Project 2 - SPI Slave with Single Port RAM

Team members:

Omar Ahmed Ragab

Modules:

RAM module:

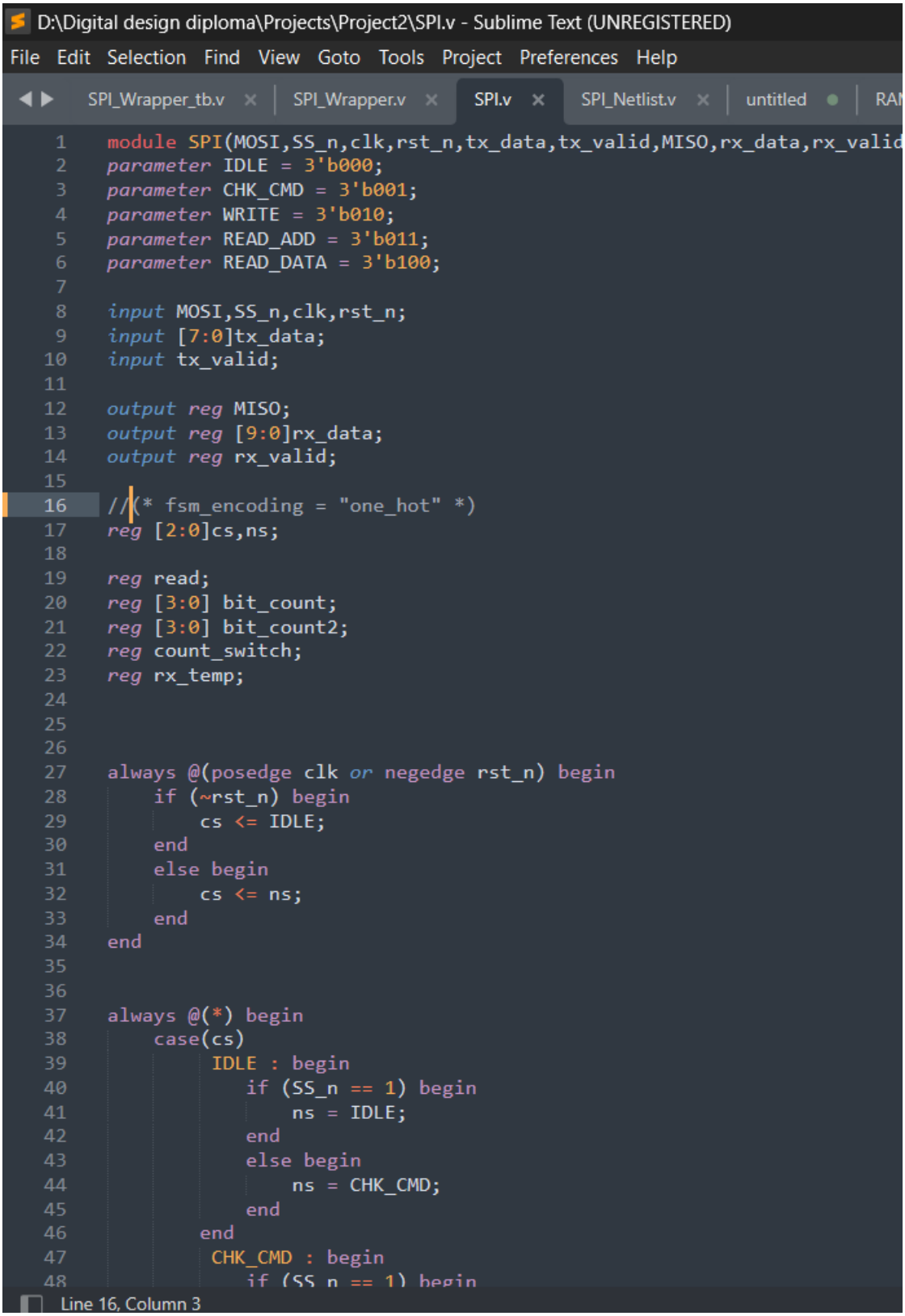
D:\Digital design diploma\Projects\Project2\RAM.v • - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

SPI_Wrapper_tb.v | SPI_Wrapper.v | SPI.v | SPI_Netlist.v | untitled

```
1   module RAM(clk,rst_n,rx_valid,din,dout,tx_valid);
2   parameter MEM_DEPTH = 256;
3   parameter ADDR_SIZE = 8;
4   input clk,rst_n,rx_valid;
5   input [ADDR_SIZE+1:0]din;
6   output reg [ADDR_SIZE-1:0]dout;
7   output reg tx_valid;
8
9
10  reg [ADDR_SIZE-1:0]mem[MEM_DEPTH-1:0];
11  reg [ADDR_SIZE-1:0]temp_address;
12
13
14  always @(posedge clk) begin
15      if (~rst_n) begin
16          dout <= 0;
17          tx_valid <= 0;
18          temp_address <= 0;
19      end
20      else begin
21          case(din[9:8])
22               2'b00 : if (rx_valid) begin
23                  temp_address <= din[ADDR_SIZE-1:0];
24                  tx_valid <= 0;
25              end
26               2'b01 : if (rx_valid) begin
27                  mem[temp_address] <= din[ADDR_SIZE-1:0];
28                  tx_valid <= 0;
29              end
30               2'b10 : begin
31                  if (rx_valid) begin
32                      temp_address <= din[ADDR_SIZE-1:0];
33                  end
34              end
35               2'b11 : begin
36                  dout <= mem[temp_address];
37                  tx_valid <= 1;
38              end
39              default : tx_valid <= 0;
40
41          endcase
42      end
43  end
44  endmodule
```

SPI module:




```
1   module SPI(MOSI,SS_n,clk,rst_n,tx_data,tx_valid,MISO,rx_data,rx_valid
2   parameter IDLE = 3'b000;
3   parameter CHK_CMD = 3'b001;
4   parameter WRITE = 3'b010;
5   parameter READ_ADD = 3'b011;
6   parameter READ_DATA = 3'b100;
7
8   input MOSI,SS_n,clk,rst_n;
9   input [7:0]tx_data;
10  input tx_valid;
11
12  output reg MISO;
13  output reg [9:0]rx_data;
14  output reg rx_valid;
15
16  //(* fsm_encoding = "one_hot" *)
17  reg [2:0]cs,ns;
18
19  reg read;
20  reg [3:0] bit_count;
21  reg [3:0] bit_count2;
22  reg count_switch;
23  reg rx_temp;
24
25
26
27  always @(posedge clk or negedge rst_n) begin
28      if (~rst_n) begin
29          cs <= IDLE;
30      end
31      else begin
32          cs <= ns;
33      end
34  end
35
36
37  always @(*) begin
38      case(cs)
39           IDLE : begin
40              if (SS_n == 1) begin
41                  ns = IDLE;
42              end
43              else begin
44                  ns = CHK_CMD;
45              end
46          end
47           CHK_CMD : begin
48              if (SS_n == 1) begin
```

D:\Digital design diploma\Projects\Project2\SPI.v - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

SPI_Wrapper_tb.v | SPI_Wrapper.v | SPI.v | SPI_Netlist.v | untitled | RAM.v

```
45                  end
46              end
47               CHK_CMD : begin
48                  if (SS_n == 1) begin
49                      ns = IDLE;
50                  end
51                  else if(SS_n == 0 && MOSI == 0) begin
52                      ns = WRITE;
53                  end
54                  else if (SS_n == 0 && MOSI == 1 && read == 1) begin
55                      ns = READ_DATA;
56                  end
57                  else if (SS_n == 0 && MOSI == 1 && read == 0) begin
58                      ns = READ_ADD;
59                  end
60                  else begin
61                      ns = cs;
62                  end
63              end
64               WRITE : begin
65                  if(SS_n == 0 && bit_count < 4'd10) begin
66                      ns = WRITE;
67                  end
68                  else begin
69                      ns = IDLE;
70                  end
71              end
72               READ_ADD : begin
73                  if(SS_n == 0 && bit_count < 4'd10) begin
74                          ns = READ_ADD;
75                      end
76                  else begin
77                      ns = IDLE;
78                  end
79              end
80               READ_DATA : begin
81                  if(SS_n == 0 && bit_count2 < 4'd8) begin
82                          ns = READ_DATA;
83                      end
84                      else begin
85                          ns = IDLE;
86                      end
87              end
88          endcase
89      end
90
```

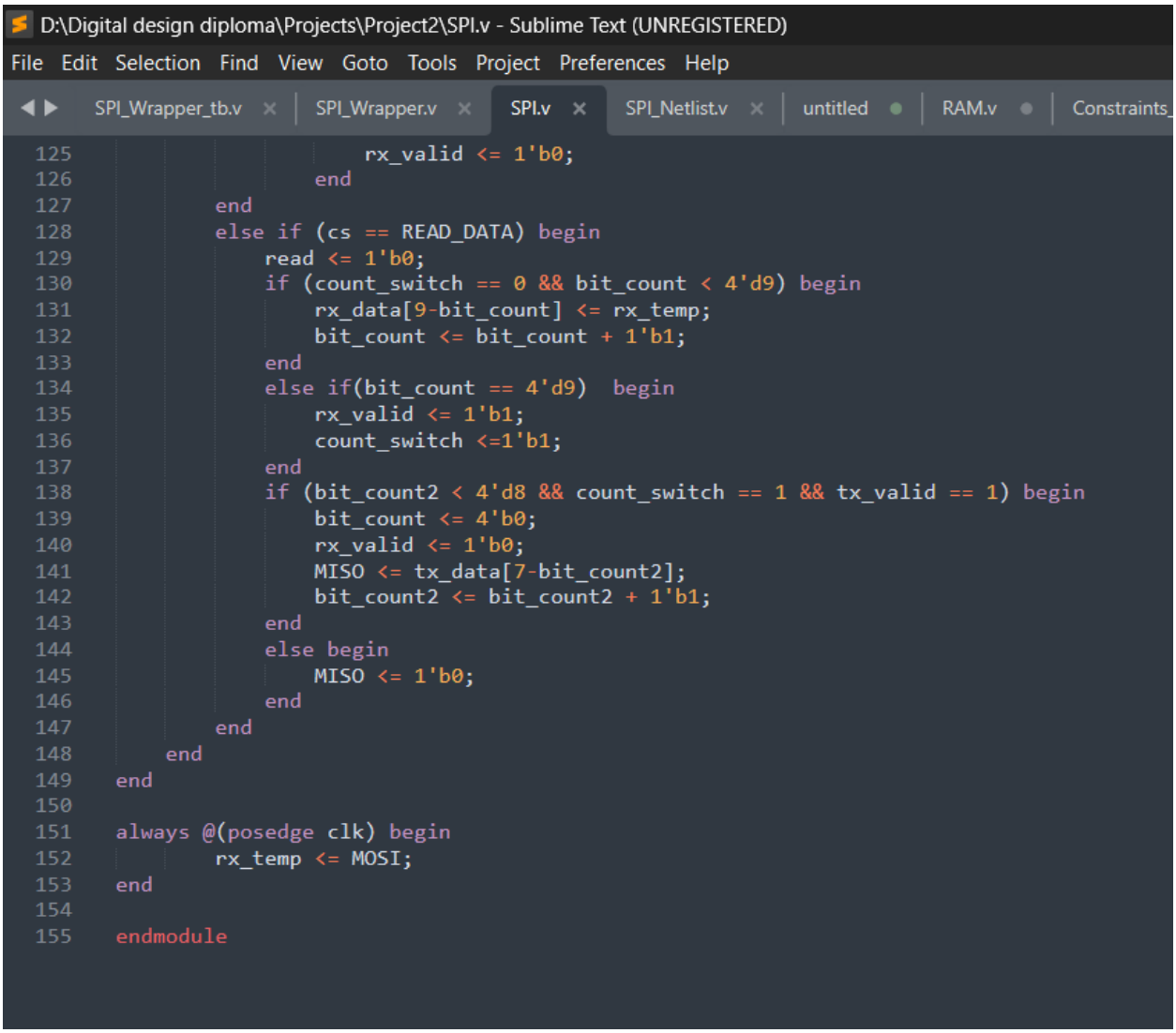
D:\Digital design diploma\Projects\Project2\SPI.v - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

