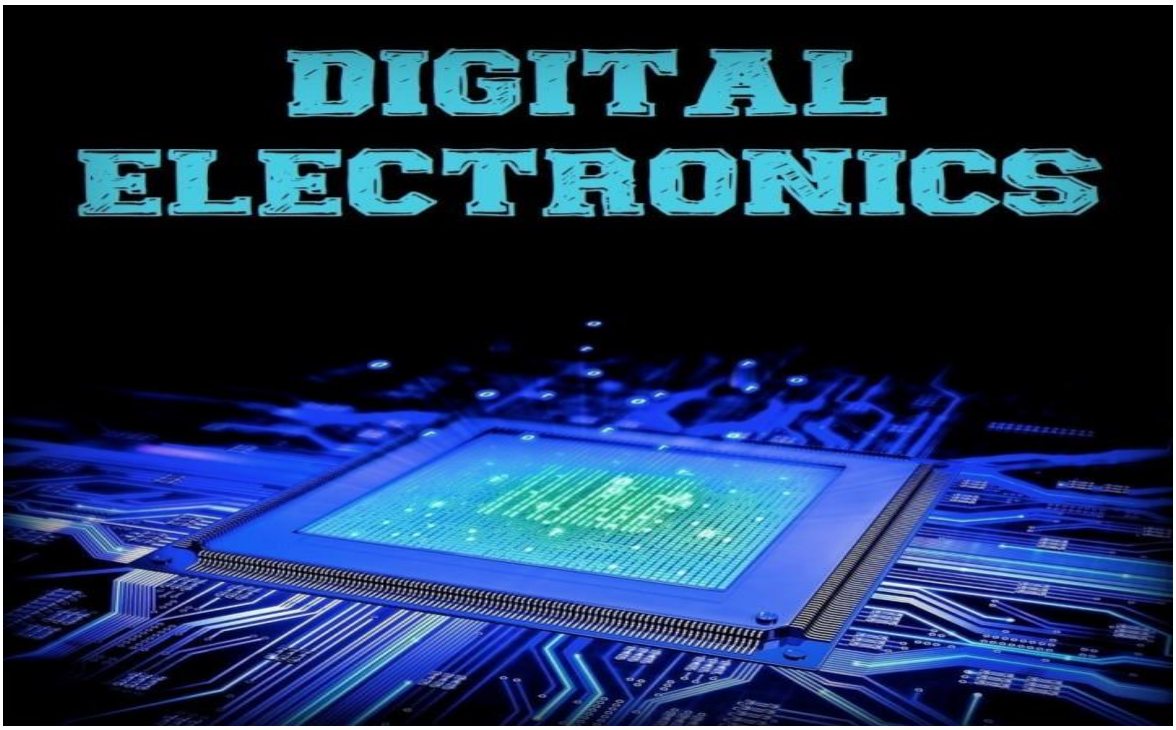
Abanob Evram

SPI-Slave with Single Port RAM



1-The Implementation (Timing report) for seq encoding:

```
-----------------------------------------------------------------------------------------------------
                 State |                     New Encoding |                Previous Encoding
-----------------------------------------------------------------------------------------------------
                  IDLE |                              000 |                             000
                CHK_CMD |                              001 |                             001
                 WRITE |                              010 |                             010
              READ_ADD |                              011 |                             011
             READ_DATA |                              100 |                             100
-----------------------------------------------------------------------------------------------------
INFO: [Synth 8-3354] encoded FSM with state register 'cs_reg' using encoding 'sequential' in module 'SPI'
-----------------------------------------------------------------------------
```

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	4.795 ns	Worst Hold Slack (WHS):	0.102 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	205	Total Number of Endpoints:	205	Total Number of Endpoints:	99

All user specified timing constraints are met.

