COL215: Assignment 3 - Implementation of AES decryption operation

INSTRUCTOR NAME-PREETI RANJAN PANDA

	NAME	ENTRY NO
MEMBER 1	AYUSH SINGH	2023CS10322
MEMBER 2	AAYUSH PRASAD	2023CS51132

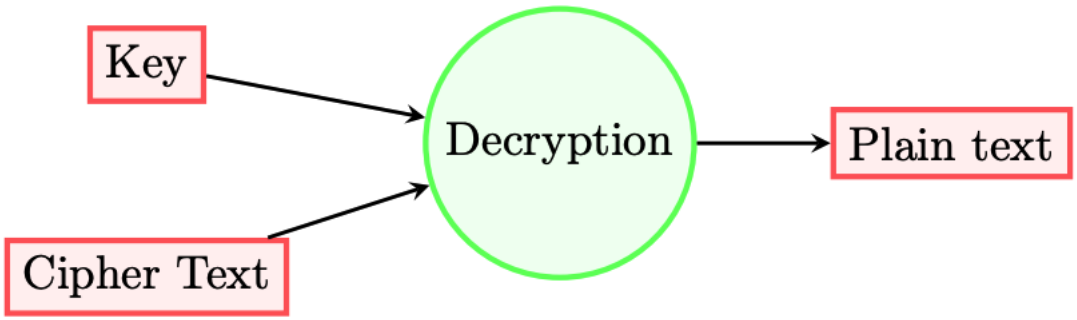
Introduction

The objective of the assignment is: implementation of AES decryption operation. The components involved are memory elements (RAM, ROM and registers) and logical unit.

Problem Description

Given a cipher text and a key, your task is to perform an AES decryption operation to display plain text on 7-seven display. An overview of the problem is shown in Figure 1.

Figure 1: Overview of task: Decryption



• Input cipher text is of the size in multiples of 128 bits and will be provided in the COE file (8 bit binary unsigned). The input file should be stored in 1-D array format in the block RAM, starting from address 0 (000016) in row major format.

• Key will be provided via the COE file. All the round keys will be provided in the COE file. You are allowed to use VHDL modules from previous hardware assignments.

In the Part I of the assignment, the objective will be to design and implement the following modules:

1. Memories (RAM/ROM)

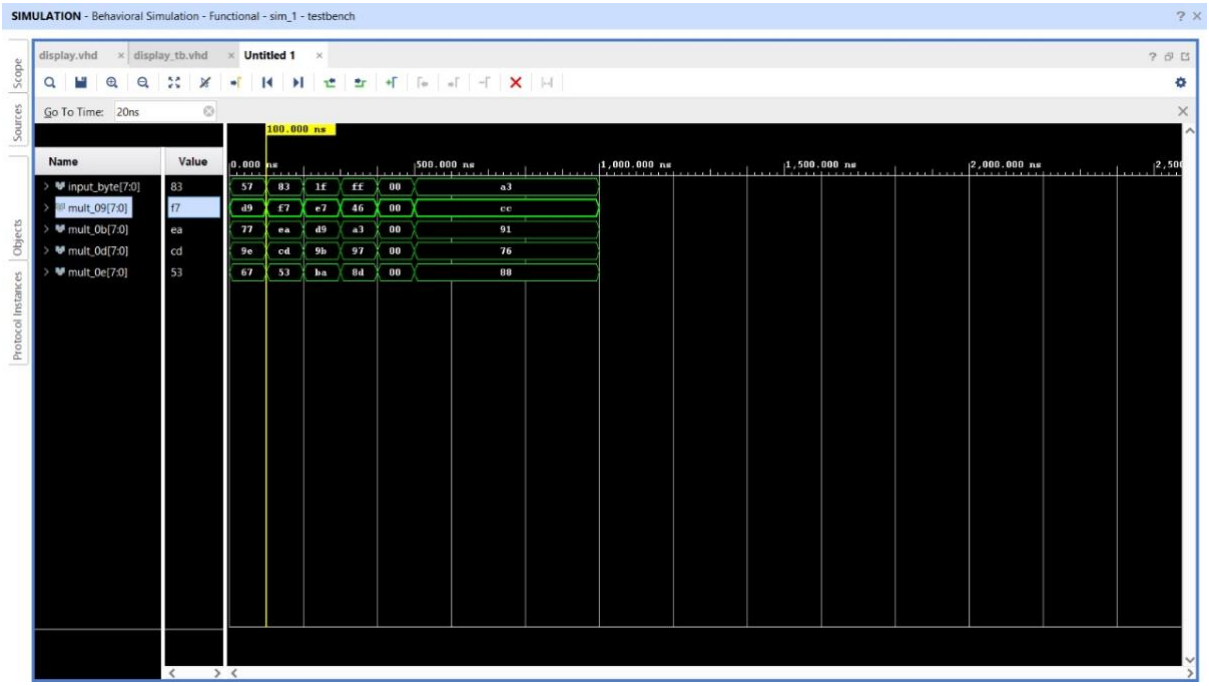
2. Logical operations in AES decryption Primary focus will be on designing the individual components.

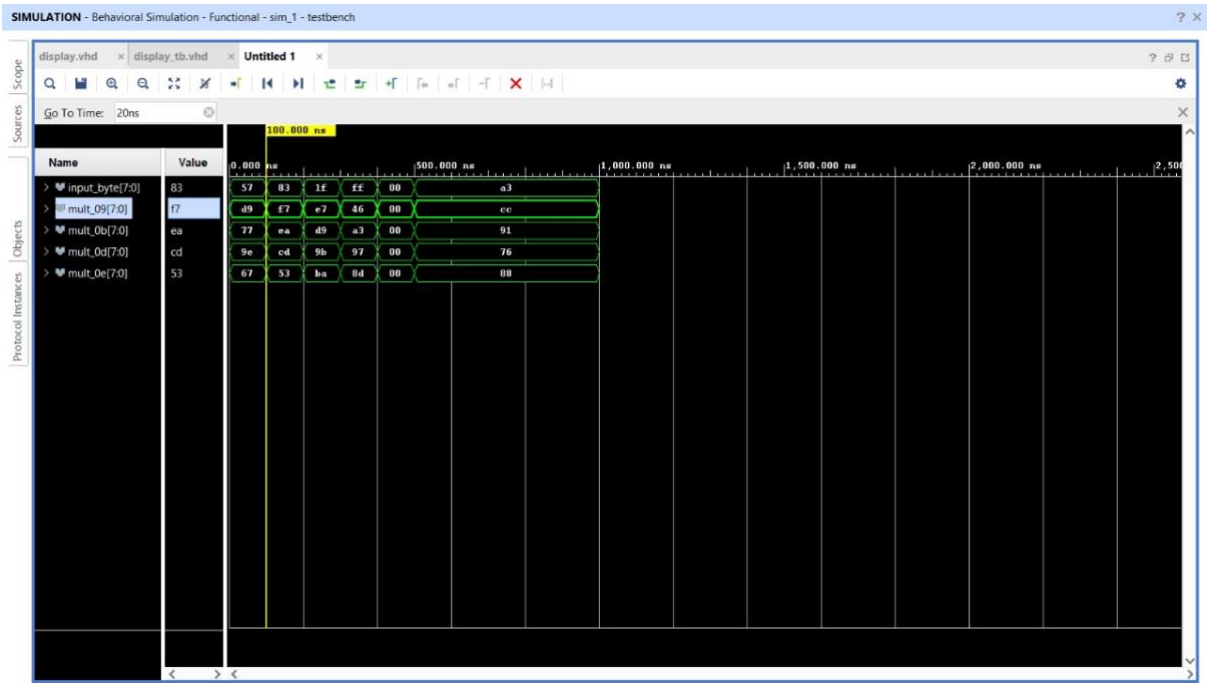
Basics of AES Encryption/Decryption

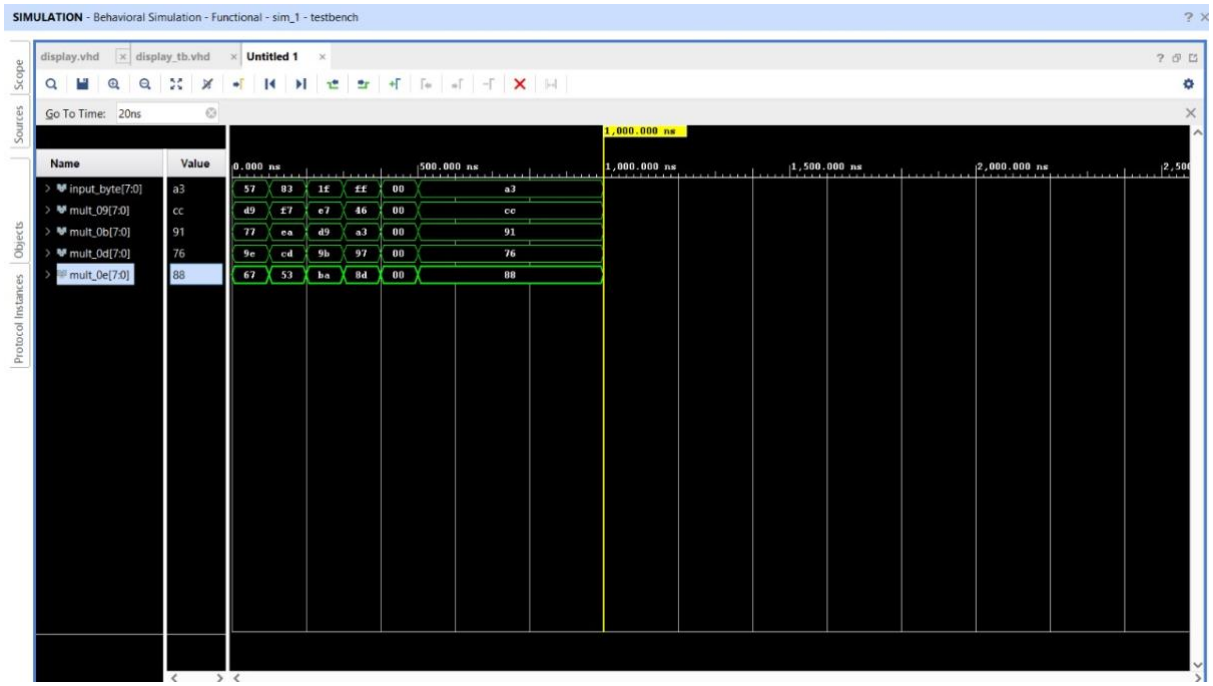
The Advanced Encryption Standard (AES) is a symmetric encryption algorithm that encrypts and decrypts data in blocks of 128 bits using keys of 128, 192, or 256 bits. The AES algorithm consists of several rounds, depending on the key size. Each round includes a series of transformations applied to the data. The decryption process mirrors the encryption process but with inverse operations. Below is a step-by-step explanation of both encryption and decryption operations