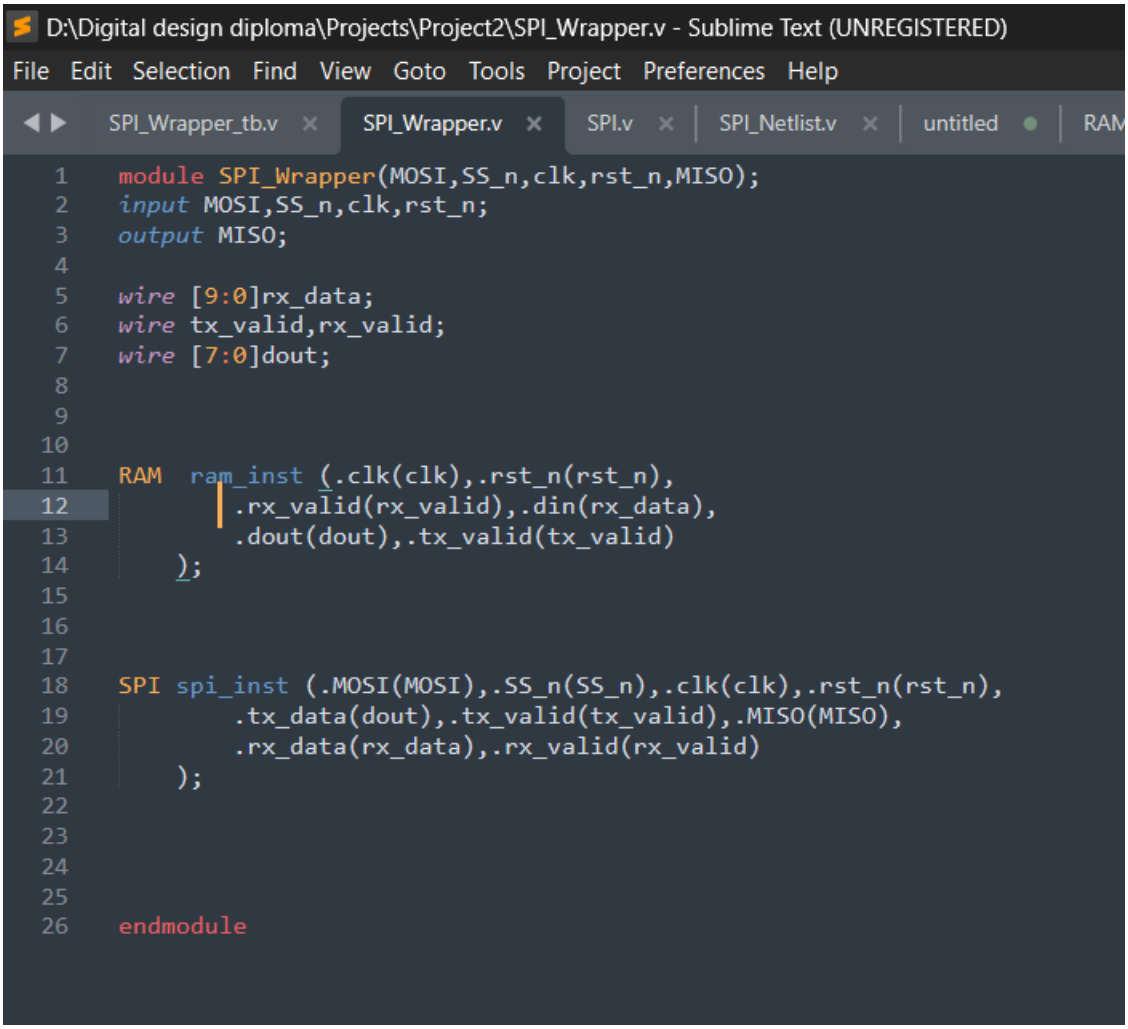
SPI_Wrapper_tb.v | SPI_Wrapper.v | SPI.v | SPI_Netlist.v | untitled | RAM.v

```
 89    end
 90
 91    always @(posedge clk or negedge rst_n) begin
 92        if(~rst_n) begin
 93            rx_valid <= 1'b0;
 94            rx_data <= 10'b0;
 95            bit_count <= 4'b0;
 96            bit_count2 <= 4'b0;
 97            count_switch <= 1'b0;
 98            read <= 1'b0;
 99            MISO <= 1'b0;
100        end
101        else begin
102            if(cs == IDLE) begin
103                rx_valid <= 1'b0;
104                rx_data <= 0;
105                bit_count <= 0;
106                bit_count2 <= 0;
107                count_switch <= 0;
108            end
109            else if (cs == WRITE) begin
110                rx_data[9-bit_count] <= rx_temp;
111                bit_count <= bit_count + 1'b1;
112                    if (bit_count == 4'd9) begin
113                        rx_valid <= 1'b1;
114                    end else begin
115                        rx_valid <= 1'b0;
116                    end
117            end
118            else if(cs == READ_ADD) begin
119                read <= 1'b1;
120                rx_data[9-bit_count] <= rx_temp;
121                bit_count <= bit_count + 1'b1;
122                    if (bit_count == 4'd9) begin
123                        rx_valid <= 1'b1;
124                    end else begin
125                        rx_valid <= 1'b0;
126                    end
127            end
128            else if (cs == READ_DATA) begin
129                read <= 1'b0;
130                if (count_switch == 0 && bit_count < 4'd9) begin
131                    rx_data[9-bit_count] <= rx_temp;
132                    bit_count <= bit_count + 1'b1;
133                end
134                else if(bit_count == 4'd9)  begin
135                    rx_valid <= 1'b1;
136                    count_switch <=1'b1;
```

