FIFO Implementation Using FPGA

Guided by:

- **Dr. Y.V. Joshi**
- **Dr. S.S. Gajre**

Industrial Guide:

- **Mr. Vaibhav Taraate**

Team members:

- Mahesh Baraskar
- Mukul Lokhande
- Pushpak Ghatode
- Tejaswini Khairnar
- Ratnamala Patil

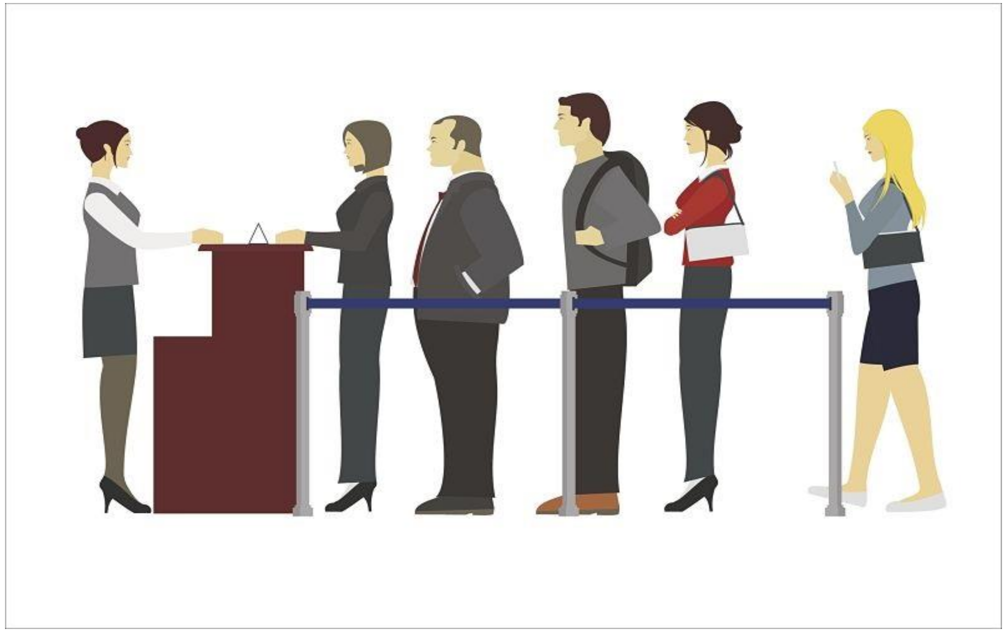
Why FIFO required?

- Exchange of data between two different PCBs working at different speed
- Otherwise slowest component will determine the speed

➢What is FIFO?