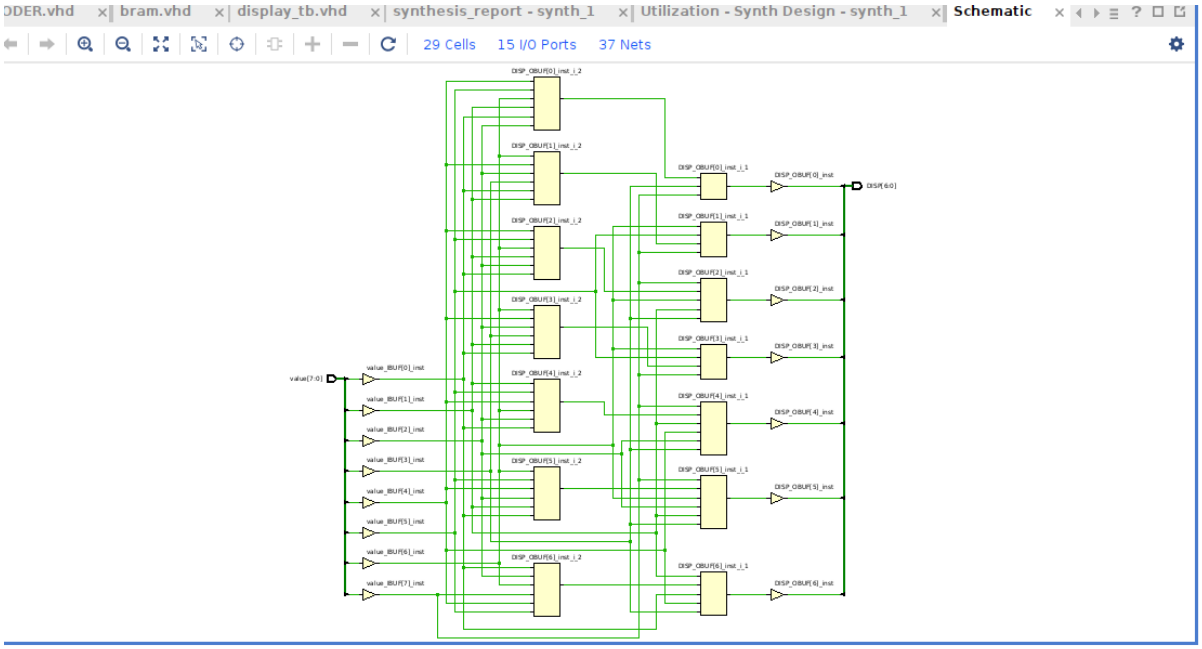
SIMULATION SNAPSHOT AND SCHEMATIC

DISPLAY UNIT :-

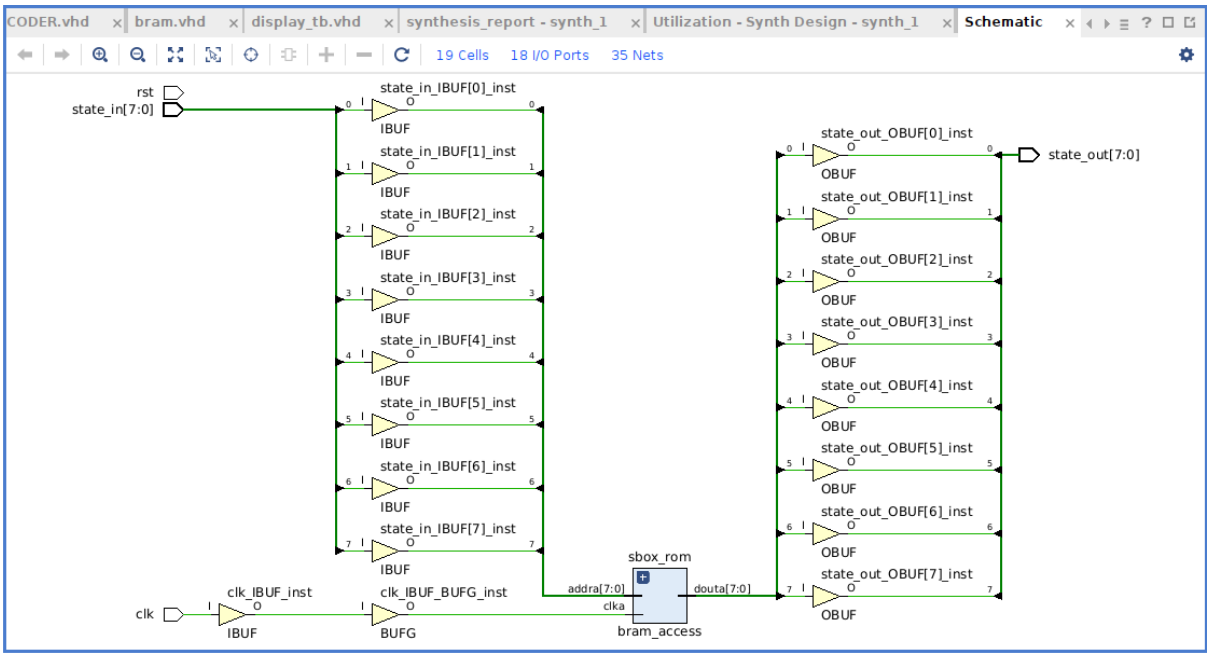


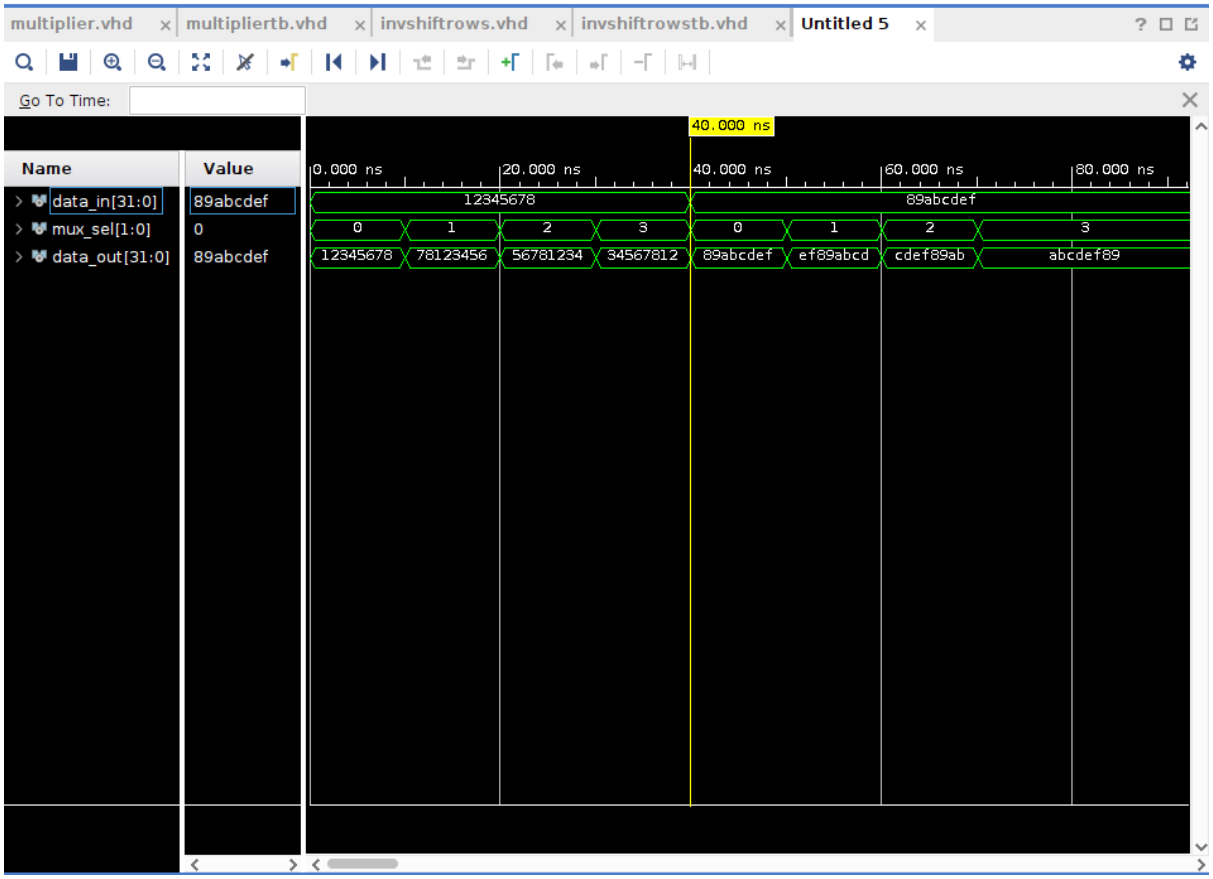


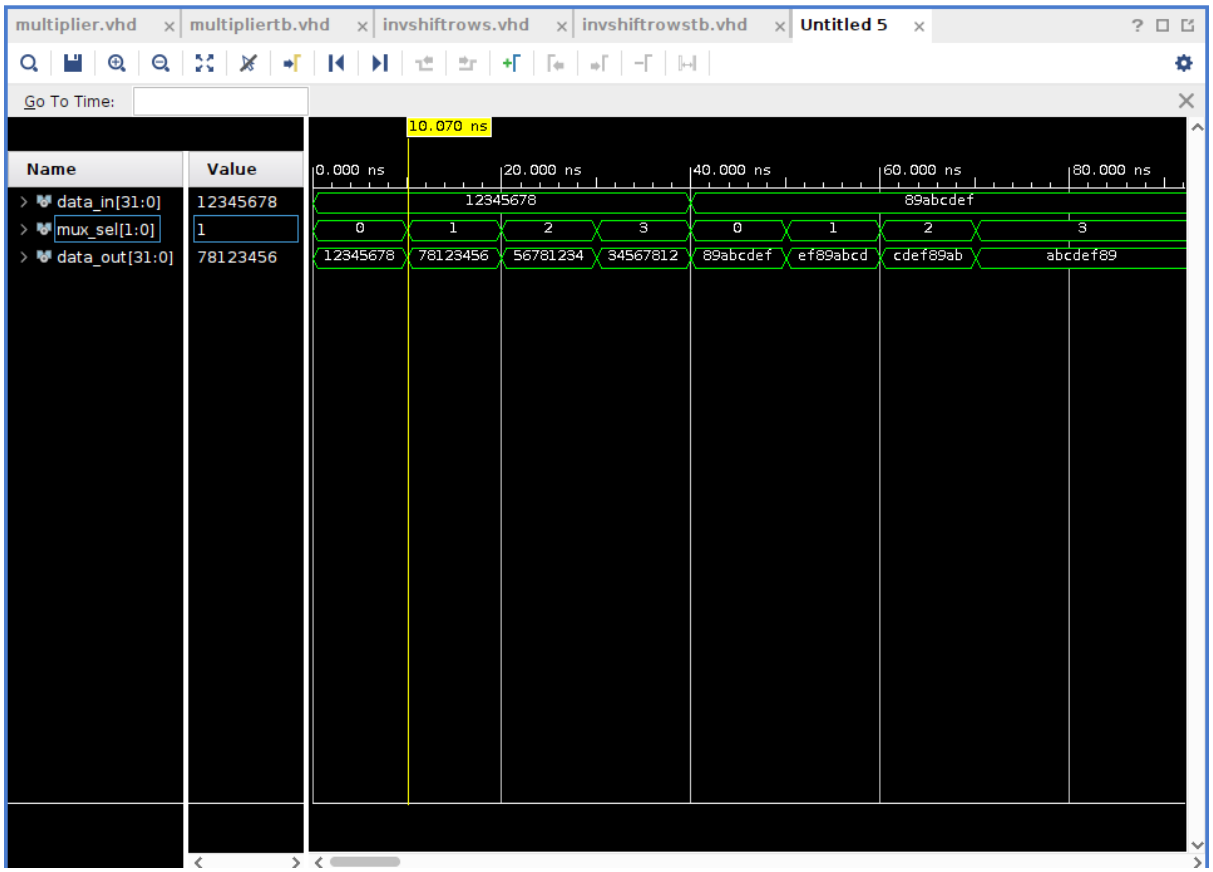

SIMULATION - Behavioral Simulation - Functional - sim_1 - testbench
display.vhd
display_tb.vhd
Untitled 1
Go To Time: 20ns
Name
Value
input_byte[7:0]
mult_09[7:0]
mult_0b[7:0]
mult_0d[7:0]
mult_0e[7:0]
a3
cc
91
76
88
0.000 ns
500.000 ns
1,000.000 ns
1,500.000 ns
2,000.000 ns