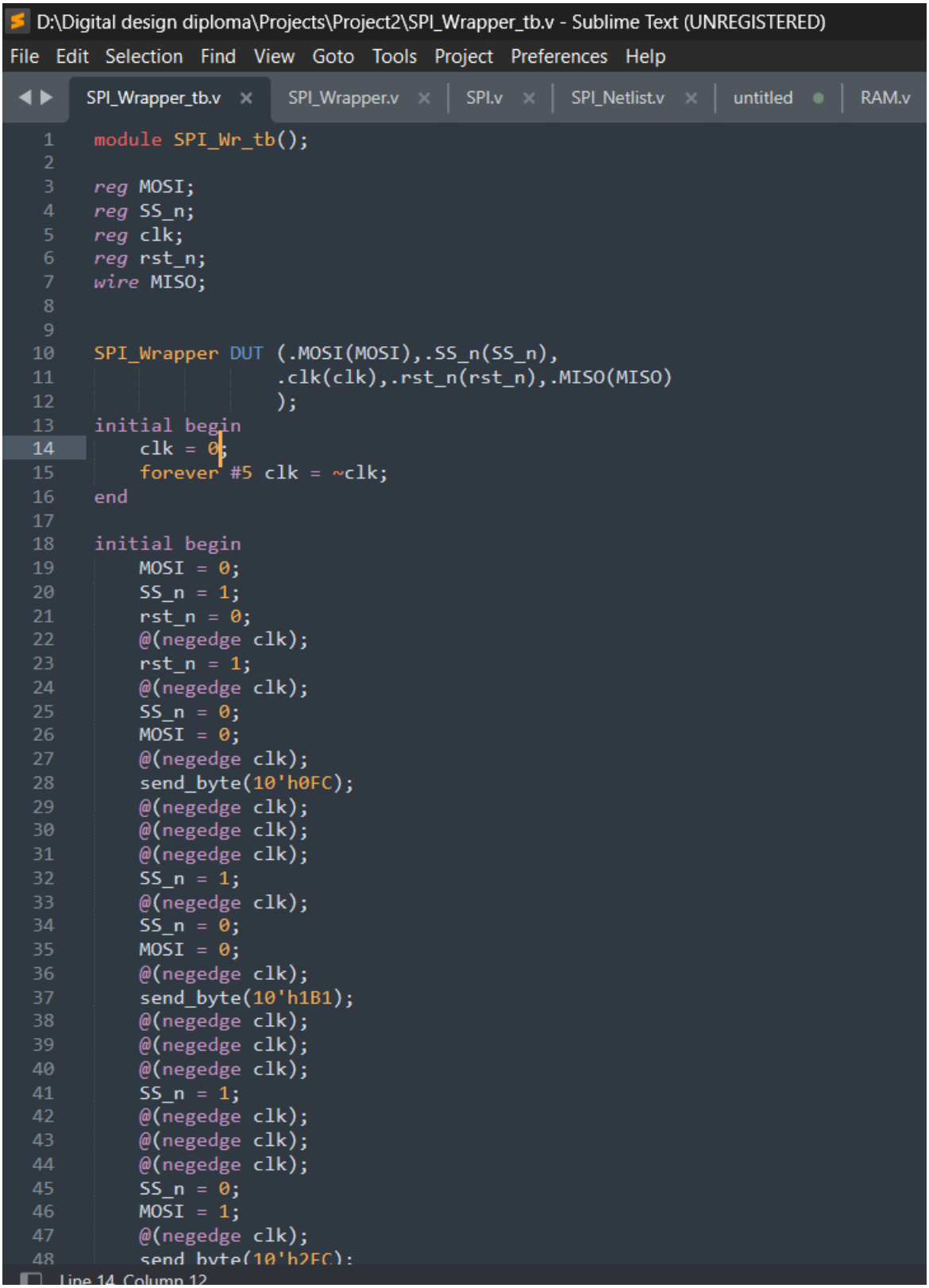
Line 16, Column 3

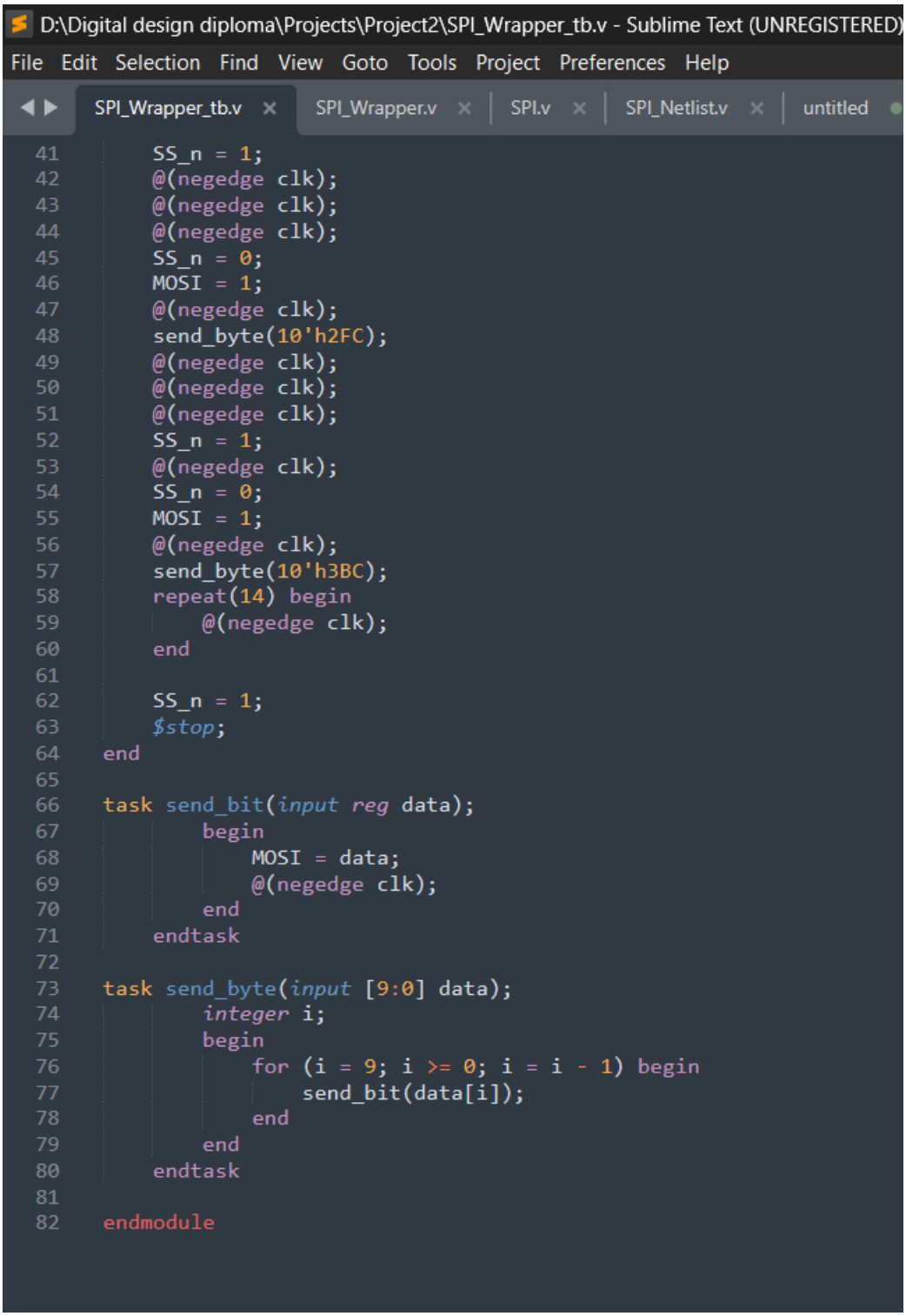

D:\Digital design diploma\Projects\Project2\SPI.v - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
SPI_Wrapper_tb.v
SPI_Wrapper.v
SPI.v
SPI_Netlist.v
untitled
RAM.v
Constraints_
125 rx_valid <= 1'b0;
126 end
127 end
128 else if (cs == READ_DATA) begin
129 read <= 1'b0;
130 if (count_switch == 0 && bit_count < 4'd9) begin
131 rx_data[9-bit_count] <= rx_temp;
132 bit_count <= bit_count + 1'b1;
133 end
134 else if(bit_count == 4'd9) begin
135 rx_valid <= 1'b1;
136 count_switch <=1'b1;
137 end
138 if (bit_count2 < 4'd8 && count_switch == 1 && tx_valid == 1) begin
139 bit_count <= 4'b0;
140 rx_valid <= 1'b0;
141 MISO <= tx_data[7-bit_count2];
142 bit_count2 <= bit_count2 + 1'b1;
143 end
144 else begin
145 MISO <= 1'b0;
146 end
147 end
148 end
149 end
150
151 always @(posedge clk) begin
152 rx_temp <= MOSI;
153 end
154
155 endmodule

Wrapper module:



Testbench:


D:\Digital design diploma\Projects\Project2\SPI_Wrapper_tb.v - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
SPI_Wrapper_tb.v
SPI_Wrapper.v
SPI.v
SPI_Netlist.v
untitled
RAM.v
module SPI_Wr_tb();
reg MOSI;
reg SS_n;
reg clk;
reg rst_n;
wire MISO;
SPI_Wrapper DUT (.MOSI(MOSI),.SS_n(SS_n),
.clk(clk),.rst_n(rst_n),.MISO(MISO)
);
initial begin
clk = 0;
forever #5 clk = ~clk;
end
initial begin
MOSI = 0;
SS_n = 1;
rst_n = 0;
@(negedge clk);
rst_n = 1;
@(negedge clk);
SS_n = 0;
MOSI = 0;
@(negedge clk);
send_byte(10'h0FC);
@(negedge clk);
@(negedge clk);
@(negedge clk);
SS_n = 1;
@(negedge clk);
SS_n = 0;
MOSI = 0;
@(negedge clk);
send_byte(10'h1B1);
@(negedge clk);
@(negedge clk);
@(negedge clk);
SS_n = 1;
@(negedge clk);
@(negedge clk);
@(negedge clk);
SS_n = 0;
MOSI = 1;
@(negedge clk);
send_byte(10'h2FC);
Line 14, Column 12



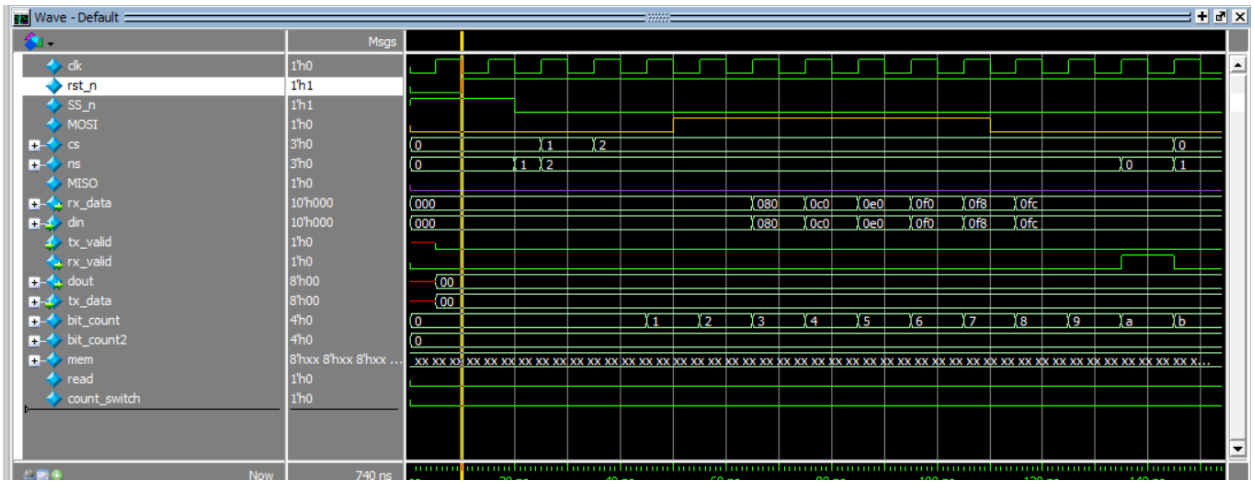
I won't lie I used chat GPT to make a TB for the SPI module alone and know how to make task.

But the overall module (SPI wrapper that connects SPI with RAM) I made myself but took the tasks.

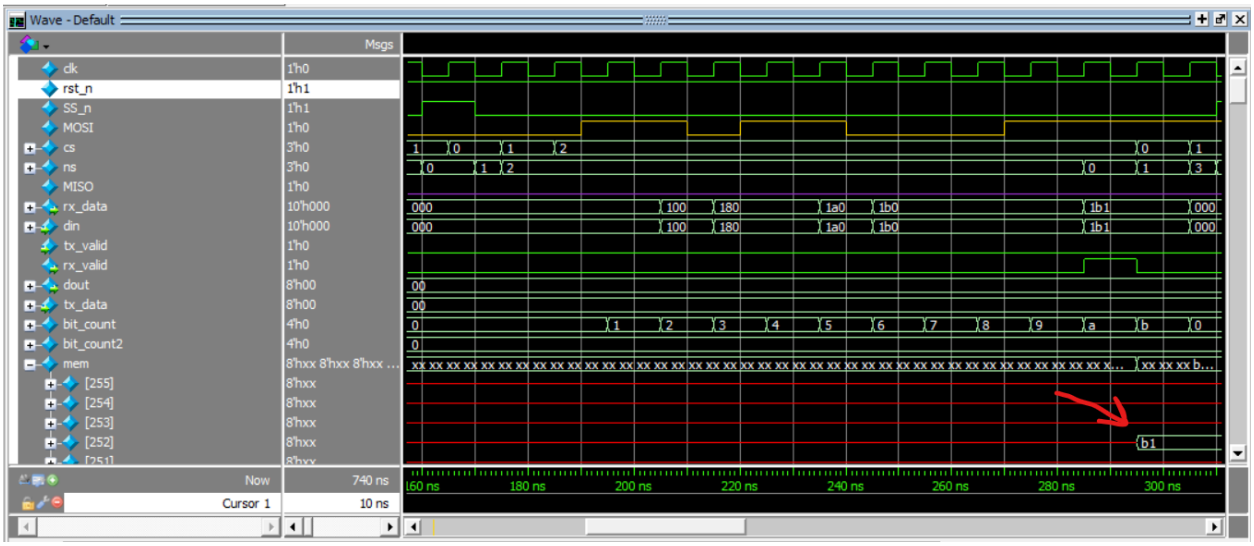
Absolutely all the design I made my self (it took more than 3 days) and it wasn't much helpful as its not trained a lot on the HDLs.

Waveform:

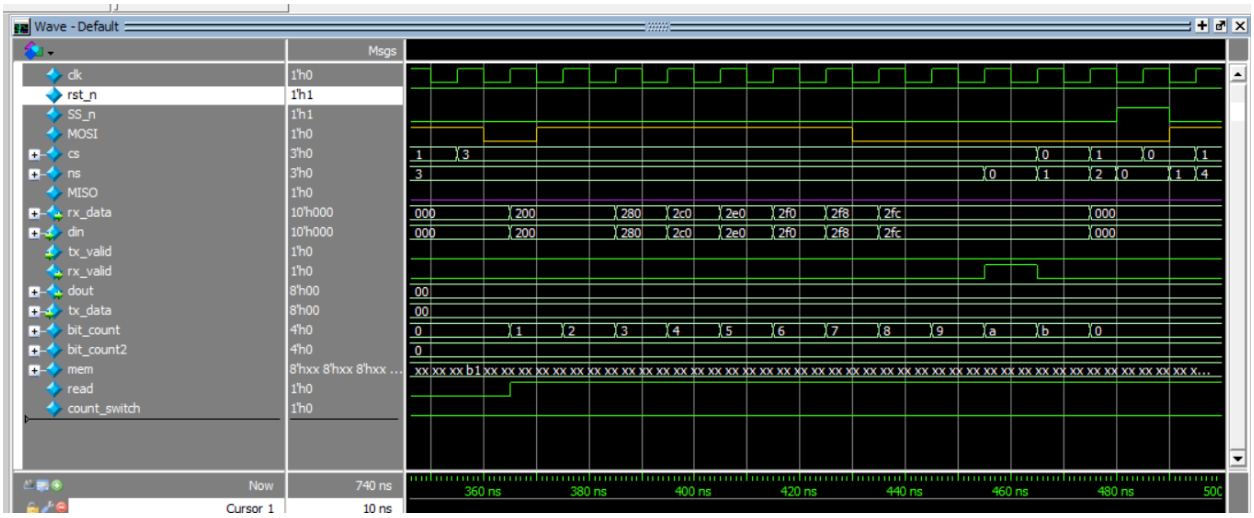
First state:



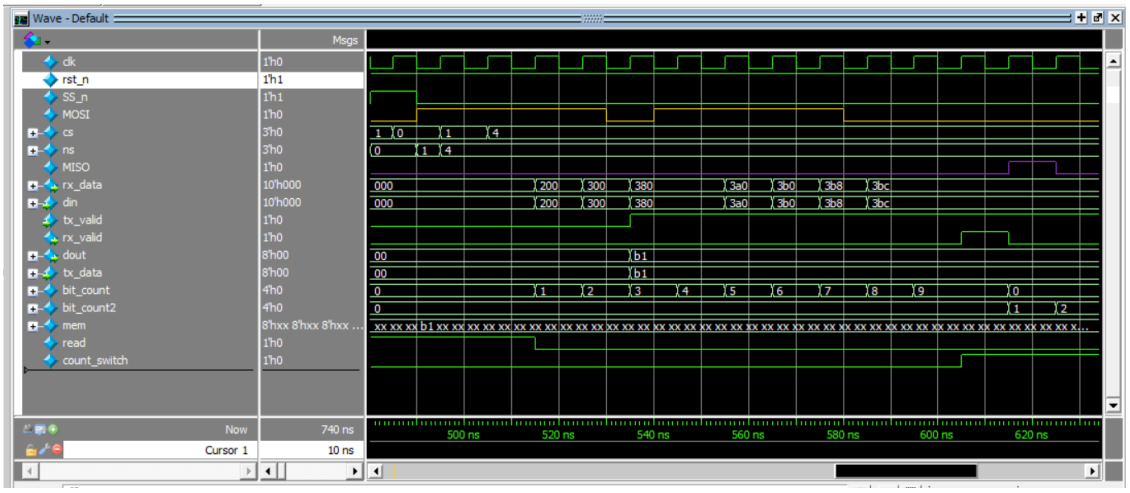
Second state:

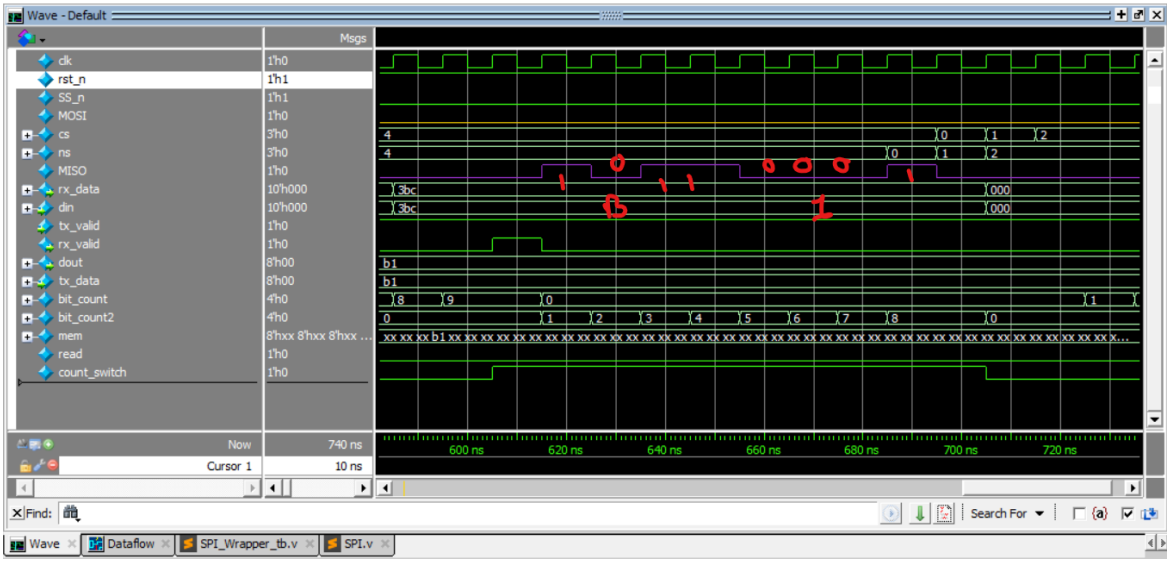


third state:



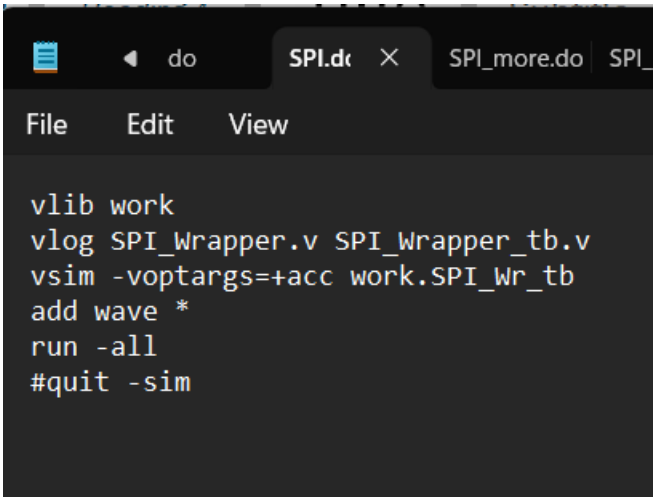
fourth state:





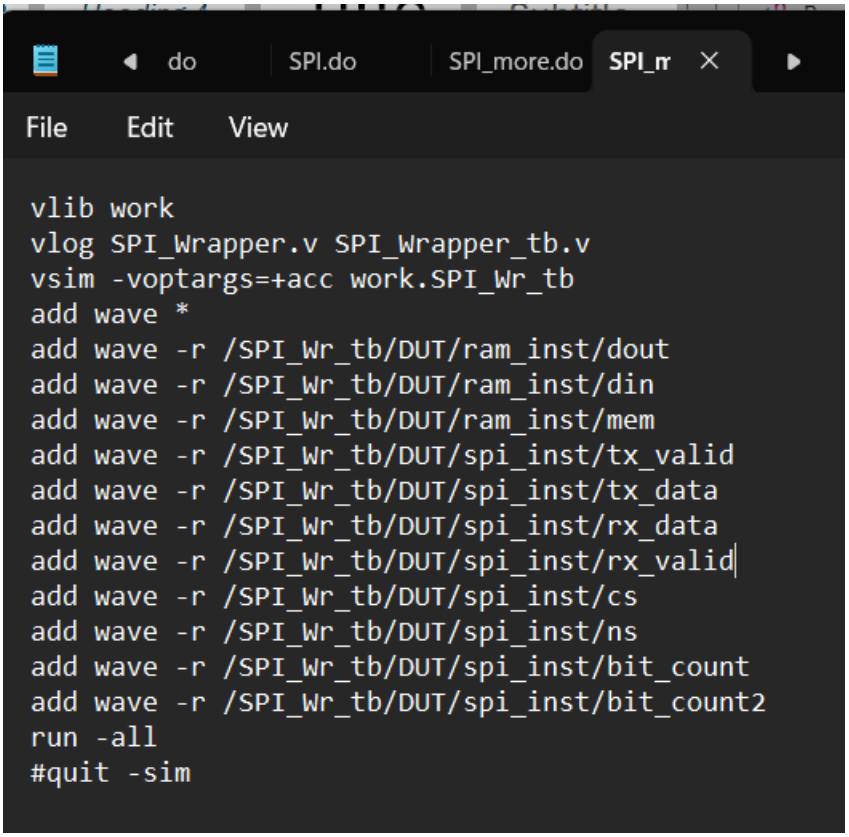
The do file:

The normal one:

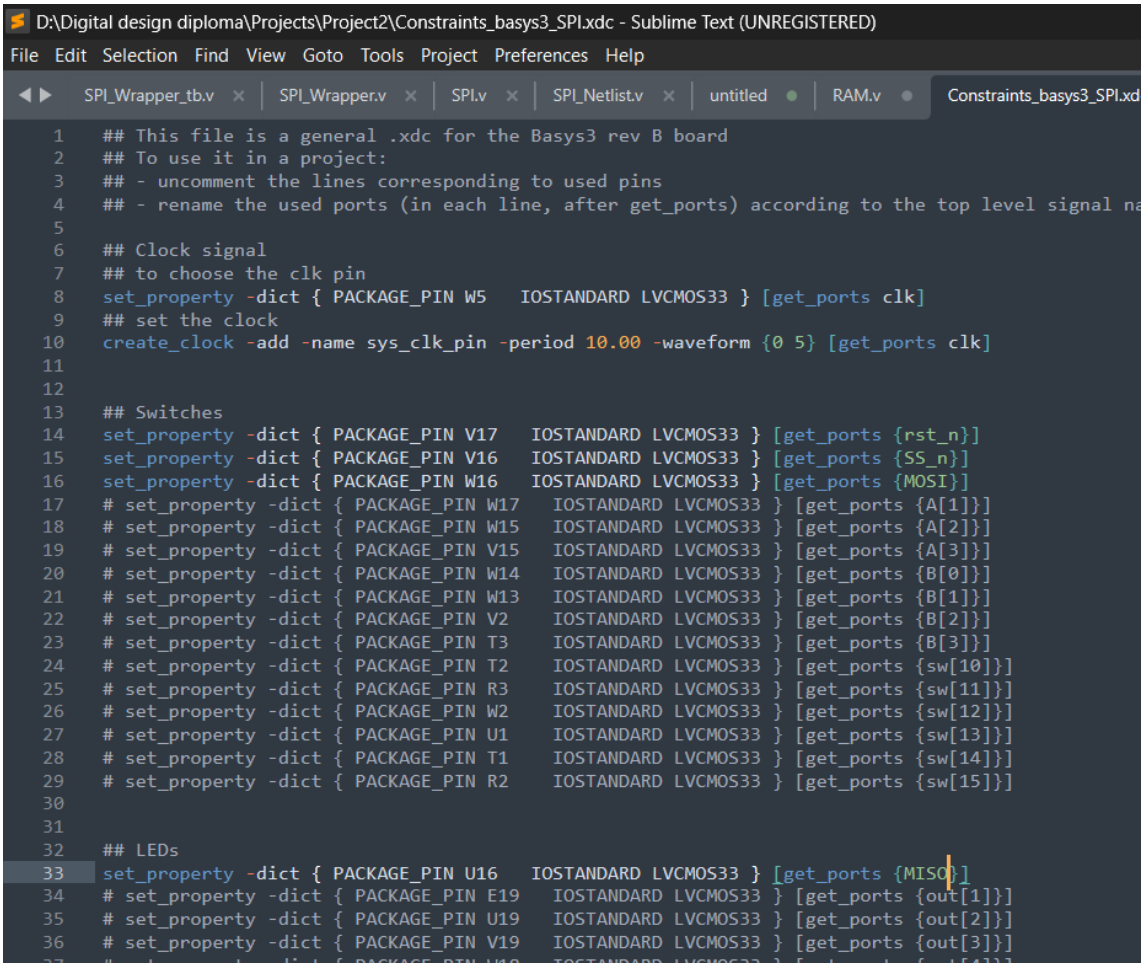


With more waves for details:

Its name SPI_more.do:



Constrain file:

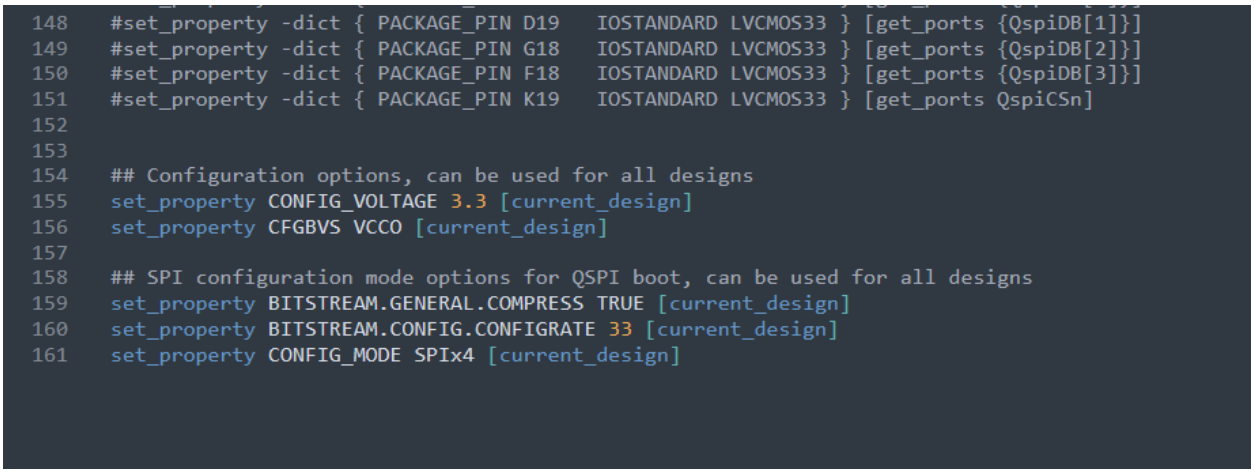


D:\Digital design diploma\Projects\Project2\Constraints_basys3_SPI.xdc - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