- First In First Out

  Ex. Queue at ticket counter



➢ FIFO Implementation

- Software

  Flexible to change in requirements

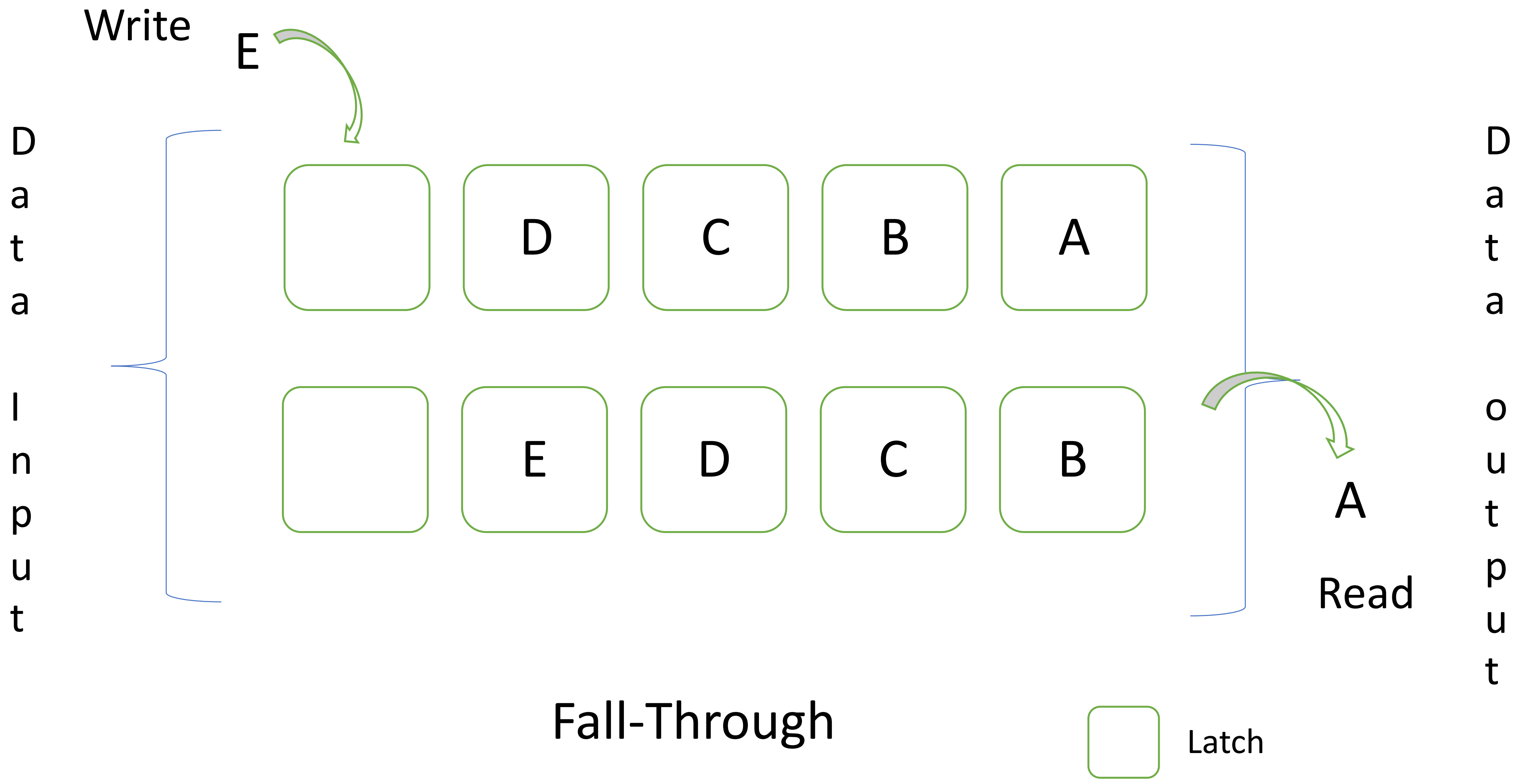
- Hardware

  Speed is High

Types of FIFOs

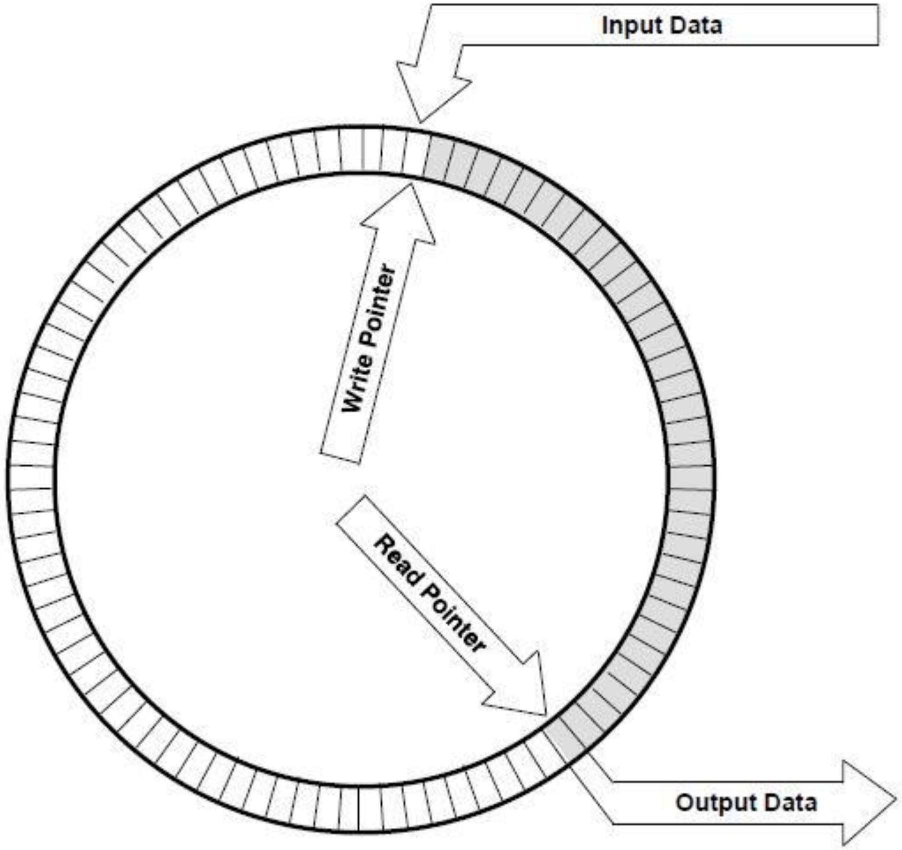
- ➢Shift Register
  - Fixed number of stored data word.
- ➢Exclusively Read/Write FIFO
  - Variable number of stored data words
  - Timing restrictions - synchronous
- ➢Concurrent Read/Write FIFO
  - Variable number of stored data words
  - No timing restriction – asynchronous
  - Two circuits with different frequencies can be connected together without the need of a synchronizing circuit