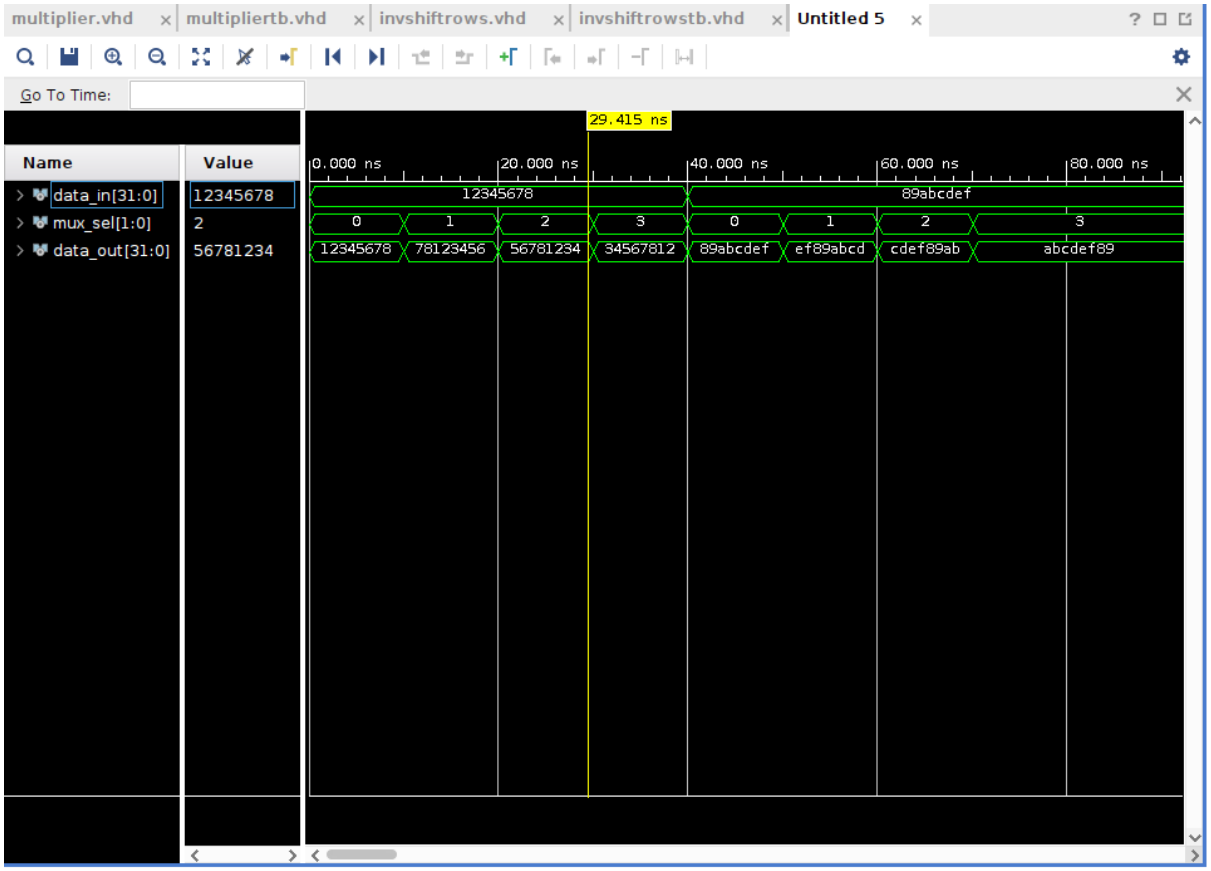

bram.vhd
display_tb.vhd
synthesis_report - synth_1
Utilization - Synth Design - synth_1
Schematic
29 Cells
15 I/O Ports
37 Nets