SPI_Wrapper_tb.v | SPI_Wrapper.v | SPI.v | SPI_Netlist.v | untitled | RAM.v | Constraints_basys3_SPI.xd

```
## This file is a general .xdc for the Basys3 rev B board
## To use it in a project:
## - uncomment the lines corresponding to used pins
## - rename the used ports (in each line, after get_ports) according to the top level signal na

## Clock signal
## to choose the clk pin
set_property -dict { PACKAGE_PIN W5   IOSTANDARD LVCMOS33 } [get_ports clk]
## set the clock
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]


## Switches
set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports {rst_n}]
set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports {SS_n}]
set_property -dict { PACKAGE_PIN W16   IOSTANDARD LVCMOS33 } [get_ports {MOSI}]
# set_property -dict { PACKAGE_PIN W17   IOSTANDARD LVCMOS33 } [get_ports {A[1]}]
# set_property -dict { PACKAGE_PIN W15   IOSTANDARD LVCMOS33 } [get_ports {A[2]}]
# set_property -dict { PACKAGE_PIN V15   IOSTANDARD LVCMOS33 } [get_ports {A[3]}]
# set_property -dict { PACKAGE_PIN W14   IOSTANDARD LVCMOS33 } [get_ports {B[0]}]
# set_property -dict { PACKAGE_PIN W13   IOSTANDARD LVCMOS33 } [get_ports {B[1]}]
# set_property -dict { PACKAGE_PIN V2    IOSTANDARD LVCMOS33 } [get_ports {B[2]}]
# set_property -dict { PACKAGE_PIN T3    IOSTANDARD LVCMOS33 } [get_ports {B[3]}]
# set_property -dict { PACKAGE_PIN T2    IOSTANDARD LVCMOS33 } [get_ports {sw[10]}]
# set_property -dict { PACKAGE_PIN R3    IOSTANDARD LVCMOS33 } [get_ports {sw[11]}]
# set_property -dict { PACKAGE_PIN W2    IOSTANDARD LVCMOS33 } [get_ports {sw[12]}]
# set_property -dict { PACKAGE_PIN U1    IOSTANDARD LVCMOS33 } [get_ports {sw[13]}]
# set_property -dict { PACKAGE_PIN T1    IOSTANDARD LVCMOS33 } [get_ports {sw[14]}]
# set_property -dict { PACKAGE_PIN R2    IOSTANDARD LVCMOS33 } [get_ports {sw[15]}]


## LEDs
set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports {MISO}]
# set_property -dict { PACKAGE_PIN E19   IOSTANDARD LVCMOS33 } [get_ports {out[1]}]
# set_property -dict { PACKAGE_PIN U19   IOSTANDARD LVCMOS33 } [get_ports {out[2]}]
# set_property -dict { PACKAGE_PIN V19   IOSTANDARD LVCMOS33 } [get_ports {out[3]}]
```

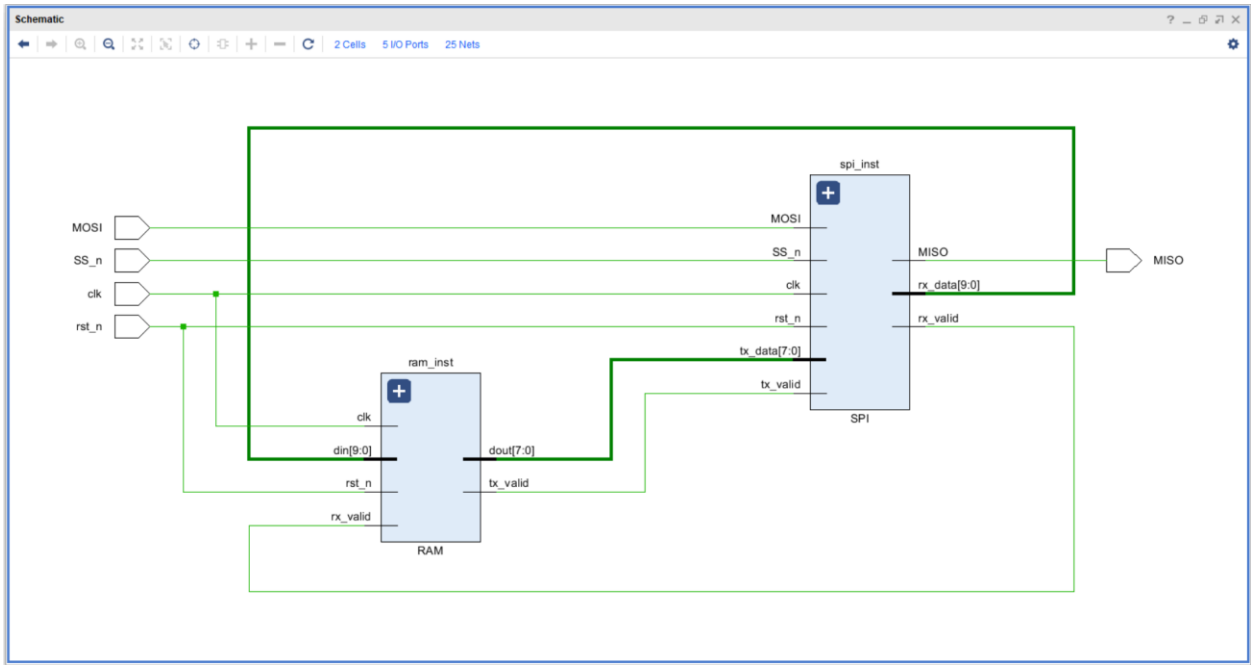


```
#set_property -dict { PACKAGE_PIN D19   IOSTANDARD LVCMOS33 } [get_ports {QspiDB[1]}]
#set_property -dict { PACKAGE_PIN G18   IOSTANDARD LVCMOS33 } [get_ports {QspiDB[2]}]
#set_property -dict { PACKAGE_PIN F18   IOSTANDARD LVCMOS33 } [get_ports {QspiDB[3]}]
#set_property -dict { PACKAGE_PIN K19   IOSTANDARD LVCMOS33 } [get_ports QspiCSn]


## Configuration options, can be used for all designs
set_property CONFIG_VOLTAGE 3.3 [current_design]
set_property CFGBVS VCCO [current_design]

## SPI configuration mode options for QSPI boot, can be used for all designs
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
set_property BITSTREAM.CONFIG.CONFIGRATE 33 [current_design]
set_property CONFIG_MODE SPIx4 [current_design]
```

Vivado:

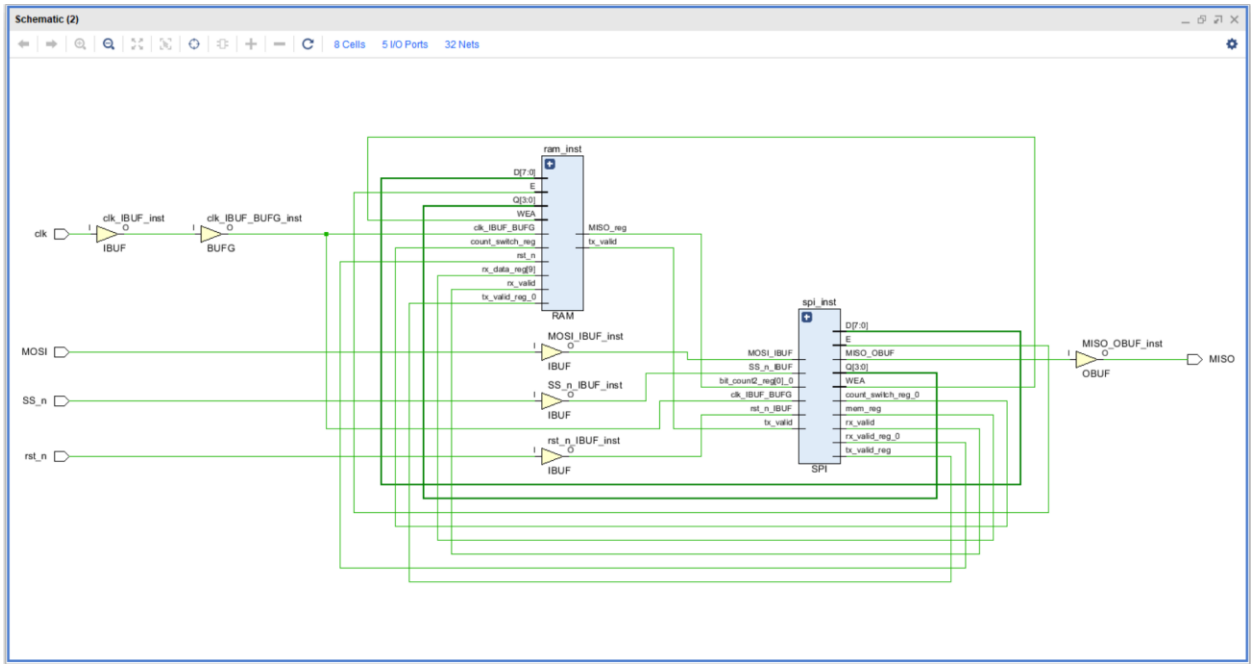
Sequential:

Elaboration design:



Synthesis:

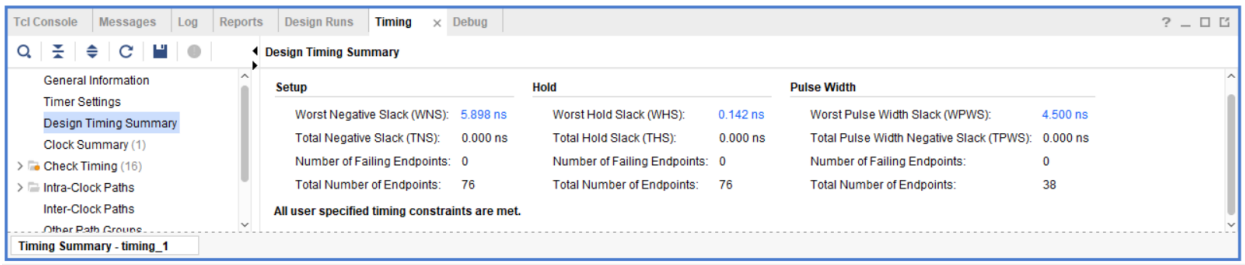
Schematic:



Report:

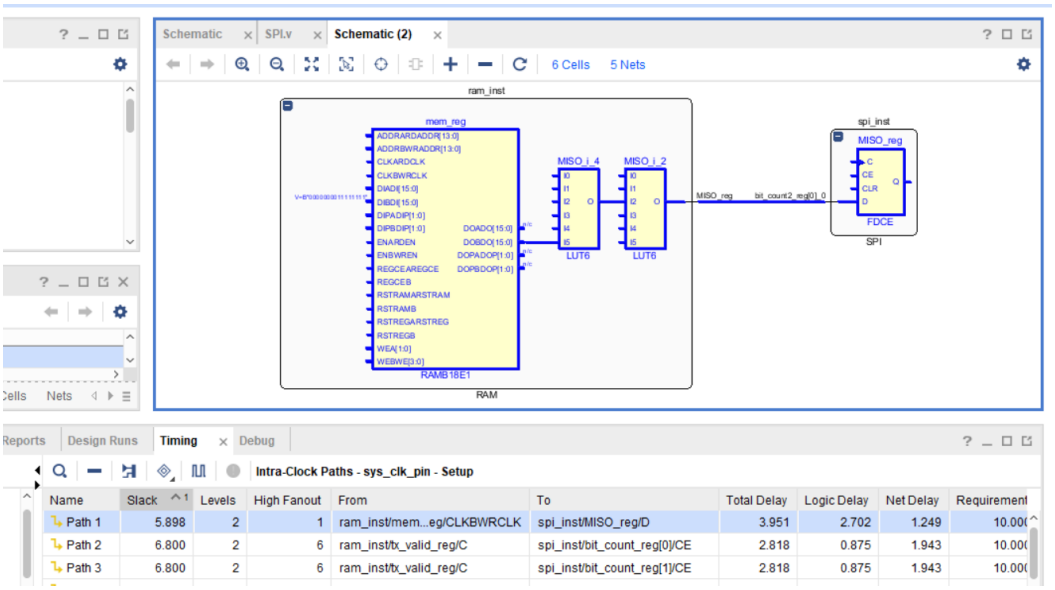
```
INFO: [Synth 8-6157] synthesizing module 'SPI' [D:/Digital design diploma/Projects/Project2/SPI.v:1]
    Parameter IDLE bound to: 3'b000
    Parameter CHK_CMD bound to: 3'b001
    Parameter WRITE bound to: 3'b010
    Parameter READ_ADD bound to: 3'b011
    Parameter READ_DATA bound to: 3'b100
INFO: [Synth 8-155] case statement is not full and has no default [D:/Digital design diploma/Projects/Project2/SPI.v:38
INFO: [Synth 8-6155] done synthesizing module 'SPI' (2#1) [D:/Digital design diploma/Projects/Project2/SPI.v:1]
INFO: [Synth 8-6155] done synthesizing module 'SPI_Wrapper' (3#1) [D:/Digital design diploma/Projects/Project2/SPI_Wrap
---------------------------------------------------------------------------------
Finished RTL Elaboration : Time (s): cpu = 00:00:02 ; elapsed = 00:00:02 . Memory (MB): peak = 431.047 ; gain = 152.555
```