Architecture



Architecture (Contd.)

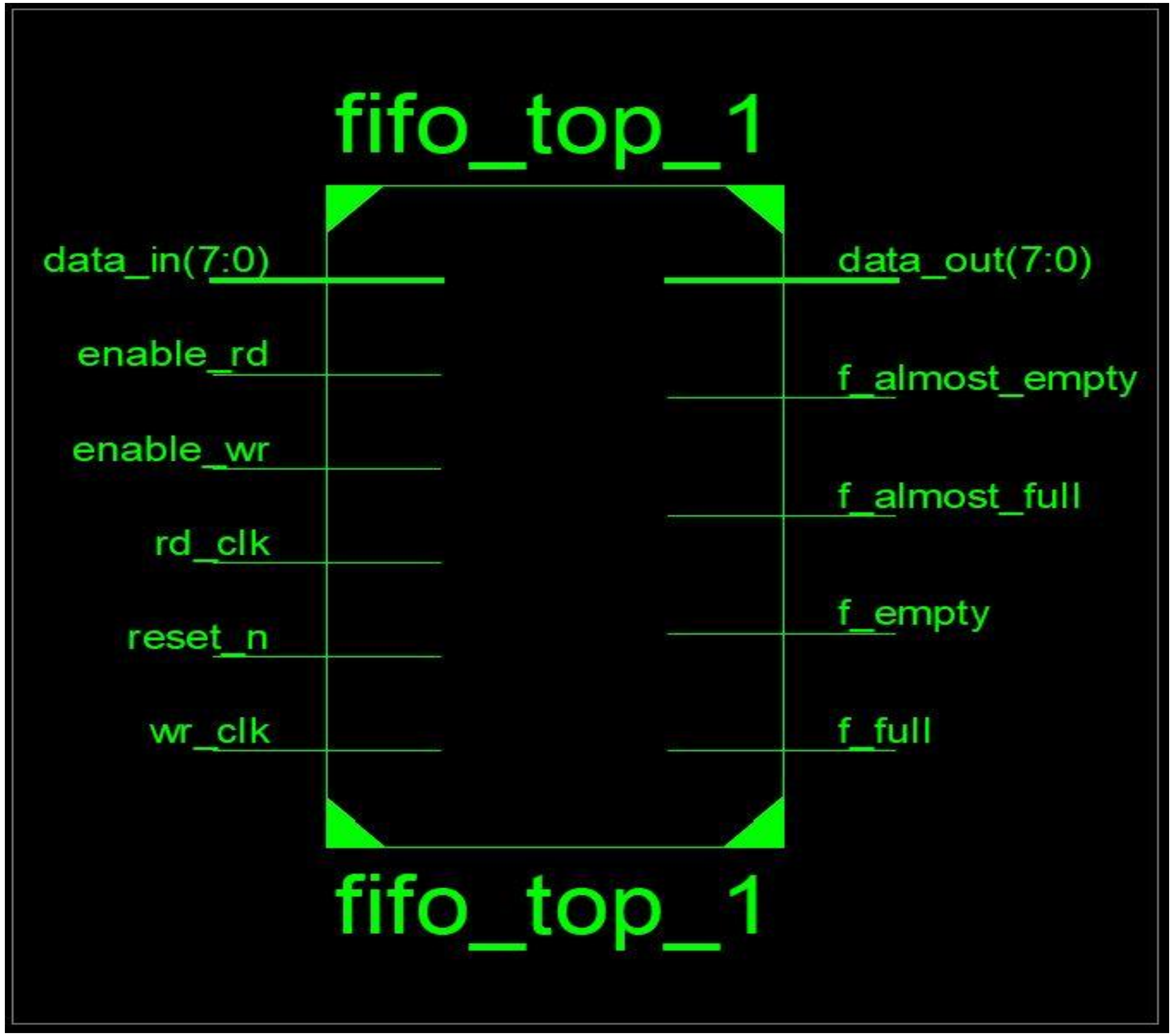
Circular FIFO with two pointer



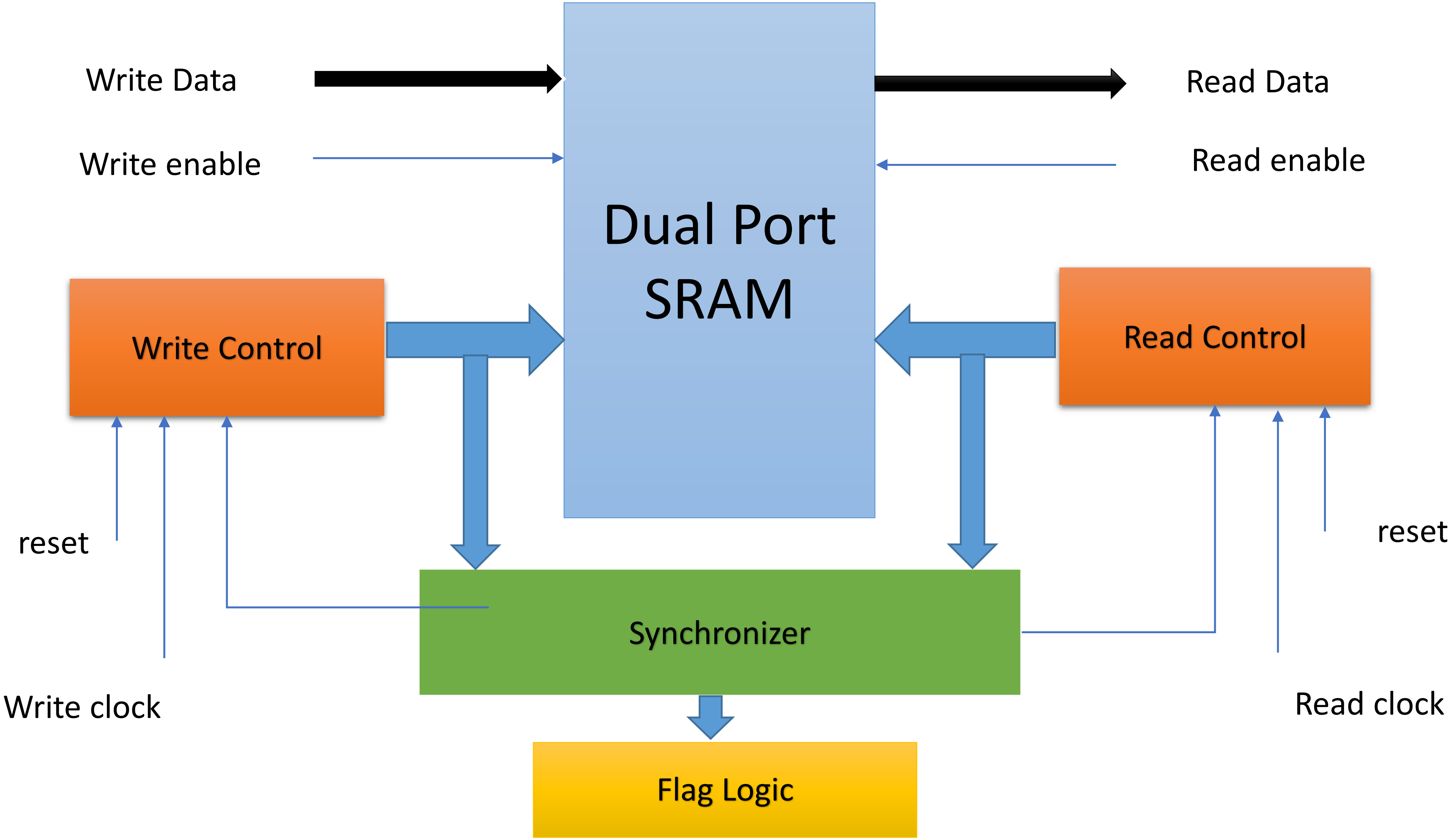
SPECIFICATIONS

- BURST Size : 1.5K bytes
- FIFO Size : 1 K bytes
- Input Data Pins: 8
- Input Pins: 2 (R/W enable)
- Output Data Pins: 8
- Write Pointer Pins: 10
- Read Pointer Pins: 10
- Flag Pins: 5 (EMPTY, FULL, HALF FULL / EMPTY, ALMOST FULL, ALMOST EMPTY)
- Reset: 1
- Write Clock (100MHz)
- Read Clock (33MHz)