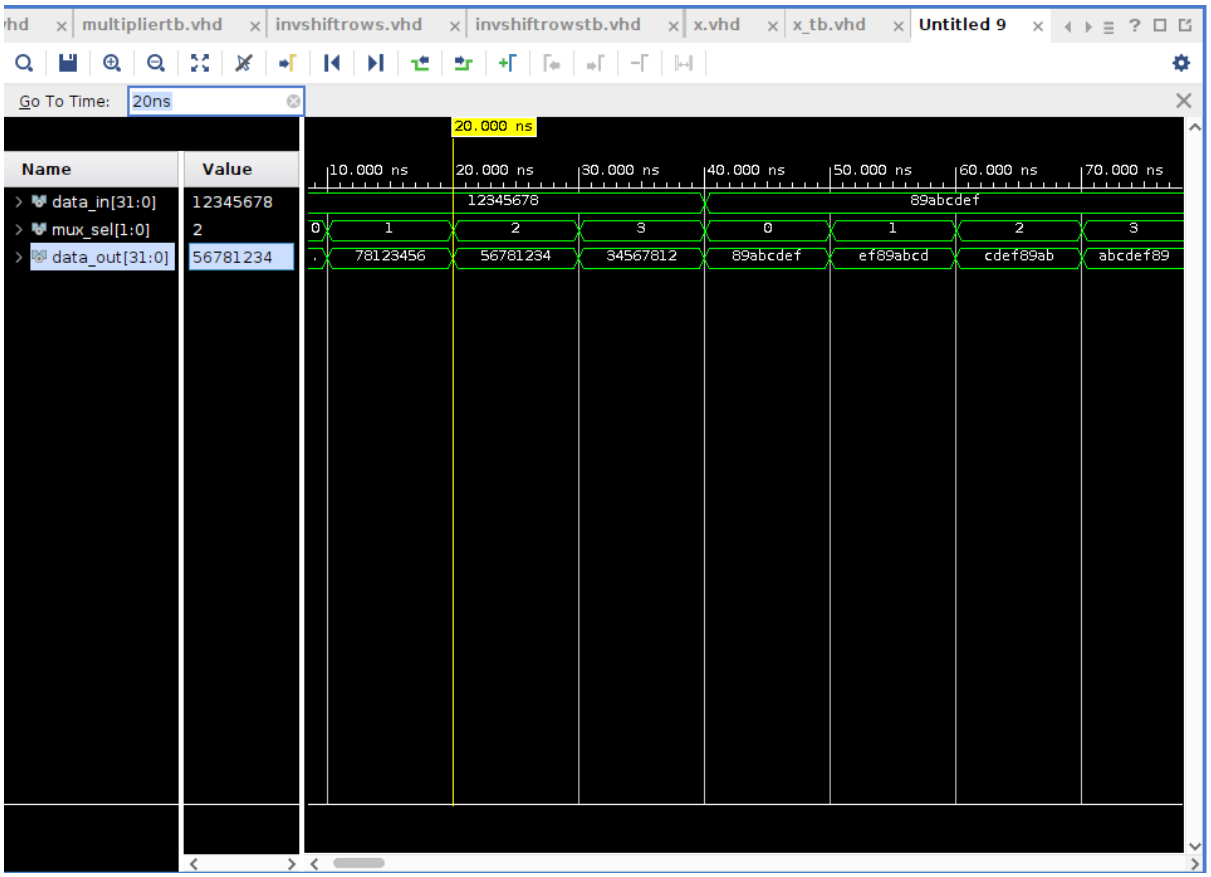
INVERSE SHIFT ROWS

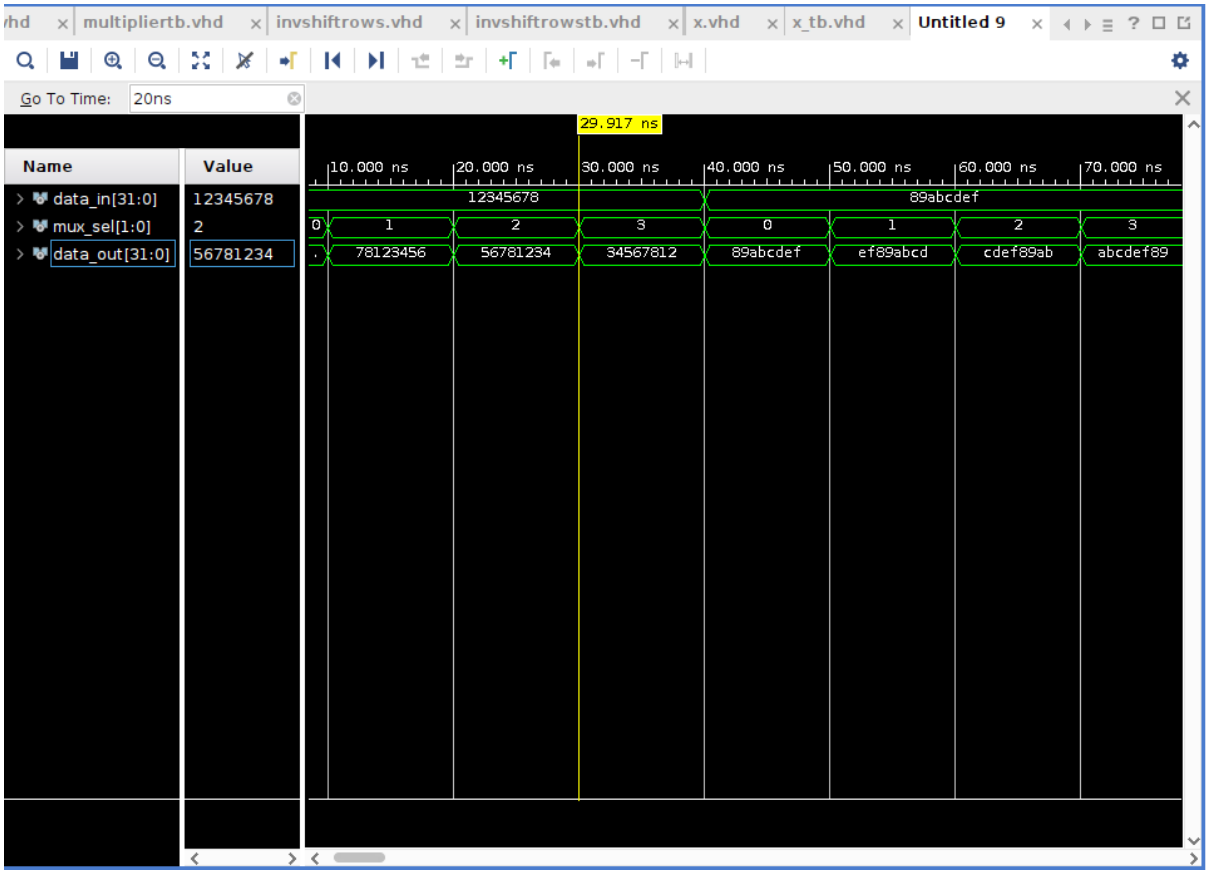



multiplier.vhd
multipliertb.vhd
invshiftrows.vhd
invshiftrowstb.vhd
Untitled 5
Go To Time:
40.000 ns
Name
Value
data_in[31:0]
89abcdef
mux_sel[1:0]
0
data_out[31:0]
89abcdef
0.000 ns
20.000 ns
40.000 ns
60.000 ns
80.000 ns
12345678
89abcdef
0
1
2
3
0
1
2
3
12345678
78123456
56781234
34567812
89abcdef
ef89abcd
cdef89ab
abcdef89

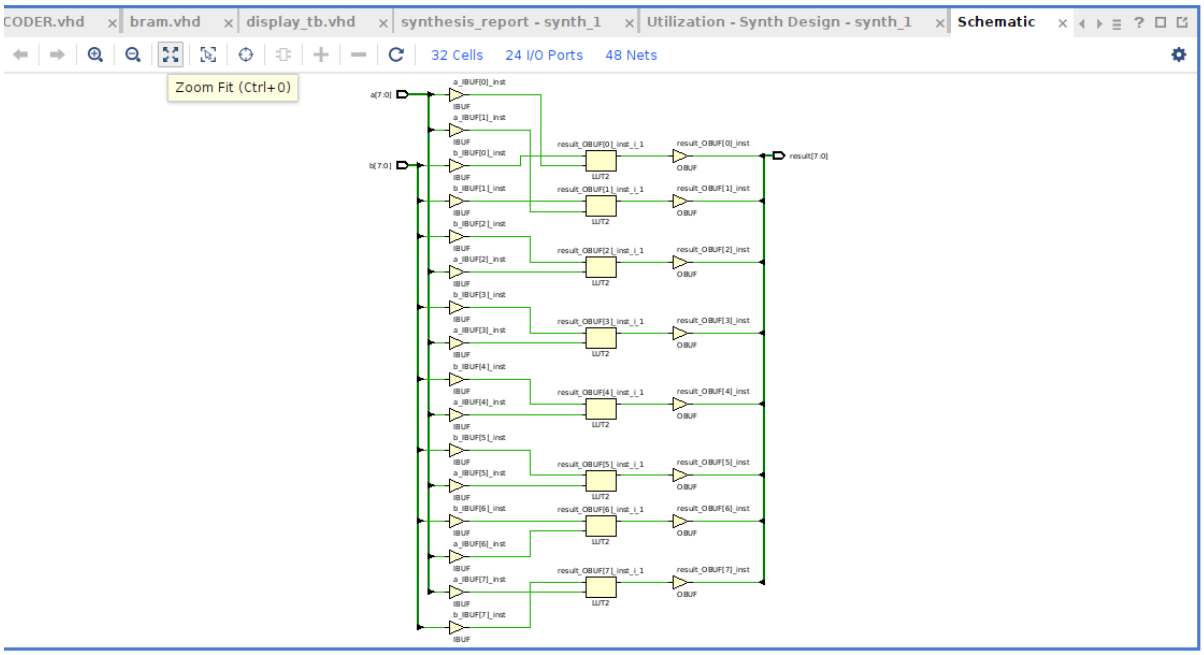

multiplier.vhd
multipliertb.vhd
invshiftrows.vhd
invshiftrowstb.vhd
Untitled 5
Go To Time:
10.070 ns
Name
Value
data_in[31:0]
12345678
mux_sel[1:0]
1
data_out[31:0]
78123456
0.000 ns
20.000 ns
40.000 ns
60.000 ns
80.000 ns
12345678
89abcdef
0
1
2
3
0
1
2
3
12345678
78123456
56781234
34567812
89abcdef
ef89abcd
cdef89ab
abcdef89

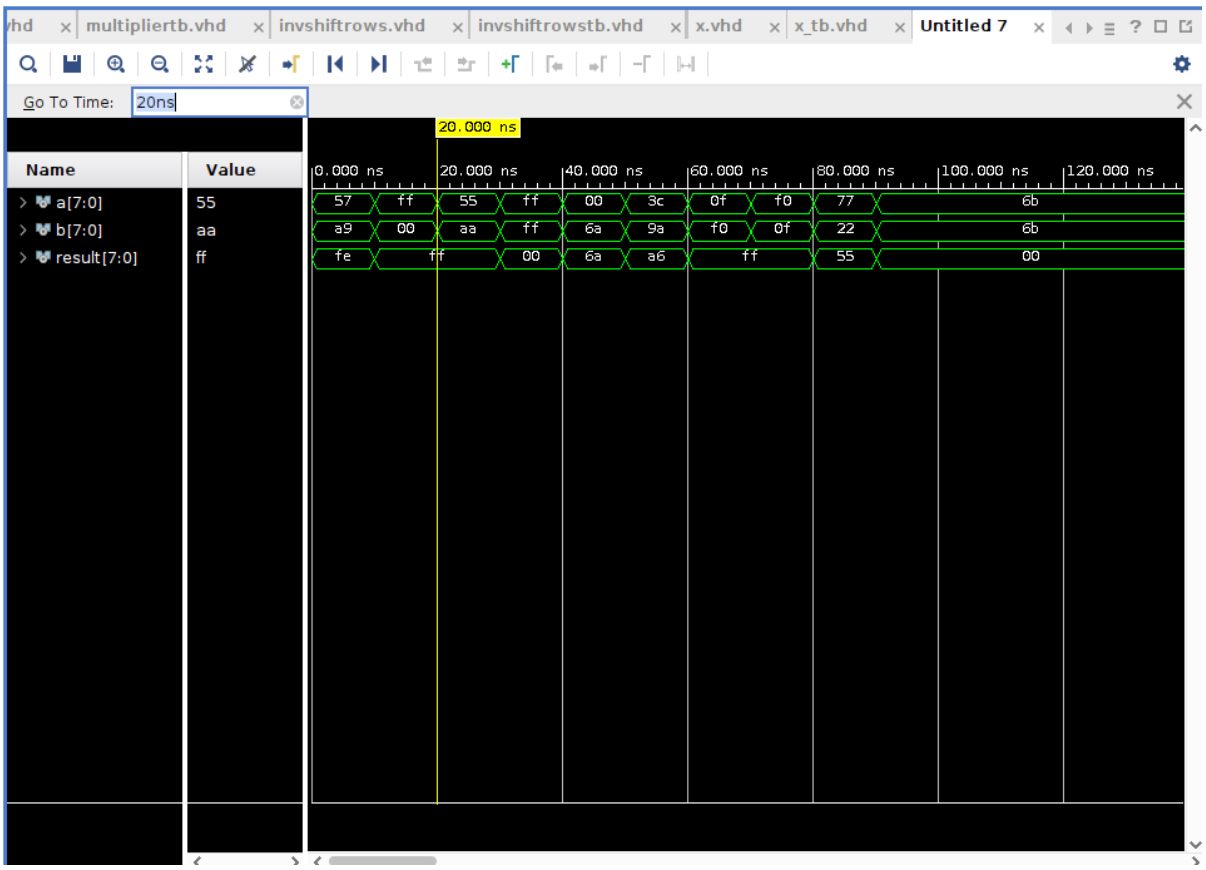

multiplier.vhd
multipliertb.vhd
invshiftrows.vhd
invshiftrowstb.vhd
Untitled 5
Go To Time:
29.415 ns
Name
Value
data_in[31:0]
12345678
mux_sel[1:0]
2
data_out[31:0]
56781234
0.000 ns
20.000 ns
40.000 ns
60.000 ns
80.000 ns
12345678
89abcdef
0
1
2
3
0
1
2
3
12345678
78123456
56781234
34567812
89abcdef
ef89abcd
cdef89ab
abcdef89