Timing report:



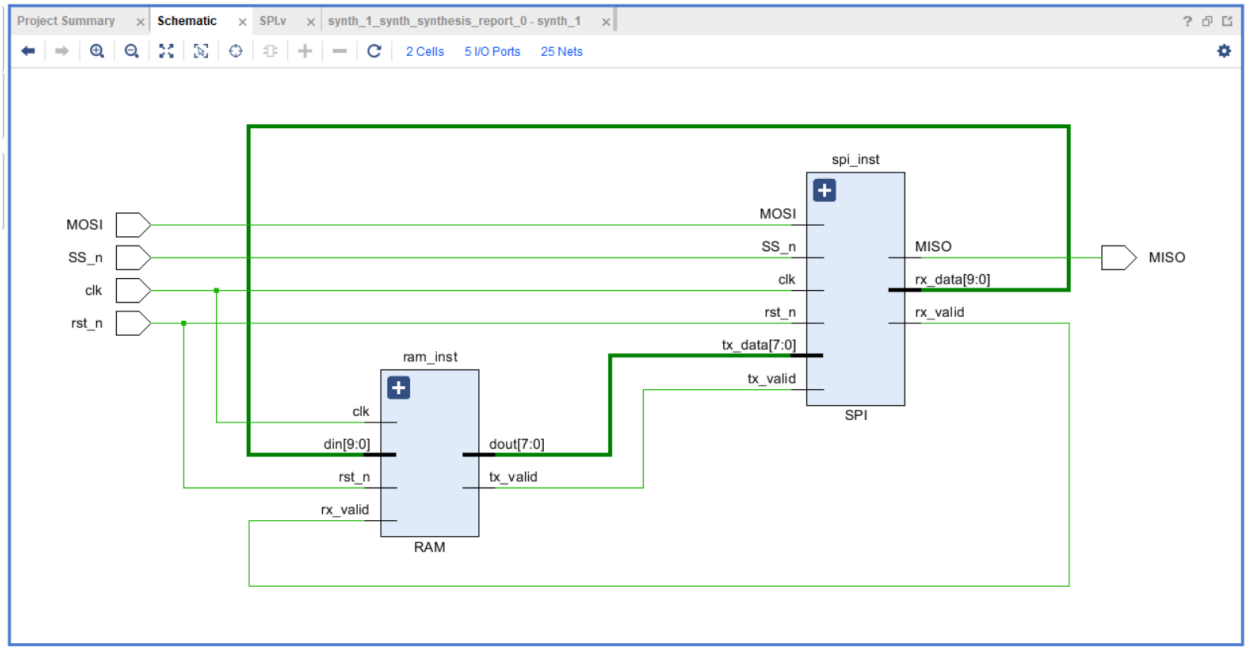
Messages:

Critical path:



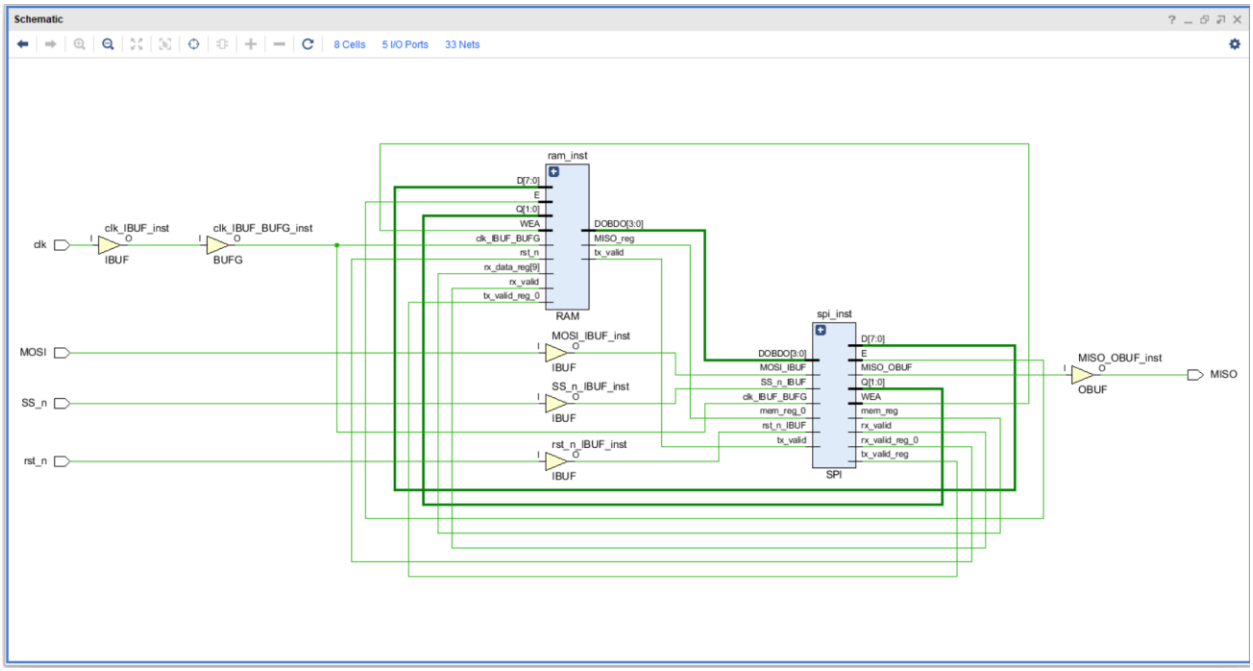
Gray encoding:

Elaboration design:

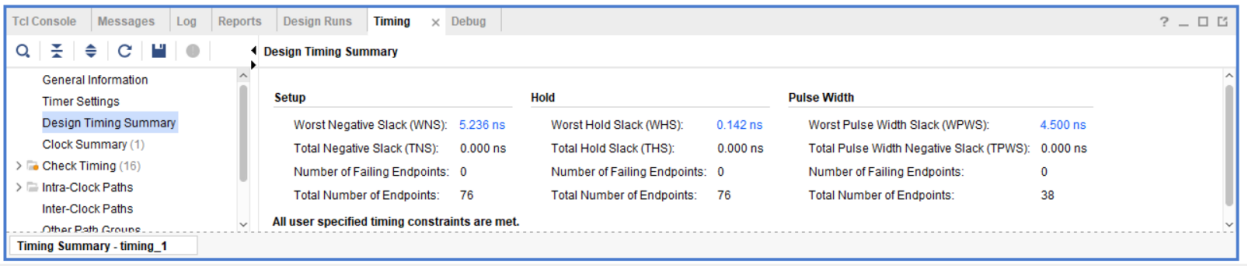


Synthesis:

Schematic:



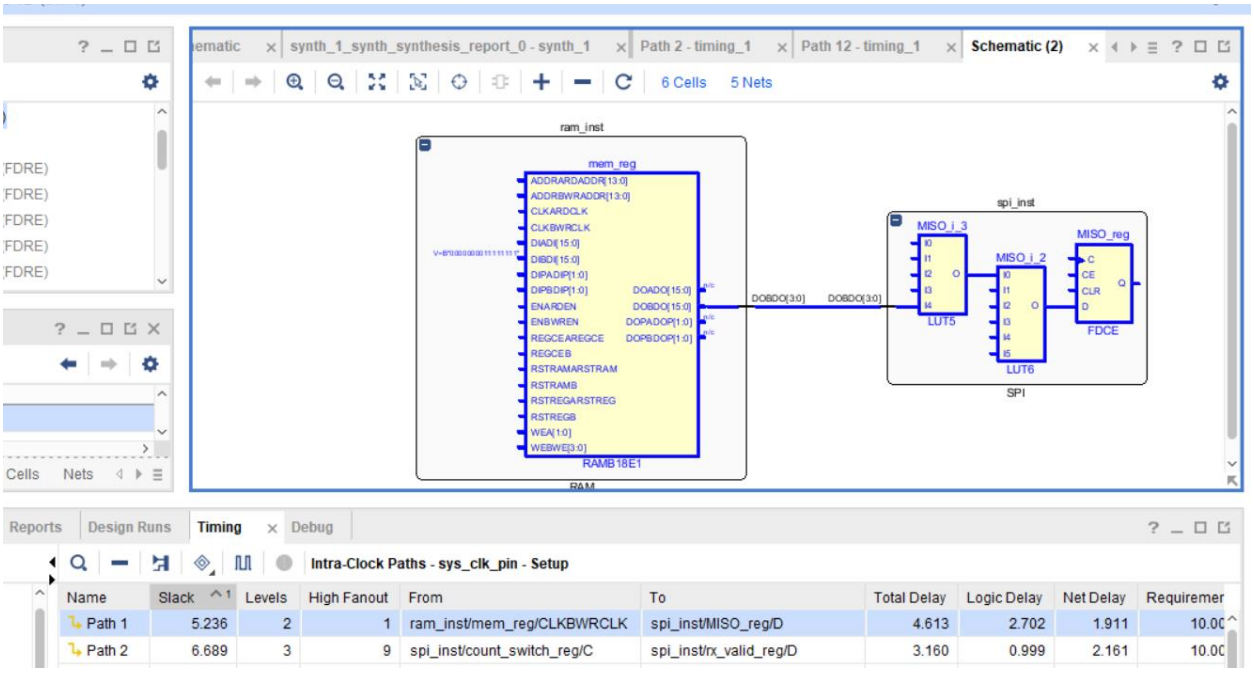
Timing report:



Report:

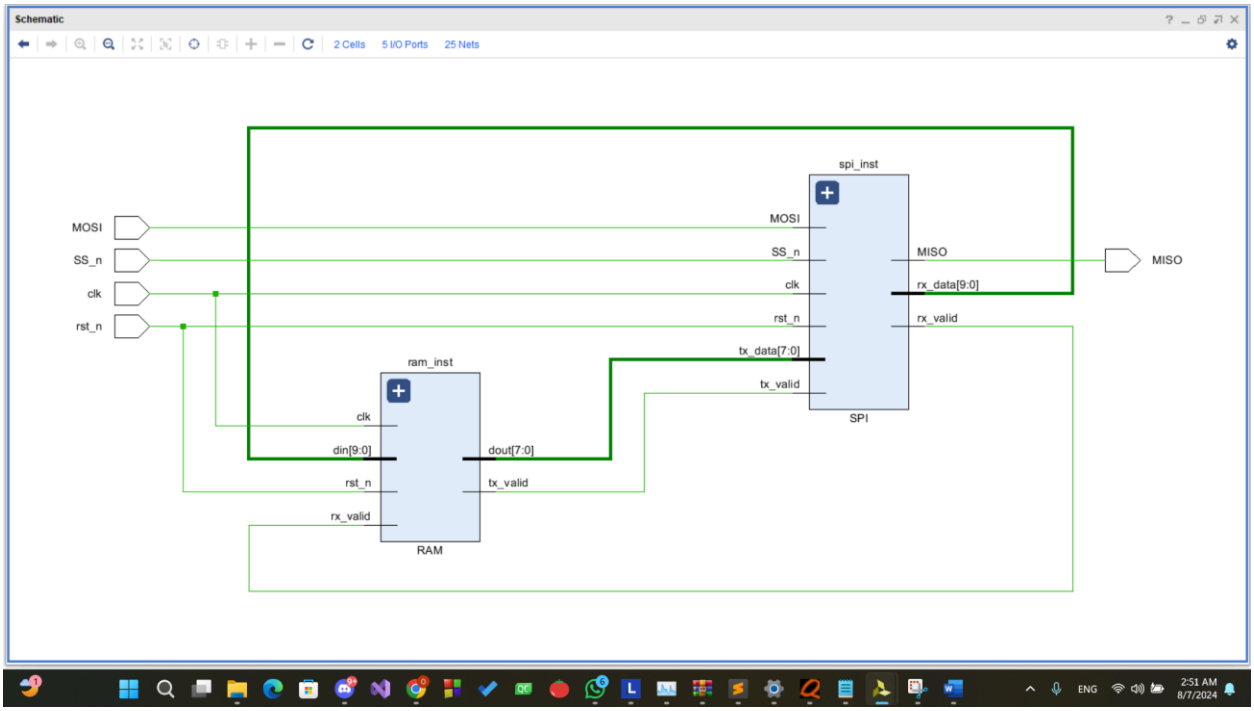
```
INFO: [Synth 8-6157] synthesizing module 'SPI' [D:/Digital design diploma/Projects/Project2/SPI.v:1]
    Parameter IDLE bound to: 3'b000
    Parameter CHK_CMD bound to: 3'b001
    Parameter WRITE bound to: 3'b010
    Parameter READ_ADD bound to: 3'b011
    Parameter READ_DATA bound to: 3'b100
INFO: [Synth 8-5534] Detected attribute (* fsm_encoding = "gray" *) [D:/Digital design diploma/Projects/Project2/SPI.v:
INFO: [Synth 8-155] case statement is not full and has no default [D:/Digital design diploma/Projects/Project2/SPI.v:38
```

