15)	7'b0000001	0000001	

Test Case : When input a = 111 and a = 000

/multiply_by_squaring_tb/error_count		32'h0	0
Testbench Input/Output			(Testbench Input/Output)
+ /multiply_by_squaring_tb/a_i_s	3		-2 -1 0 1 2 3 -4 -3 -2 -1 0 1 2 3 -4 -3 -2 -1 0 1
+ /multiply_by_squaring_tb/x_i_s	3		-2 -1 0 1 2 3 -4 -3 -2 -1 0 1 2 3 -4 -3 -2 -1 0 1
+ /multiply_by_squaring_tb/product_o_s	9		4 1 0 -1 -2 -3 4 3 2 1 0 1 2 3 -4 -3 -2 -1 0 1
Entity			(Entity)
+ /multiply_by_squaring_tb/duv/sg_ext_a	4'd3		-2 -1 0 1 2 3 -4 -3 -2 -1 0 1 2 3 -4 -3 -2 -1 0 1
+ /multiply_by_squaring_tb/duv/sg_ext_x	4'd3		-2 -1 0 1 2 3 -4 -3 -2 -1 0 1 2 3 -4 -3 -2 -1 0 1
+ /multiply_by_squaring_tb/duv/addra	4'd6		-4 -2 -1 0 1 2 -5 -4 -3 -2 0 1 2 3 -4 -3 -2 -1 0 2
+ /multiply_by_squaring_tb/duv/addrb	4'd0		0 -1 -2 -3 -4 3 2 1 0 -1 -2 -3 4 3 2 1 0
+ /multiply_by_squaring_tb/duv/douta	7'd36		16 4 1 0 1 4 25 16 9 4 0 1 4 9 16 9 4 1 0 4
+ /multiply_by_squaring_tb/duv/doutb	7'd0		0 1 4 9 16 9 4 1 0 1 4 9 16 9 4 1 0
+ /multiply_by_squaring_tb/duv/douta_b	8'd36		16 4 0 -4 -8 -12 16 12 8 4 0 1 4 9 16 9 4 1 0
+ /multiply_by_squaring_tb/duv/div4	8'd9		4 1 0 -1 -2 -3 4 3 2 1 0 1 2 3 -4 -3 -2 -1 0 1
ROM			(ROM)
+ /multiply_by_squaring_tb/duv/inst_square_blo...	4'h6		C E F 0 1 2 B C D E 0 1 2 3 C D E F 0 2
+ /multiply_by_squaring_tb/duv/inst_square_blo...	7'h24		10 04 01 00 01 04 19 10 09 04 00 01 04 09 10 09 04 01 00 0
+ /multiply_by_squaring_tb/duv/inst_square_blo...	4'h0		0 F E D C 3 2 1 0 F E D 4 3 2 1 0
+ /multiply_by_squaring_tb/duv/inst_square_blo...	7'h00		00 01 04 09 10 09 04 01 00 01 04 09 10 09 04 01 00
+ /multiply_by_squaring_tb/duv/inst_square_blo...	{7'h00} {7'h01} {...		{00} {01} {04} {09} {10} {19} {24} {31} {40} {31} {24} {19} {10} {09} {04} {01}
+ (0)	7'b0000000		0000000
+ (1)	7'b0000001		0000001
+ (2)	7'b0000100		0000100
+ (3)	7'b0001001		0001001
+ (4)	7'b0010000		0010000
+ (5)	7'b0011001		0011001
+ (6)	7'b0100100		0100100
+ (7)	7'b0110001		0110001
+ (8)	7'b1000000		1000000
+ (9)	7'b0110001		0110001
+ (10)	7'b0100100		0100100
+ (11)	7'b0011001		0011001
+ (12)	7'b0010000		0010000
+ (13)	7'b0001001		0001001
+ (14)	7'b0000100		0000100
+ (15)	7'b0000001		0000001

Test Case : when input a = 100 and 101



Console Output

```
Transcript
# ** Warning: NUMERIC_STD.TO_INTEGER: metavalue detected, returning 0
#   Time: 0 ns  Iteration: 1  Instance: /multiply_by_squaring_tb/duv/inst_square_block
# ** Warning: NUMERIC_STD.TO_INTEGER: metavalue detected, returning 0
#   Time: 0 ns  Iteration: 1  Instance: /multiply_by_squaring_tb/duv/inst_square_block
#
# *****
# RESULTS
# *****
#
# Simulation completed without errors
#
# *****
# Simulation Ending
# *****
#
# Break in Process stimuli_proc at ../tb/multiply_by_squaring_tb.vhd line 134
VSIM 2>
```

Coverage Result

```
vccover report -verbose -output final_report_verbose.txt merged_coverage.ucdb
End time: 22:49:44 on Nov 03,2022, Elapsed time: 0:00:00
Errors: 0, Warnings: 0
Reading pref.tcl

# 2020.4_2

# coverage read -dataset merged_coverage merged_coverage.ucdb
# merged_coverage.ucdb opened as coverage dataset "merged_coverage"
# coverage report -output final_report_by_srcfile.txt -srcfile=* -code {s b c e f x}
# exit
Coverage Report Summary Data by file

=====
=== File: ../src/dp_rom.vhd
=====


| Enabled Coverage | Bins | Hits | Misses | Coverage |
|------------------|------|------|--------|----------|
| Statements       | 2    | 2    | 0      | 100.00%  |


=====
=== File: ../src/multiply_by_squaring.vhd
=====


| Enabled Coverage | Bins | Hits | Misses | Coverage |
|------------------|------|------|--------|----------|
| Statements       | 7    | 7    | 0      | 100.00%  |


Total Coverage By File (code coverage only, filtered view): 100.00%
```