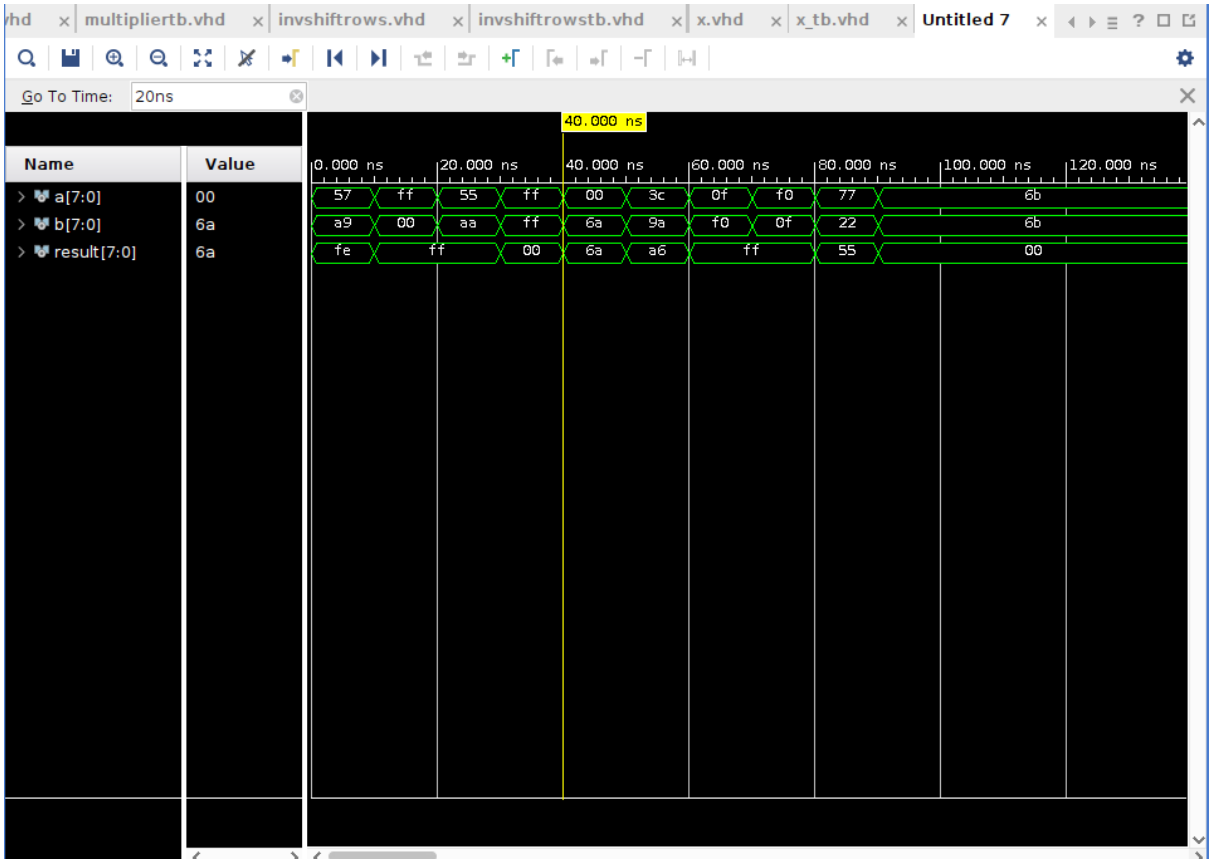

multipliertb.vhd
invshiftrows.vhd
invshiftrowstb.vhd
x.vhd
x_tb.vhd
Untitled 9
Go To Time:
20ns
20.000 ns
Name
Value
data_in[31:0]
12345678
mux_sel[1:0]
2
data_out[31:0]
56781234
10.000 ns
20.000 ns
30.000 ns
40.000 ns
50.000 ns
60.000 ns
70.000 ns
12345678
89abcdef
0
1
2
3
0
1
2
3
78123456
56781234
34567812
89abcdef
ef89abcd
cdef89ab
abcdef89



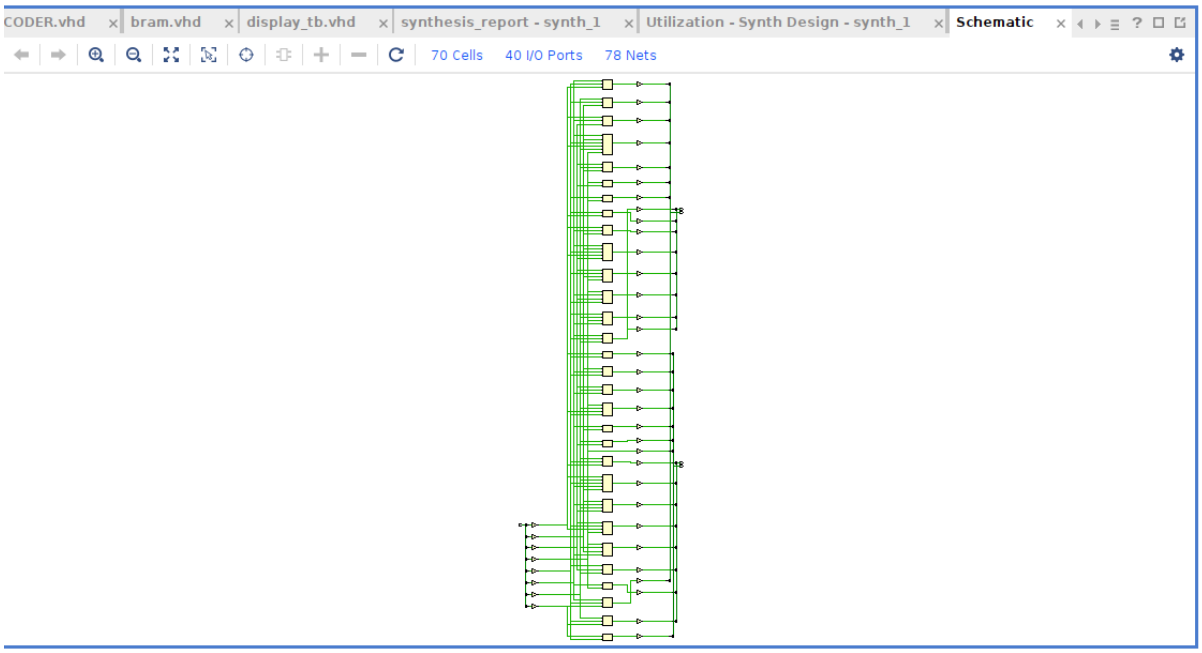
BITWISE XOR OPERATION

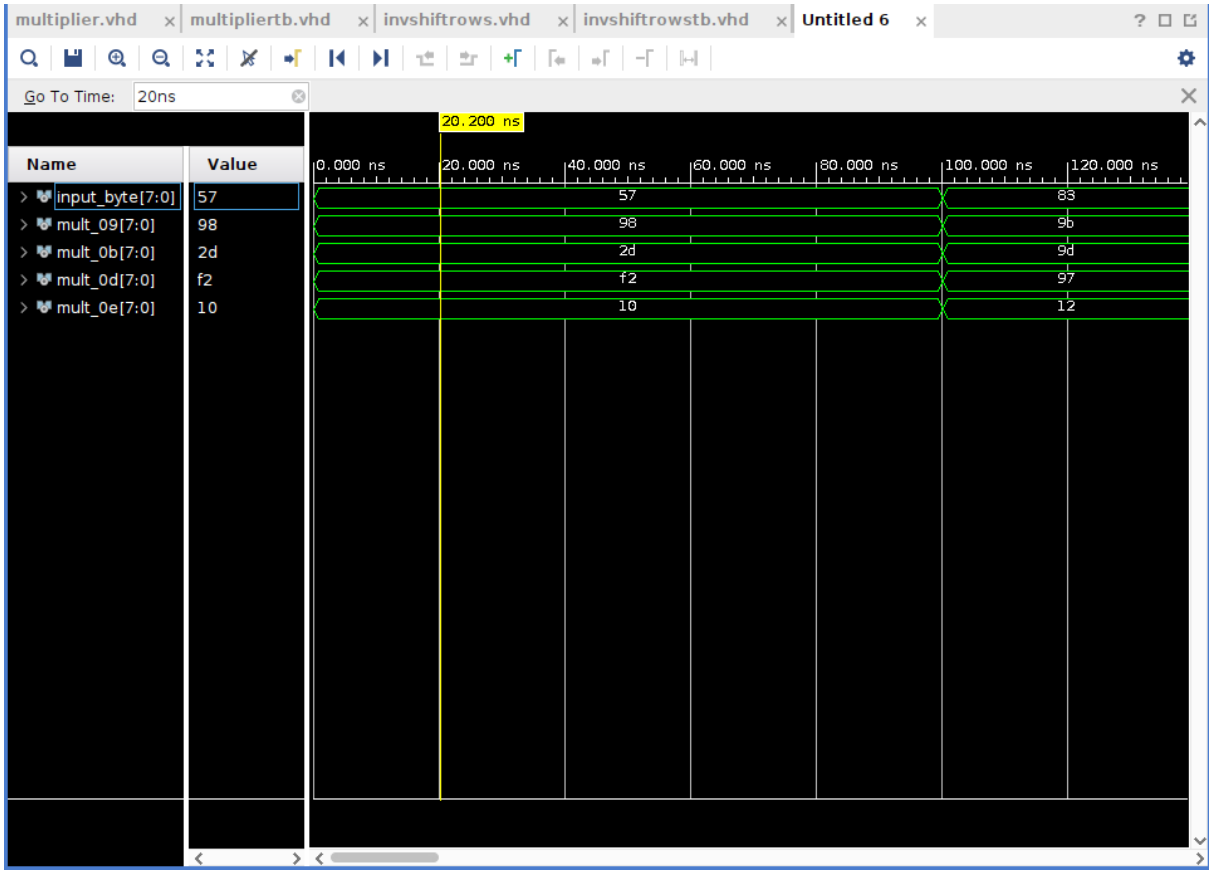

CODER.vhd
bram.vhd
display_tb.vhd
synthesis_report - synth_1
Utilization - Synth Design - synth_1
Schematic
32 Cells
24 I/O Ports
48 Nets
Zoom Fit (Ctrl+0)
a[7:0]
b[7:0]
result[7:0]
IBUF
OBUF
LUT2

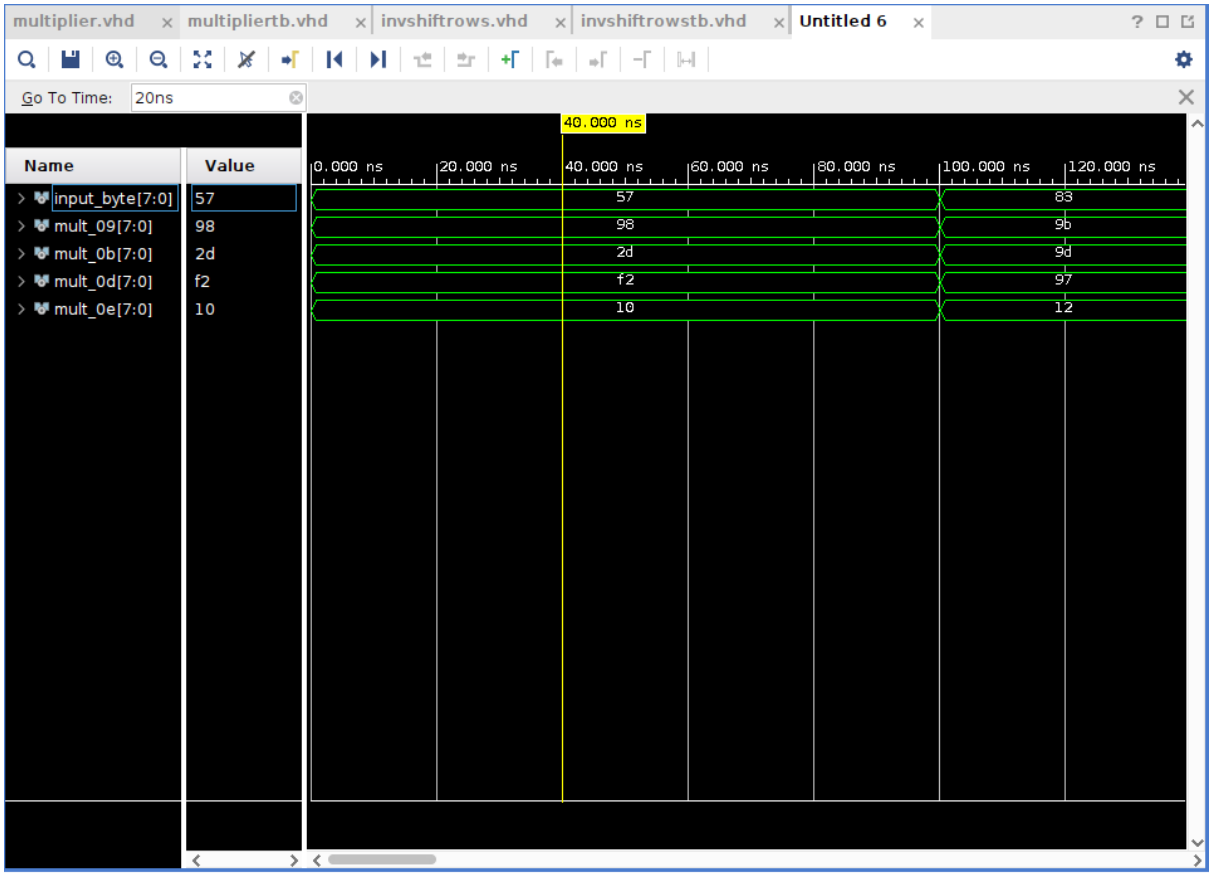

multiplier_tb.vhd
invshiftrows.vhd
invshiftrowstb.vhd
x.vhd
x_tb.vhd
Untitled 7
Go To Time:
20ns
20.000 ns
Name
Value
a[7:0]
55
b[7:0]
aa
result[7:0]
ff
0.000 ns
20.000 ns
40.000 ns
60.000 ns
80.000 ns
100.000 ns
120.000 ns
57 ff 55 ff 00 3c 0f f0 77 6b
a9 00 aa ff 6a 9a f0 0f 22 6b
fe ff 00 6a a6 ff 55 00