Critical path:



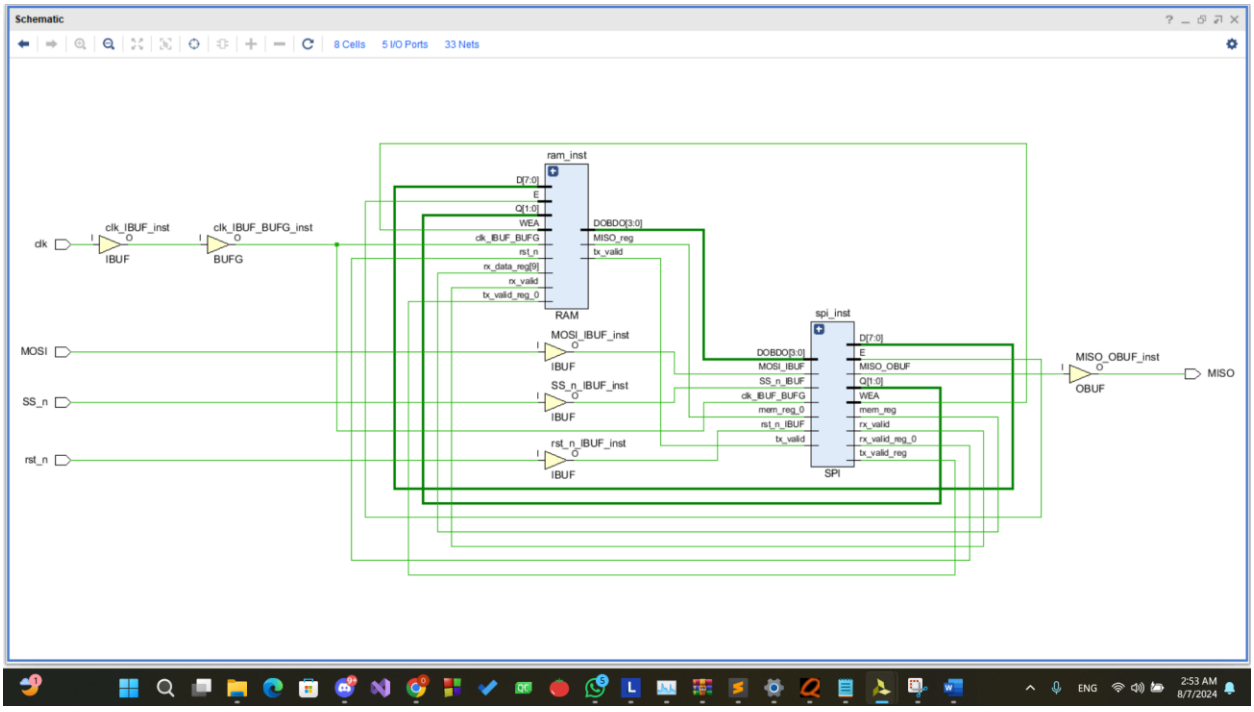
One Hot encoding:

Elaboration design:

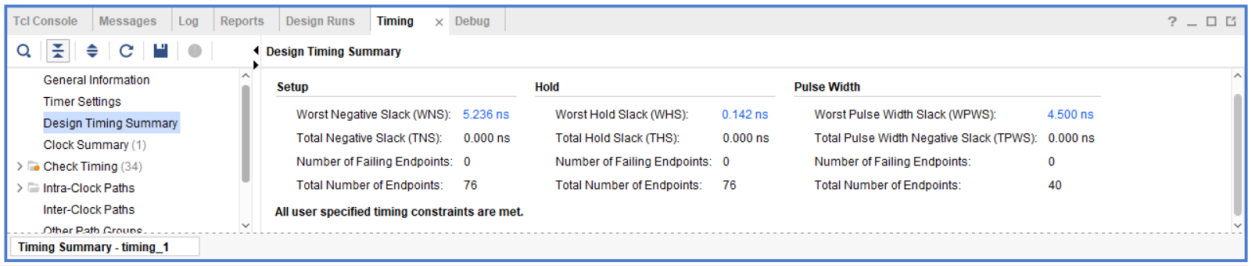


Synthesis:

Schematic:



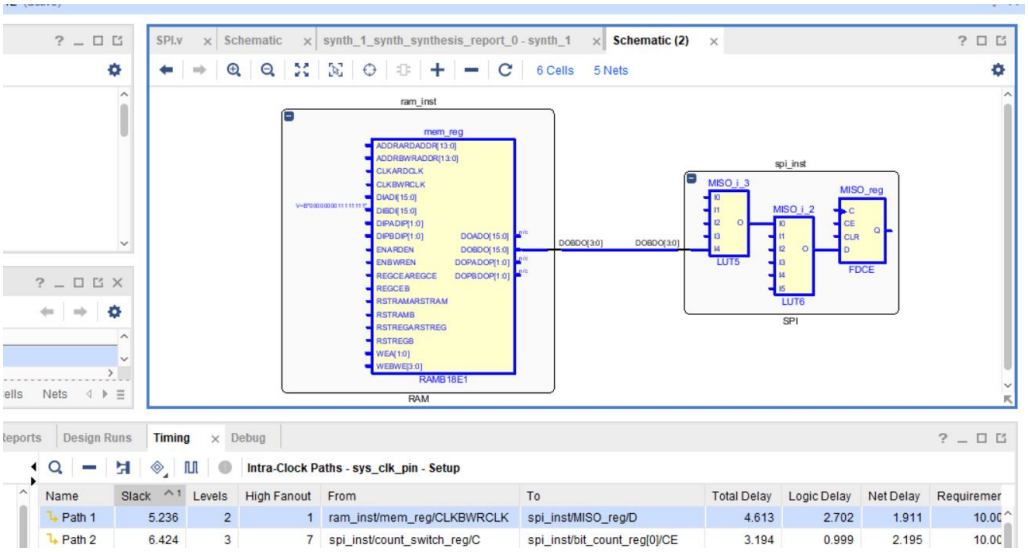
Timing report:



Report:

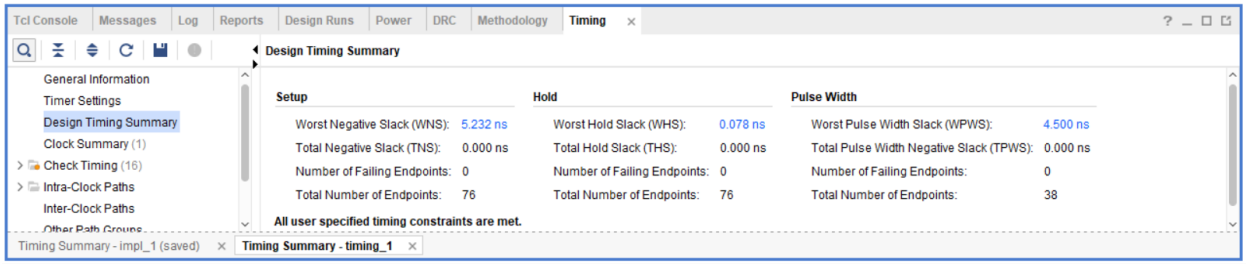
```
INFO: [Synth 8-226] default block is never used [D:/Digital design diploma/Projects/Project2/RAM.v:21]
INFO: [Synth 8-6155] done synthesizing module 'RAM' (1#1) [D:/Digital design diploma/Projects/Project2/RAM.v:1]
INFO: [Synth 8-6157] synthesizing module 'SPI' [D:/Digital design diploma/Projects/Project2/SPI.v:1]
    Parameter IDLE bound to: 3'b000
    Parameter CHK_CMD bound to: 3'b001
    Parameter WRITE bound to: 3'b010
    Parameter READ_ADD bound to: 3'b011
    Parameter READ_DATA bound to: 3'b100
INFO: [Synth 8-5534] Detected attribute (* fsm_encoding = "one_hot" *) [D:/Digital design diploma/Projects/Project2/SPI
INFO: [Synth 8-155] case statement is not full and has no default [D:/Digital design diploma/Projects/Project2/SPI.v:38
```

Critical path:

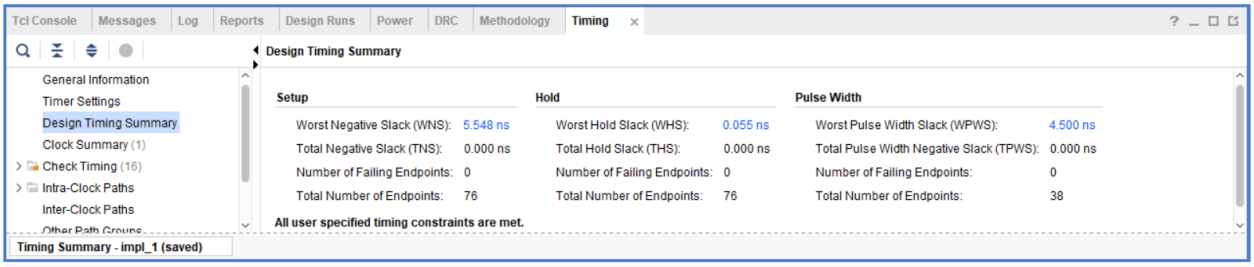


After implementation I compared between the 3 encoding types:

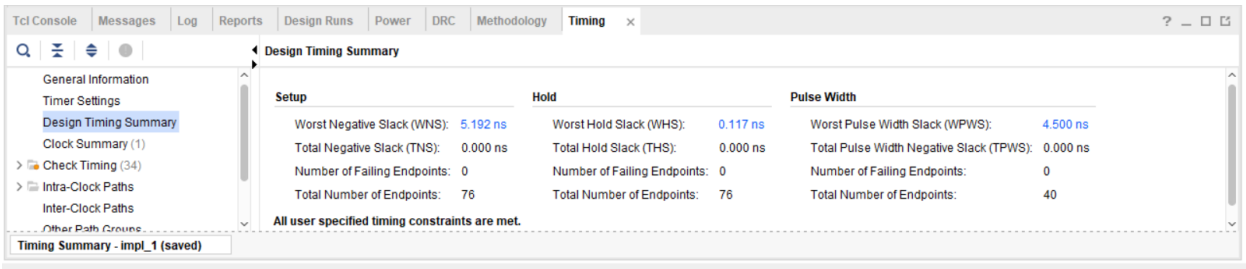
For gray encoding:



For seq encoding:



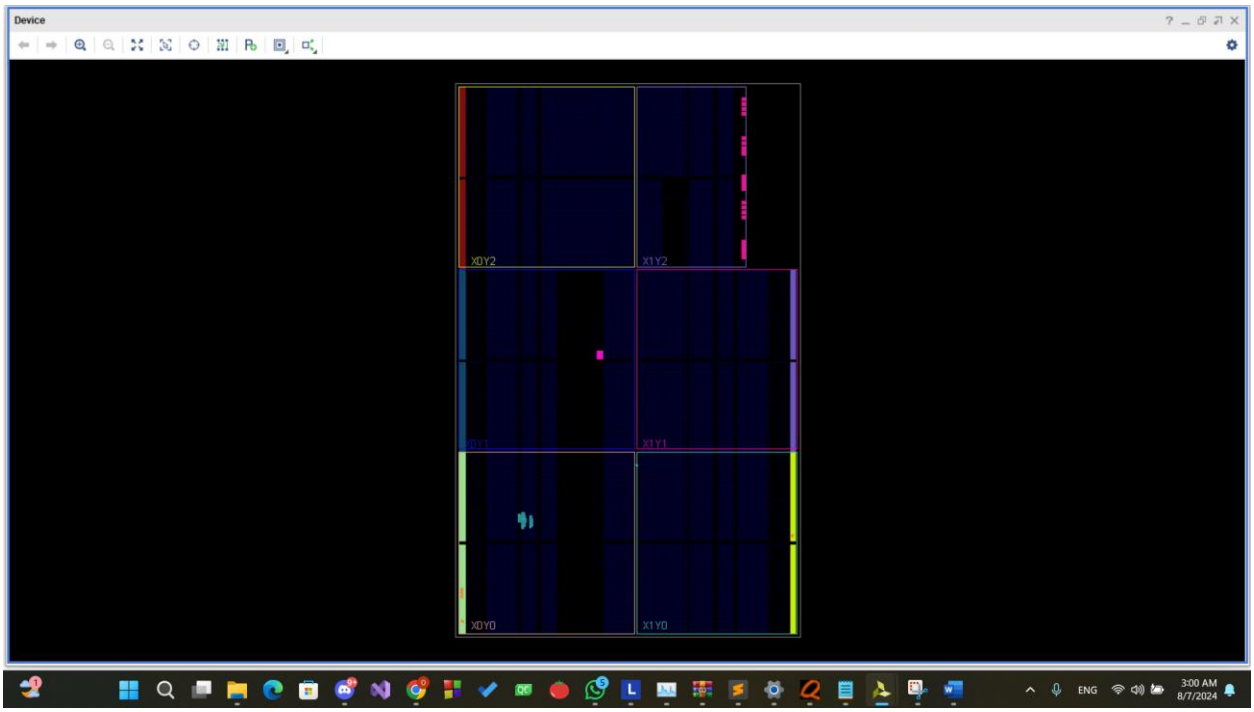
For one_hot encoding:



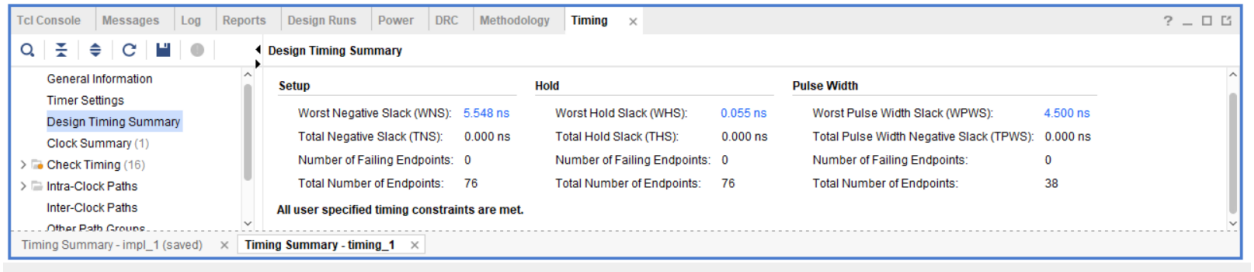
The sequential showed best setup time slack so, I completed the implementation using sequential.

Implementation:

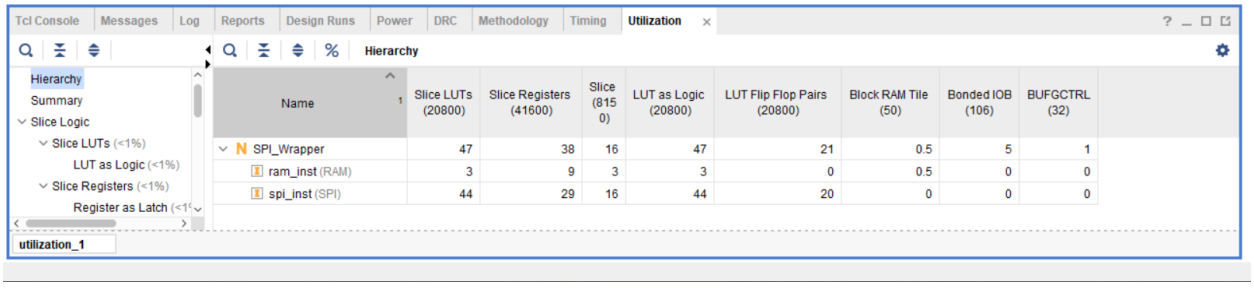
Device:



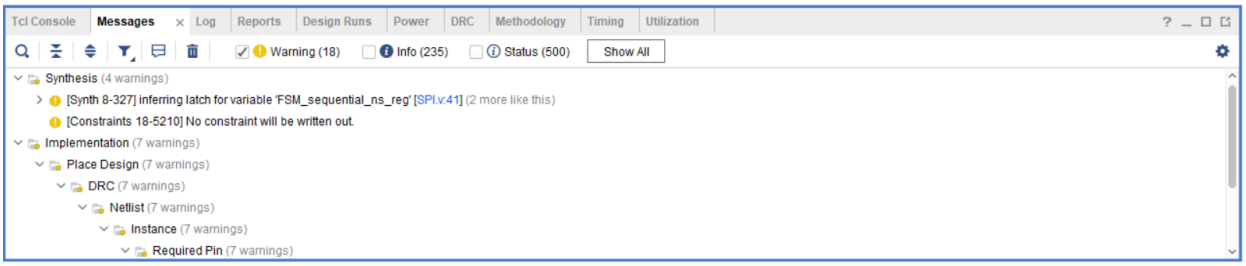
Timing report:



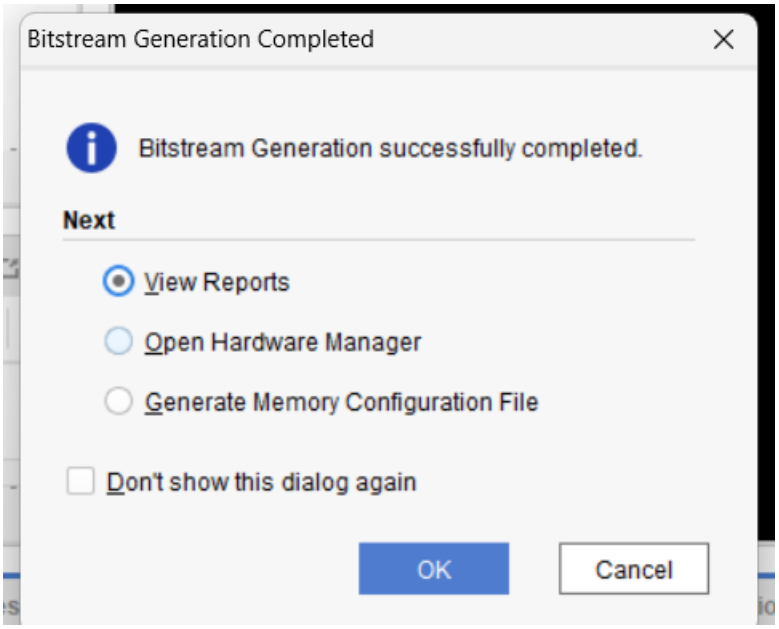
Utilization report:



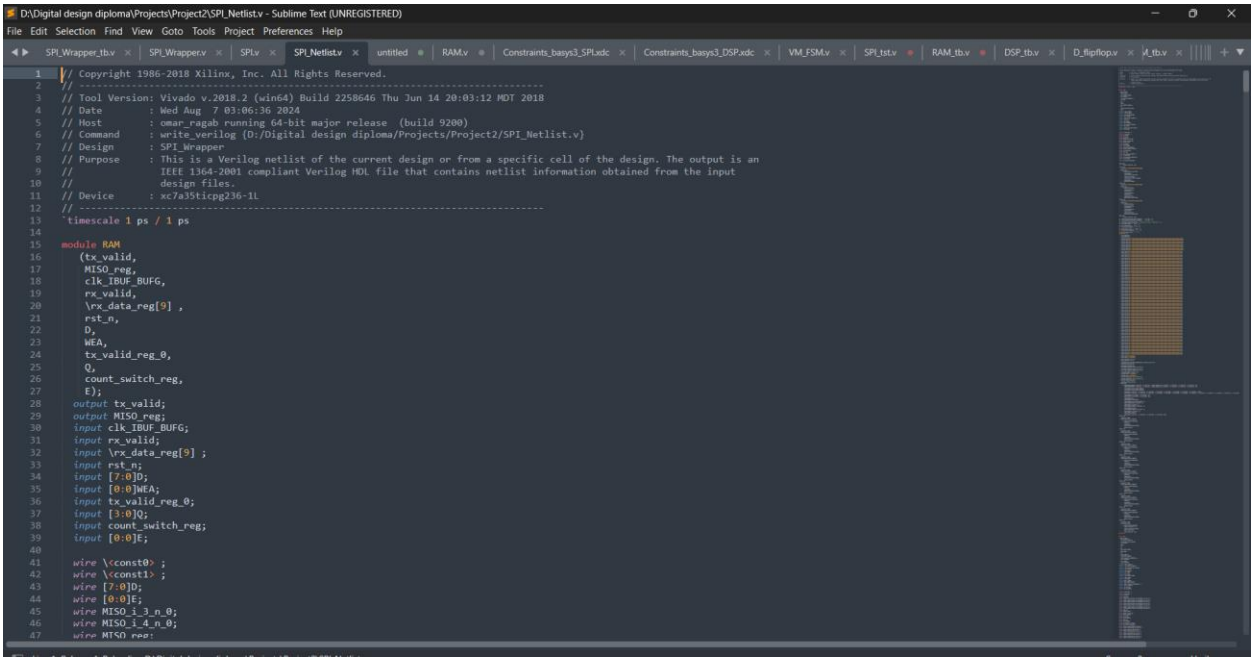
Messages:



Bitstream:



Netlist:


D:\Digital design diploma\Projects\Project2\SPI_Netlist.v - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
SPI_Wrapper_tb.v
SPI_Wrapper.v
SPI.v
SPI_Netlist.v
untitled
RAM.v
Constraints_basys3_SPI.xdc
Constraints_basys3_DSP.xdc
VM_FSM.v
SPI_tst.v
RAM_tb.v
DSP_tb.v
D_flipflop.v
// Copyright 1986-2018 Xilinx, Inc. All Rights Reserved.
// Tool Version: Vivado v.2018.2 (win64) Build 2258646 Thu Jun 14 20:03:12 MDT 2018
// Date : Wed Aug 7 03:06:36 2024
// Host : omar_ragab running 64-bit major release (build 9200)
// Command : write_verilog {D:/Digital design diploma/Projects/Project2/SPI_Netlist.v}
// Design : SPI_Wrapper
// Purpose : This is a Verilog netlist of the current design or from a specific cell of the design. The output is an
// IEEE 1364-2001 compliant Verilog HDL file that contains netlist information obtained from the input
// design files.
// Device : xc7a35ticpg236-1L
`timescale 1 ps / 1 ps
module RAM
(tx_valid,
MISO_reg,
clk_IBUF_BUFG,
rx_valid,
\rx_data_reg[9] ,
rst_n,
D,
WEA,
tx_valid_reg_0,
Q,
count_switch_reg,
E);
output tx_valid;
output MISO_reg;
input clk_IBUF_BUFG;
input rx_valid;
input \rx_data_reg[9] ;
input rst_n;
input [7:0]D;
input [0:0]WEA;
input tx_valid_reg_0;
input [3:0]Q;
input count_switch_reg;
input [0:0]E;
wire \<const0> ;
wire \<const1> ;
wire [7:0]D;
wire [0:0]E;
wire MISO_i_3_n_0;
wire MISO_i_4_n_0;