2-The Implementation (Timing report) for one_hot encoding:

```
-----------------------------------------------------------------------------------------------------
                 State |                     New Encoding |                Previous Encoding
-----------------------------------------------------------------------------------------------------
                  IDLE |                            00001 |                             000
                CHK_CMD |                            00010 |                             001
                 WRITE |                            00100 |                             010
              READ_ADD |                            01000 |                             011
             READ_DATA |                            10000 |                             100
-----------------------------------------------------------------------------------------------------
INFO: [Synth 8-3354] encoded FSM with state register 'cs_reg' using encoding 'one-hot' in module 'SPI'
```

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	4.042 ns	Worst Hold Slack (WHS):	0.052 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	207	Total Number of Endpoints:	207	Total Number of Endpoints:	101

All user specified timing constraints are met.

3-The Implementation (Timing report) for gray encoding:

State	New Encoding	Previous Encoding
IDLE	000	000
CHK_CMD	001	001
WRITE	011	010
READ_ADD	010	011
READ_DATA	111	100

```
INFO: [Synth 8-3354] encoded FSM with state register 'cs_reg' using encoding 'gray' in module 'SPI'
```

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	4.861 ns	Worst Hold Slack (WHS):	0.075 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	205	Total Number of Endpoints:	205	Total Number of Endpoints:	99

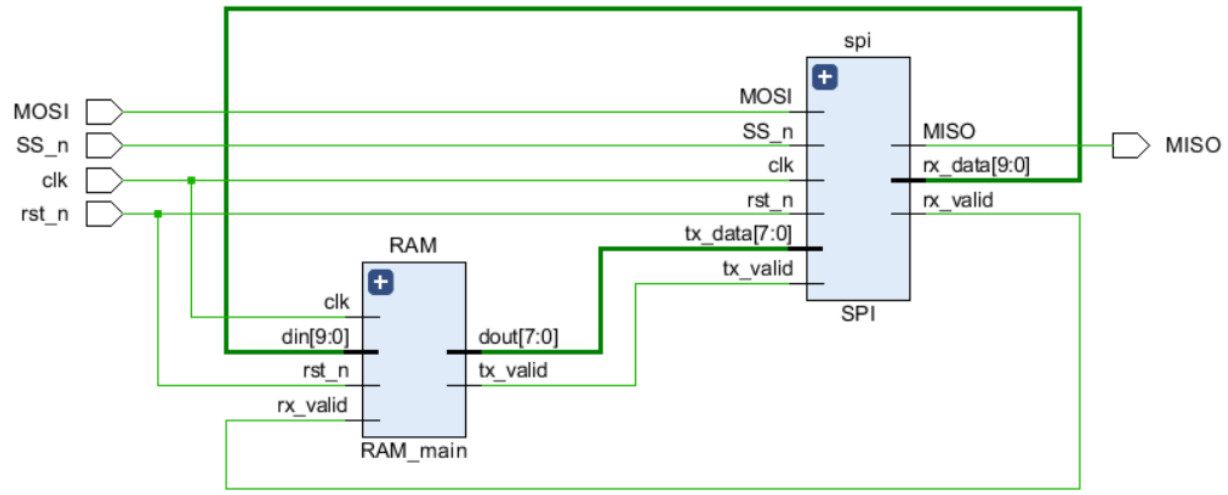
All user specified timing constraints are met.

Then the best encoding is hot_encoding

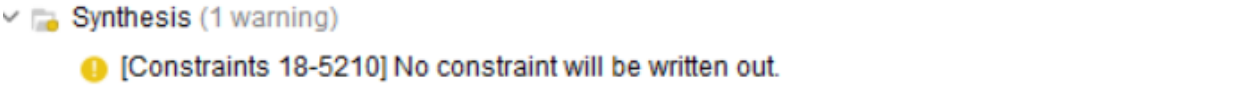
Because it's the best time

1-The elaboration (message) for one_hot encoding:

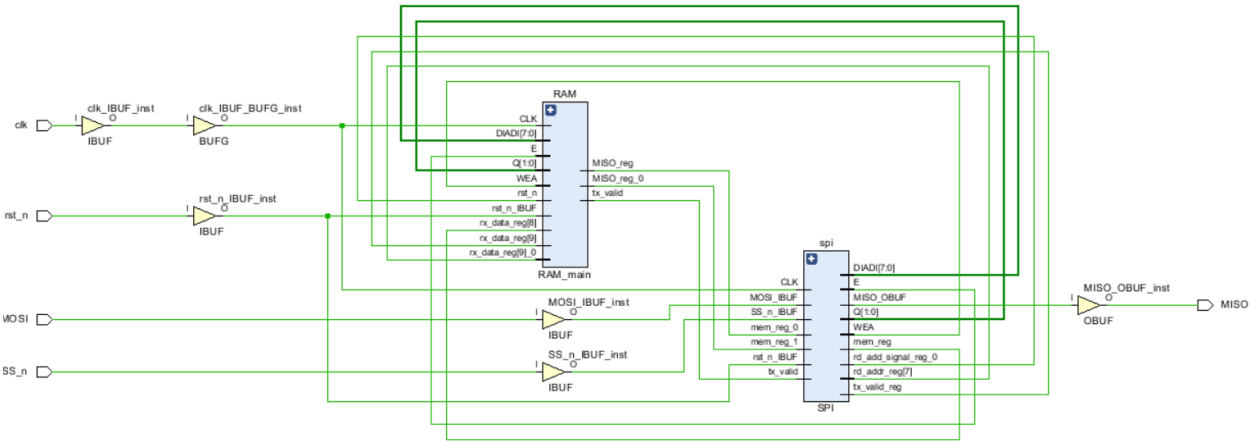
2-The elaboration (schematic) for one_hot encoding:



3-The synthesis (message) for one_hot encoding:



4-The synthesis (schematic) for one_hot encoding:



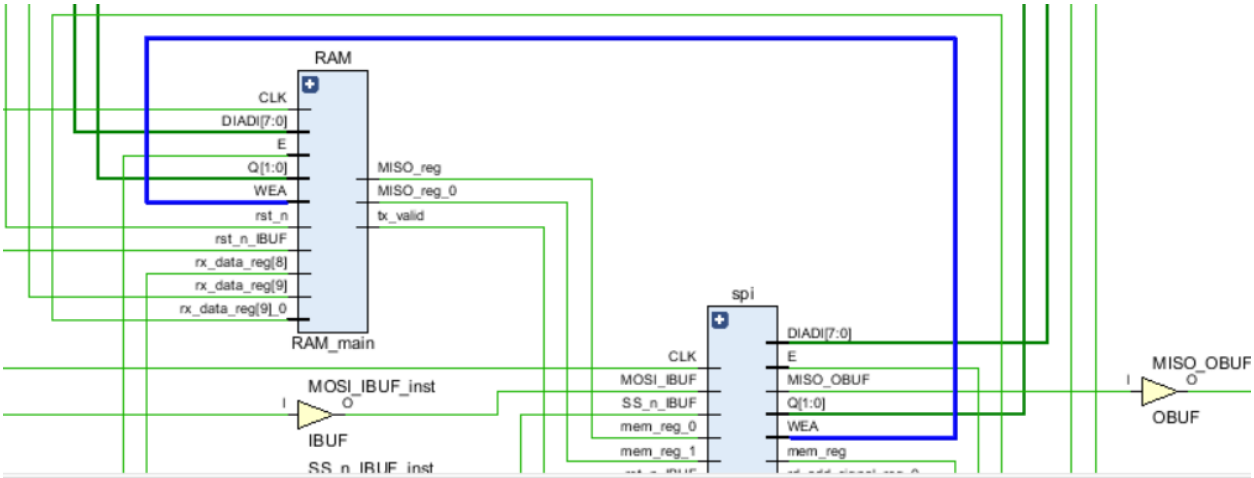
5-The synthesis (report) for one_hot encoding:

State	New Encoding	Previous Encoding
IDLE	00001	000
CHK_CMD	00010	001
WRITE	00100	010
READ_ADD	01000	011
READ_DATA	10000	100

6-The synthesis (Timing report) for one_hot encoding:

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	5.194 ns	Worst Hold Slack (WHS):	0.149 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	206	Total Number of Endpoints:	206	Total Number of Endpoints:	101

7-The synthesis (critical path) for one_hot encoding:



Intra-Clock Paths - sys_clk_pin - Setup

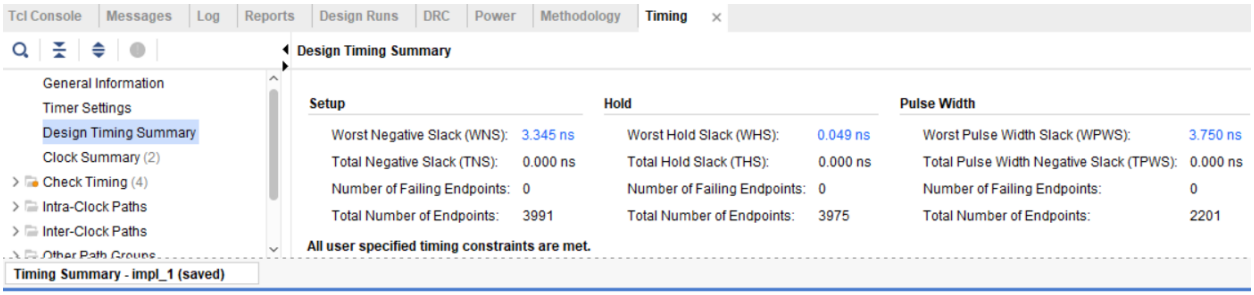
Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net De
Path 1	5.194	6	30	spi/count_rx_reg[3]/C	RAM/mem_reg/WEA[0]	4.094	1.777	2.

8-The Implementation (message) for one_hot encoding:

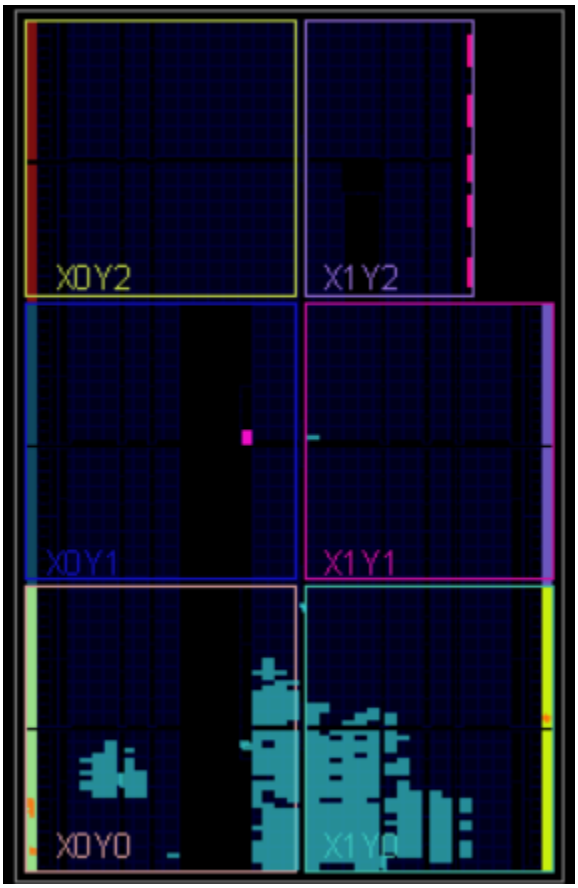
9-The Implementation (Utilization report) for one_hot encoding:

Name	Slice LUTs (20800)	Slice Registers (41600)	F7 Muxes (16300)	Slice (8150)	LUT as Logic (20800)	LUT as Memory (9600)	LUT Flip Flop Pairs (20800)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)	BSCANE2 (4)
Maindesign	1415	2014	10	676	1307	108	808	1	5	2	1
dbg_hub (dbg_hub)	475	727	0	243	451	24	306	0	0	1	1
RAM (RAM_main)	2	17	0	6	2	0	0	0.5	0	0	0
spi (SPI)	174	81	0	57	174	0	81	0	0	0	0
u_ila_0 (u_ila_0)	764	1189	10	377	680	84	420	0.5	0	0	0

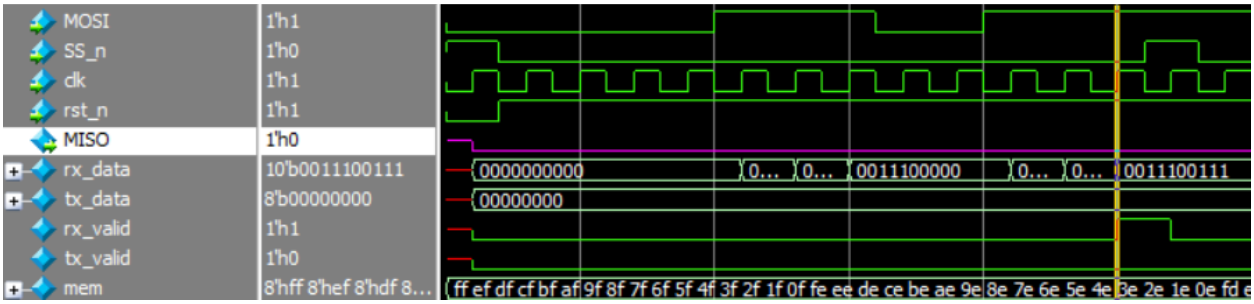
10-The Implementation (Timing report) for one_hot encoding:

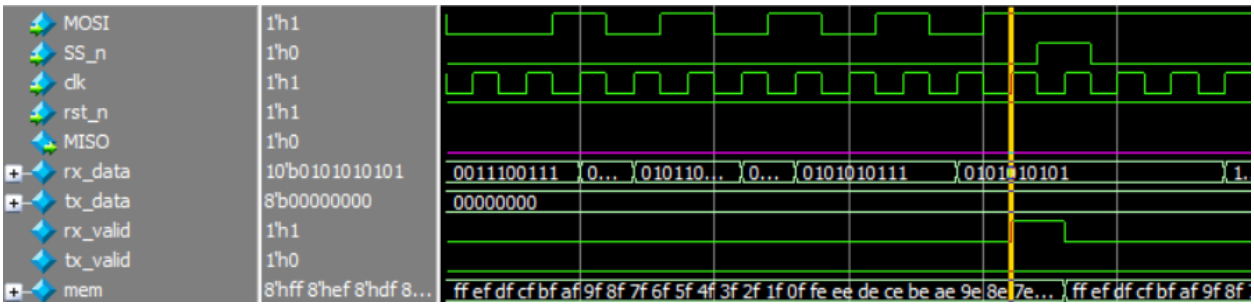


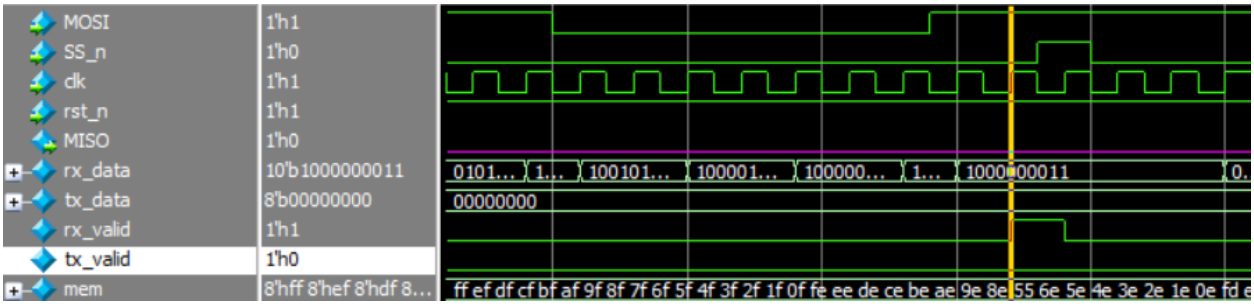
11-The Implementation (device) for one_hot encoding:

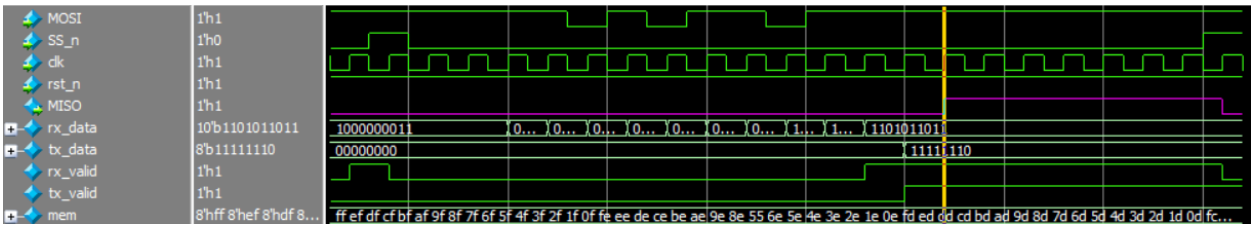


Snippets from the waveforms captured from QuestaSim:









```
# Activate reset
# MOSI = 0, MISO = x, SS_n = 1
# MOSI = 0, MISO = 0, SS_n = 1
# Case1 :Test Write Address
# MOSI = 0, MISO = 0, SS_n = 0
# MOSI = 1, MISO = 0, SS_n = 0
# MOSI = 0, MISO = 0, SS_n = 0
# MOSI = 1, MISO = 0, SS_n = 0
# rx_data = 0011100111
# MOSI = 1, MISO = 0, SS_n = 1
# Case2 :Test Write Data
# MOSI = 1, MISO = 0, SS_n = 0
# MOSI = 0, MISO = 0, SS_n = 0
# MOSI = 1, MISO = 0, SS_n = 0
# MOSI = 0, MISO = 0, SS_n = 0
# MOSI = 1, MISO = 0, SS_n = 0
# MOSI = 0, MISO = 0, SS_n = 0
# MOSI = 1, MISO = 0, SS_n = 0
# MOSI = 0, MISO = 0, SS_n = 0
# MOSI = 1, MISO = 0, SS_n = 0
# MOSI = 0, MISO = 0, SS_n = 0
# MOSI = 1, MISO = 0, SS_n = 0
# rx_data = 0101010101
# MOSI = 1, MISO = 0, SS_n = 1
# Case3 :Test Read Address
# MOSI = 1, MISO = 0, SS_n = 0
# MOSI = 0, MISO = 0, SS_n = 0
# MOSI = 1, MISO = 0, SS_n = 0
# rx_data = 1000000011
# MOSI = 1, MISO = 0, SS_n = 1
# Case4 :Test Read data
# MOSI = 1, MISO = 0, SS_n = 0
# MOSI = 0, MISO = 0, SS_n = 0
# MOSI = 1, MISO = 0, SS_n = 0
# MOSI = 0, MISO = 0, SS_n = 0
# MOSI = 1, MISO = 0, SS_n = 0
# MOSI = 0, MISO = 0, SS_n = 0
# MOSI = 1, MISO = 0, SS_n = 0
# MOSI = 1, MISO = 1, SS_n = 0
# tx_data = 11111110
# MOSI = 1, MISO = 1, SS_n = 1
# MOSI = 1, MISO = 0, SS_n = 1
```