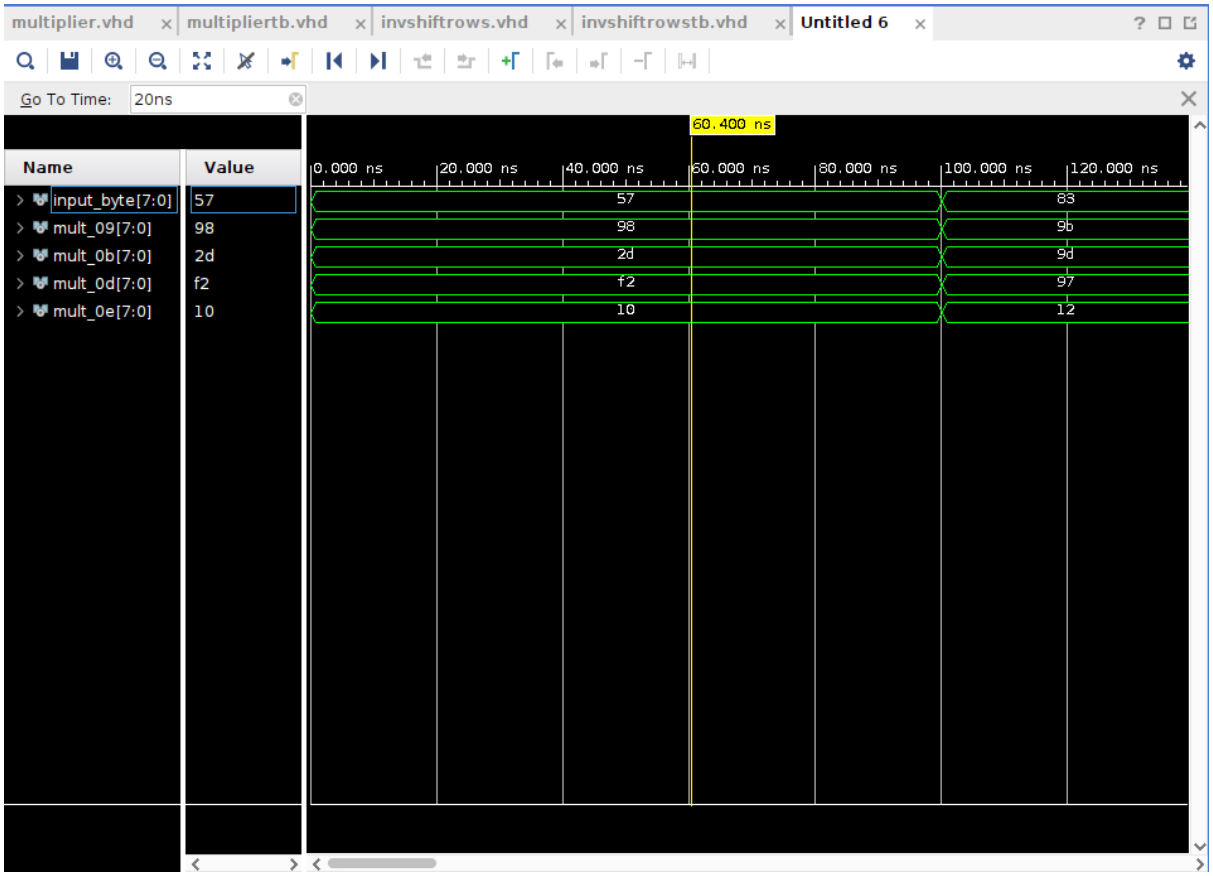


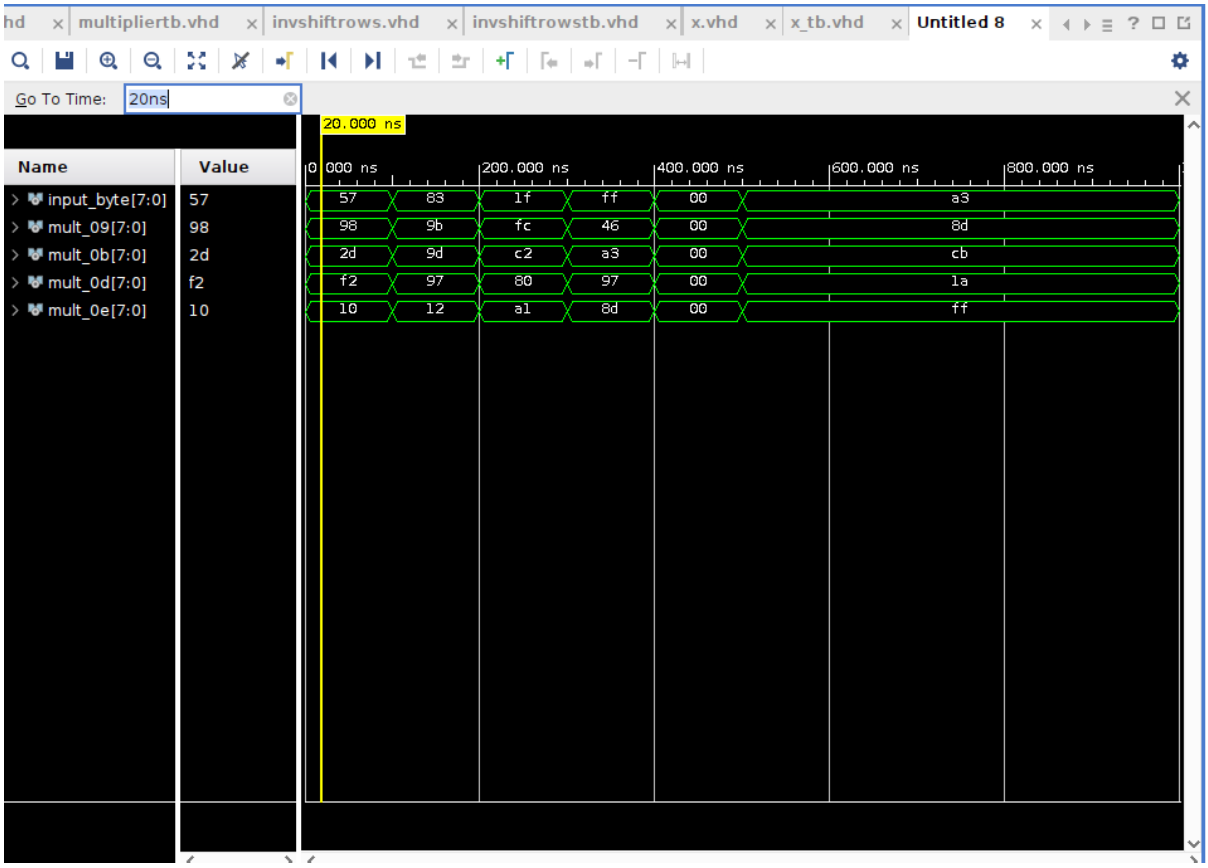
INVERSE MIX COLUMN


CODER.vhd
bram.vhd
display_tb.vhd
synthesis_report - synth_1
Utilization - Synth Design - synth_1
Schematic
70 Cells
40 I/O Ports
78 Nets

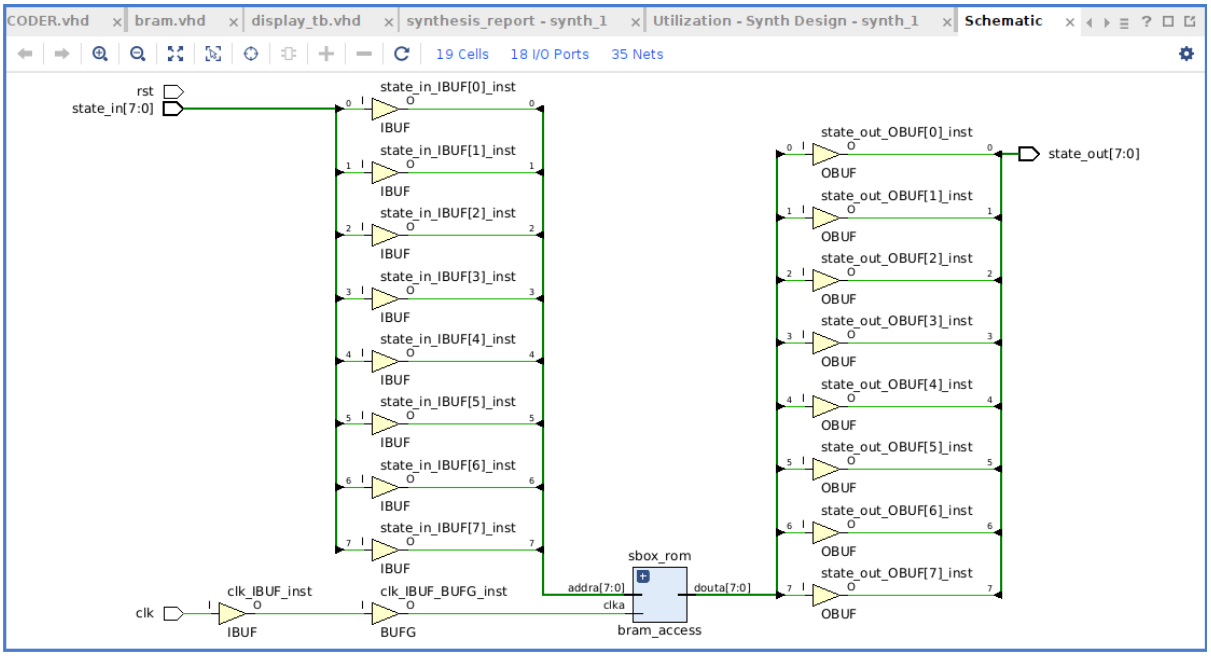

multiplier.vhd
multipliertb.vhd
invshiftrows.vhd
invshiftrowstb.vhd
Untitled 6
Go To Time:
20ns
20.200 ns
Name
Value
input_byte[7:0]
mult_09[7:0]
mult_0b[7:0]
mult_0d[7:0]
mult_0e[7:0]
57
98
2d
f2
10
0.000 ns
20.000 ns
40.000 ns
60.000 ns
80.000 ns
100.000 ns
120.000 ns
83
9b
9d
97
12

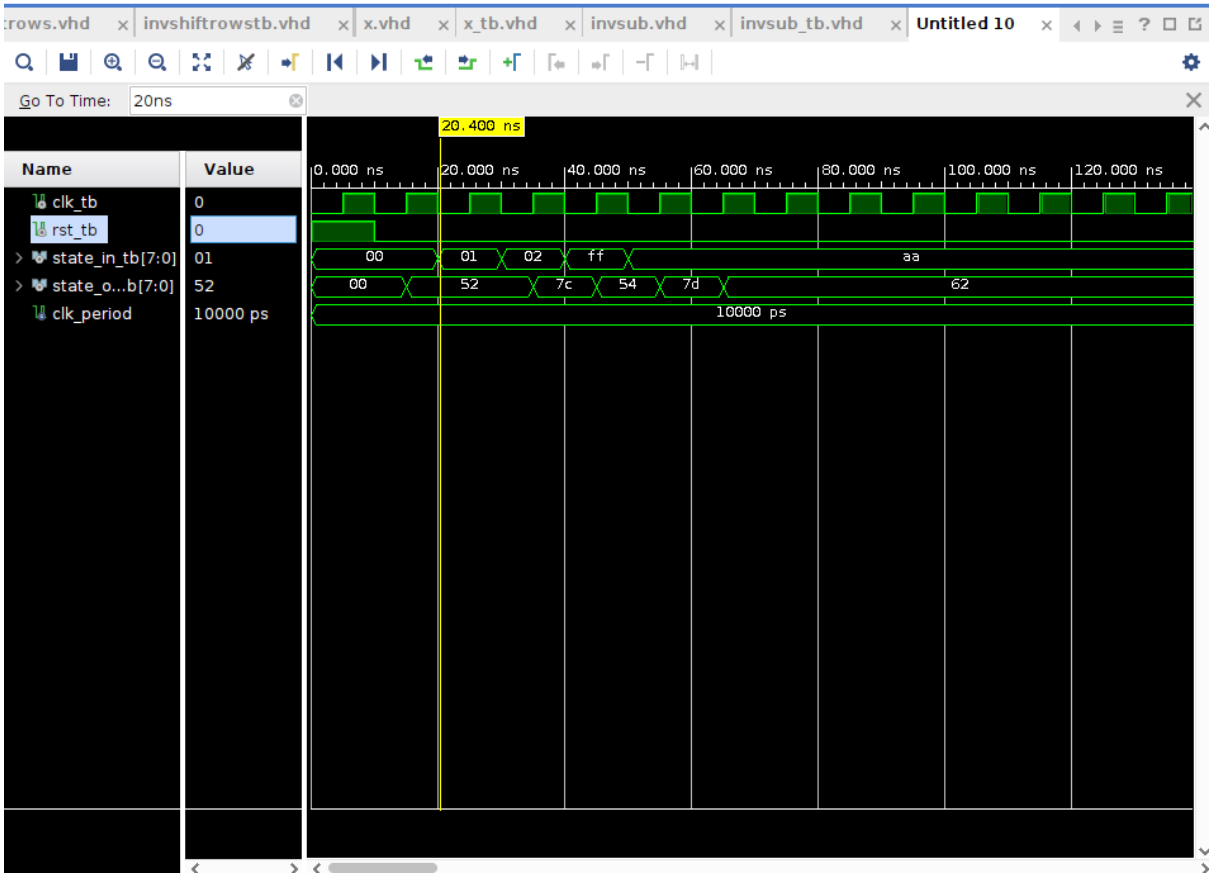

multiplier.vhd
multipliertb.vhd
invshiftrows.vhd
invshiftrowstb.vhd
Untitled 6
Go To Time: 20ns
40.000 ns
Name
Value
0.000 ns
20.000 ns
40.000 ns
60.000 ns
80.000 ns
100.000 ns
120.000 ns
input_byte[7:0] 57
mult_09[7:0] 98
mult_0b[7:0] 2d
mult_0d[7:0] f2
mult_0e[7:0] 10
57 83
98 9b
2d 9d
f2 97
10 12

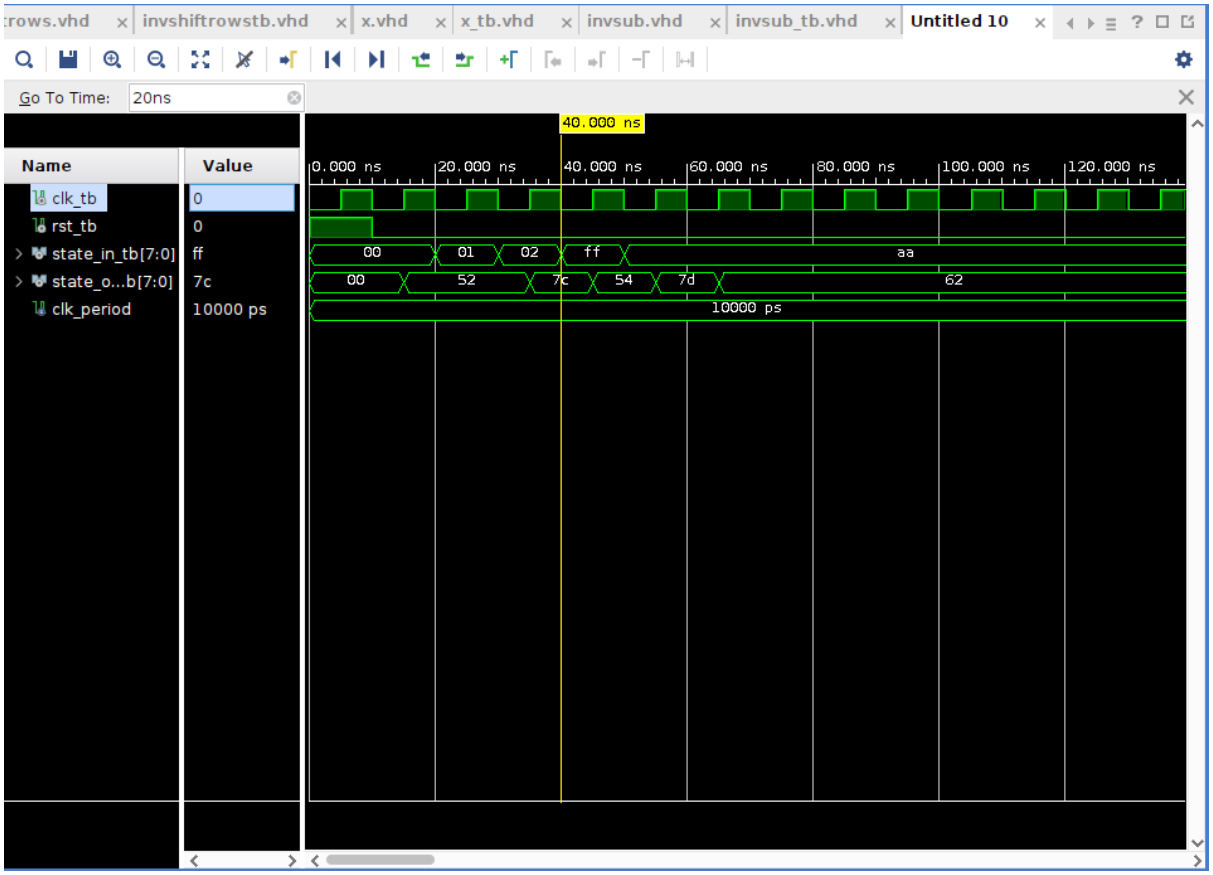

multiplier.vhd
multipliertb.vhd
invshiftrows.vhd
invshiftrowstb.vhd
Untitled 6
Go To Time: 20ns
60.400 ns
Name
Value
0.000 ns
20.000 ns
40.000 ns
60.000 ns
80.000 ns
100.000 ns
120.000 ns
input_byte[7:0] 57
mult_09[7:0] 98
mult_0b[7:0] 2d
mult_0d[7:0] f2
mult_0e[7:0] 10
57 83
98 9b
2d 9d
f2 97
10 12

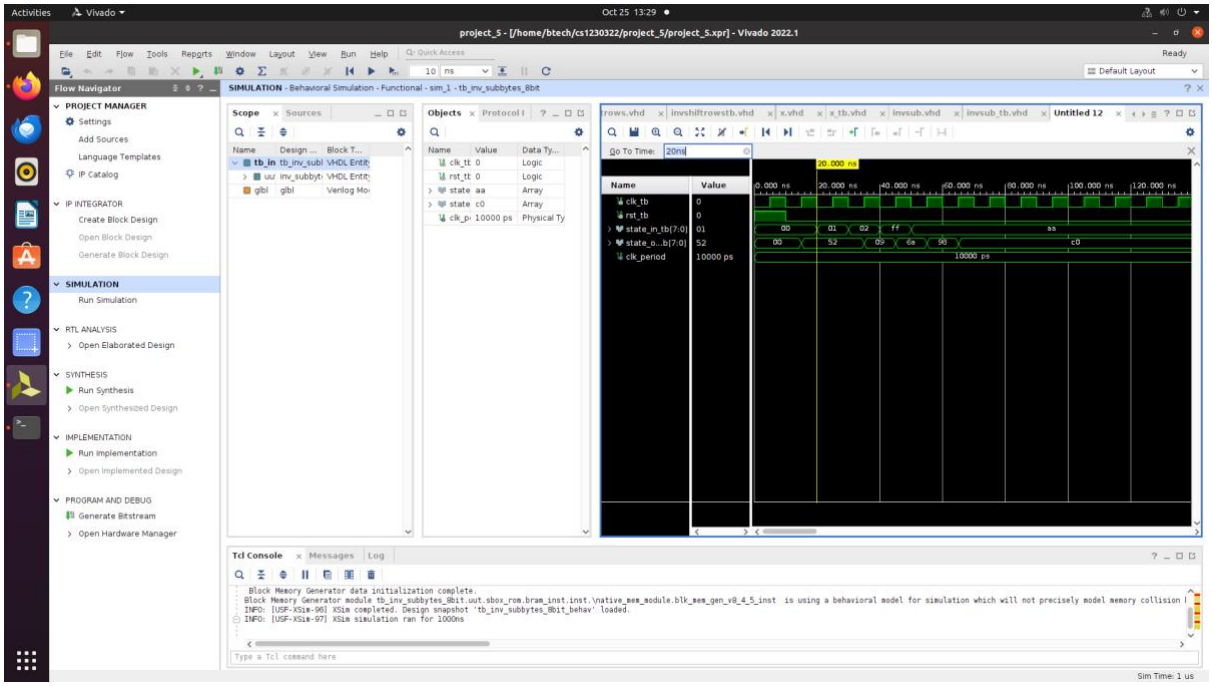


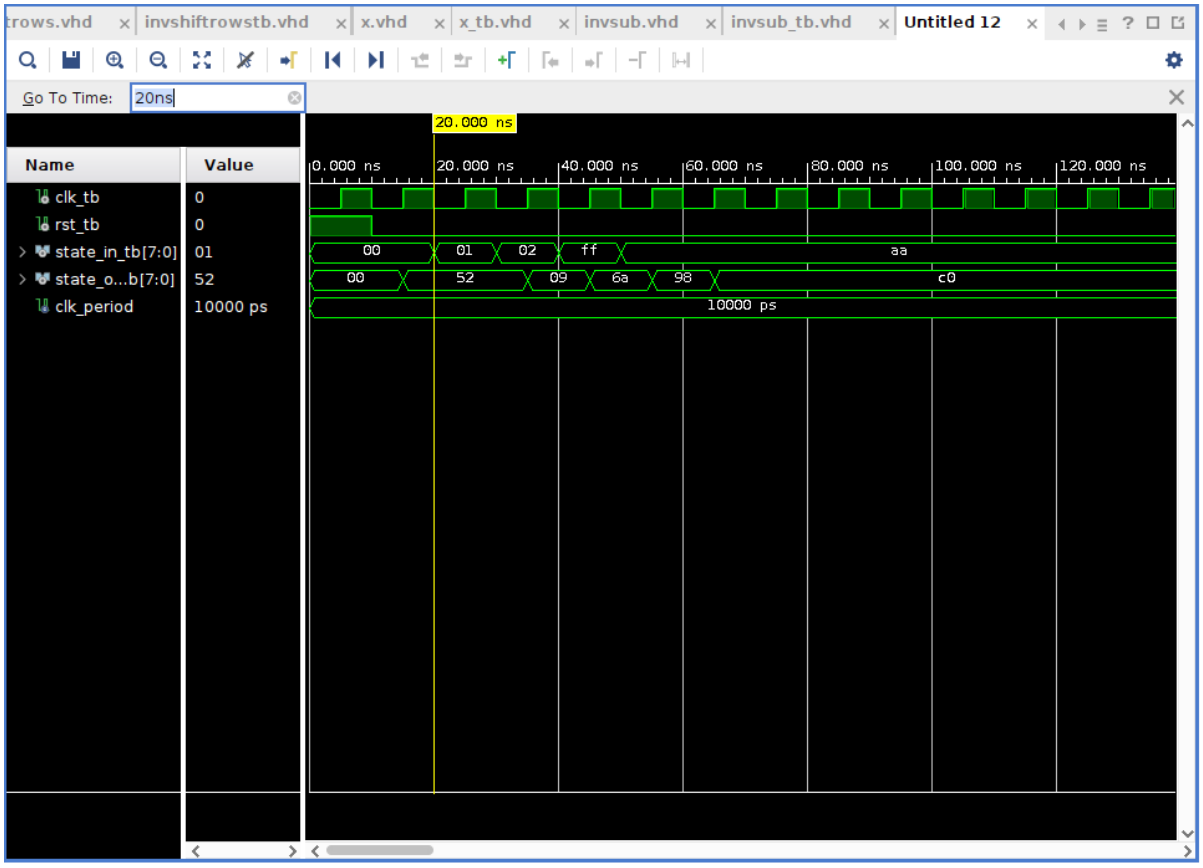
INVERSE SUB BYTES

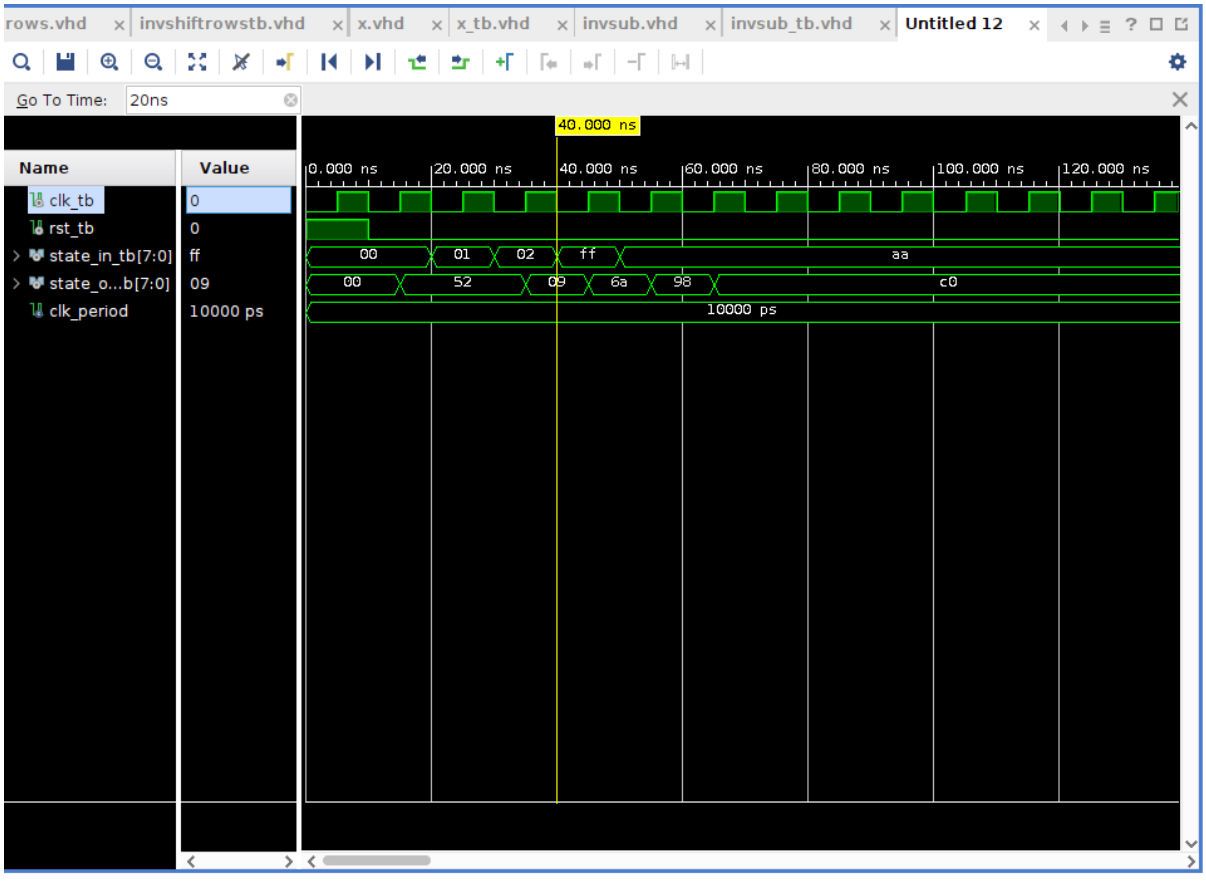

CODER.vhd
bram.vhd
display_tb.vhd
synthesis_report - synth_1
Utilization - Synth Design - synth_1
Schematic
19 Cells
18 I/O Ports
35 Nets
rst
state_in[7:0]
state_in_IBUF[0]_inst
state_in_IBUF[1]_inst
state_in_IBUF[2]_inst
state_in_IBUF[3]_inst
state_in_IBUF[4]_inst
state_in_IBUF[5]_inst
state_in_IBUF[6]_inst
state_in_IBUF[7]_inst
IBUF
state_out_OBUF[0]_inst
state_out_OBUF[1]_inst
state_out_OBUF[2]_inst
state_out_OBUF[3]_inst
state_out_OBUF[4]_inst
state_out_OBUF[5]_inst
state_out_OBUF[6]_inst
state_out_OBUF[7]_inst
OBUF
state_out[7:0]
sbox_rom
addra[7:0]
clka
douta[7:0]
bram_access
clk
clk_IBUF_inst
clk_IBUF_BUFG_inst
BUFG


rows.vhd
invshiftrowstb.vhd
x.vhd
x_tb.vhd
invsub.vhd
invsub_tb.vhd
Untitled 10
Go To Time:
20ns
20.400 ns
Name
Value
clk_tb
rst_tb
state_in_tb[7:0]
state_o...b[7:0]
clk_period
0
0
01
52
10000 ps
0.000 ns
20.000 ns
40.000 ns
60.000 ns
80.000 ns
100.000 ns
120.000 ns
00
01
02
ff
aa
00
52
7c
54
7d
62
10000 ps


rows.vhd
invshiftrowstb.vhd
x.vhd
x_tb.vhd
invsub.vhd
invsub_tb.vhd
Untitled 10
Go To Time: 20ns
40.000 ns
Name
Value
clk_tb
0
rst_tb
0
state_in_tb[7:0]
ff
state_o...b[7:0]
7c
clk_period
10000 ps
0.000 ns
20.000 ns
40.000 ns
60.000 ns
80.000 ns
100.000 ns
120.000 ns
00
01
02
ff
aa
00
52
7c
54
7d
62
10000 ps


Activities
Vivado
Oct 25 13:29
project_5 - [/home/btech/cs1230322/project_5/project_5.xpr] - Vivado 2022.1
File Edit Flow Tools Reports Window Layout View Run Help
Ready
Default Layout
Flow Navigator
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
PROGRAM AND DEBUG
Generate Bitstream
Open Hardware Manager
SIMULATION - Behavioral Simulation - Functional - sim_1 - tb_inv_subbytes_8bit
Scope
Sources
Objects
Protocol I
Tcl Console
Messages
Log
Type a Tcl command here
Sim Time: 1 us


trows.vhd
invshiftrowstb.vhd
x.vhd
x_tb.vhd
invsub.vhd
invsub_tb.vhd
Untitled 12
Go To Time:
20ns
Name
Value
clk_tb
0
rst_tb
0
state_in_tb[7:0]
01
state_o...b[7:0]
52
clk_period
10000 ps
20.000 ns
0.000 ns
20.000 ns
40.000 ns
60.000 ns
80.000 ns
100.000 ns
120.000 ns
00
01
02
ff
aa
00
52
09
6a
98
c0
10000 ps


rows.vhd
invshiftrowstb.vhd
x.vhd
x_tb.vhd
invsub.vhd
invsub_tb.vhd
Untitled 12
Go To Time:
20ns
Name
Value
clk_tb
0
rst_tb
0
state_in_tb[7:0]
ff
state_o...b[7:0]
09
clk_period
10000 ps
40.000 ns
0.000 ns
20.000 ns
40.000 ns
60.000 ns
80.000 ns
100.000 ns
120.000 ns
00
01
02
ff
aa
00
52
09
6a
98
c0
10000 ps

THE S BOX IS STORED IN THE ROM AND THIS IS TESTED USING THE INVERSE SUB BYTES BECAUSE WE ARE READING FROM THIS FILE.

NOTE:- SIR CROPPING THE IMAGES MADE THEM BLURRY SO WE HAVE ATTACHED IT LIKE THIS ONLY