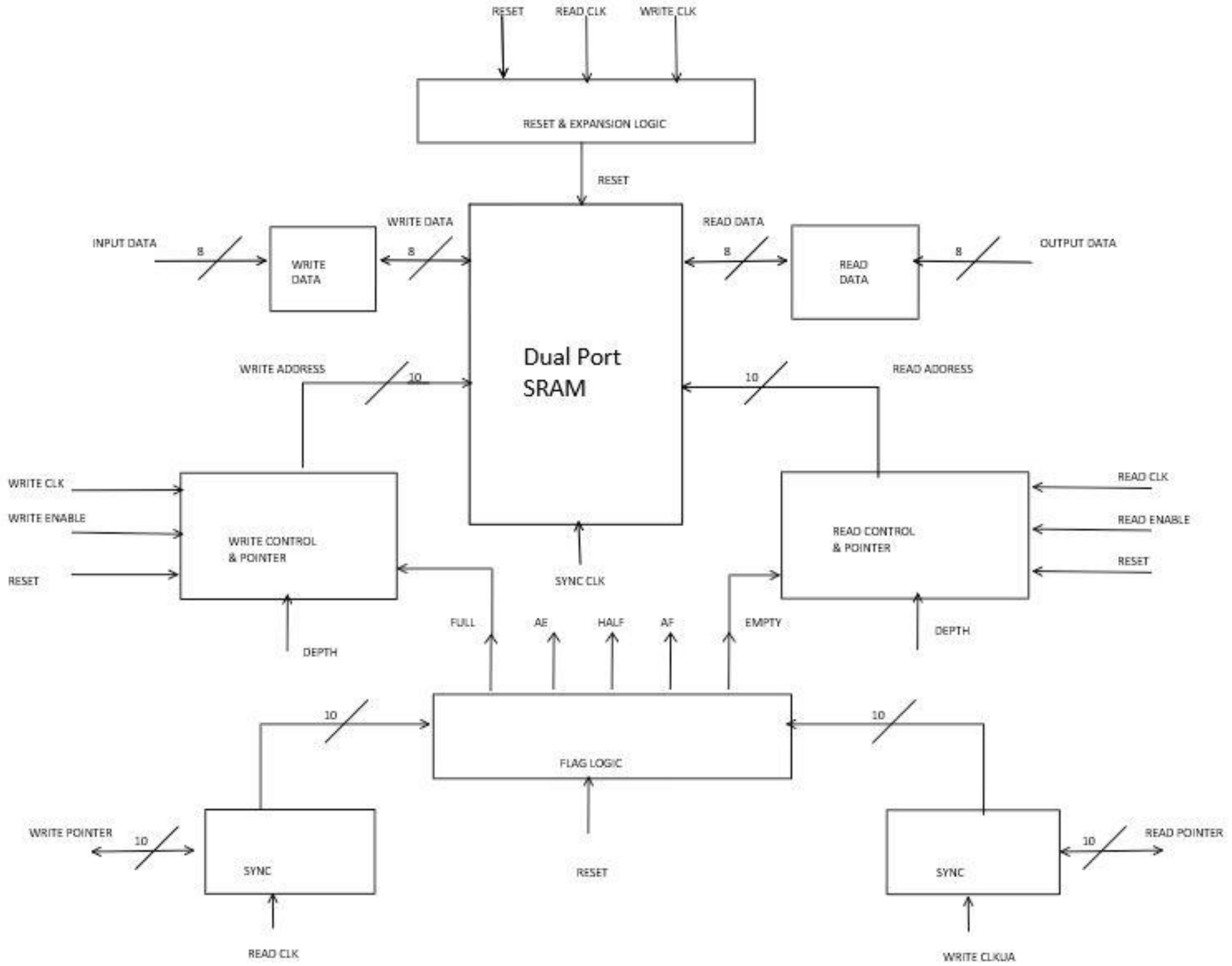
Top Level Diagram:



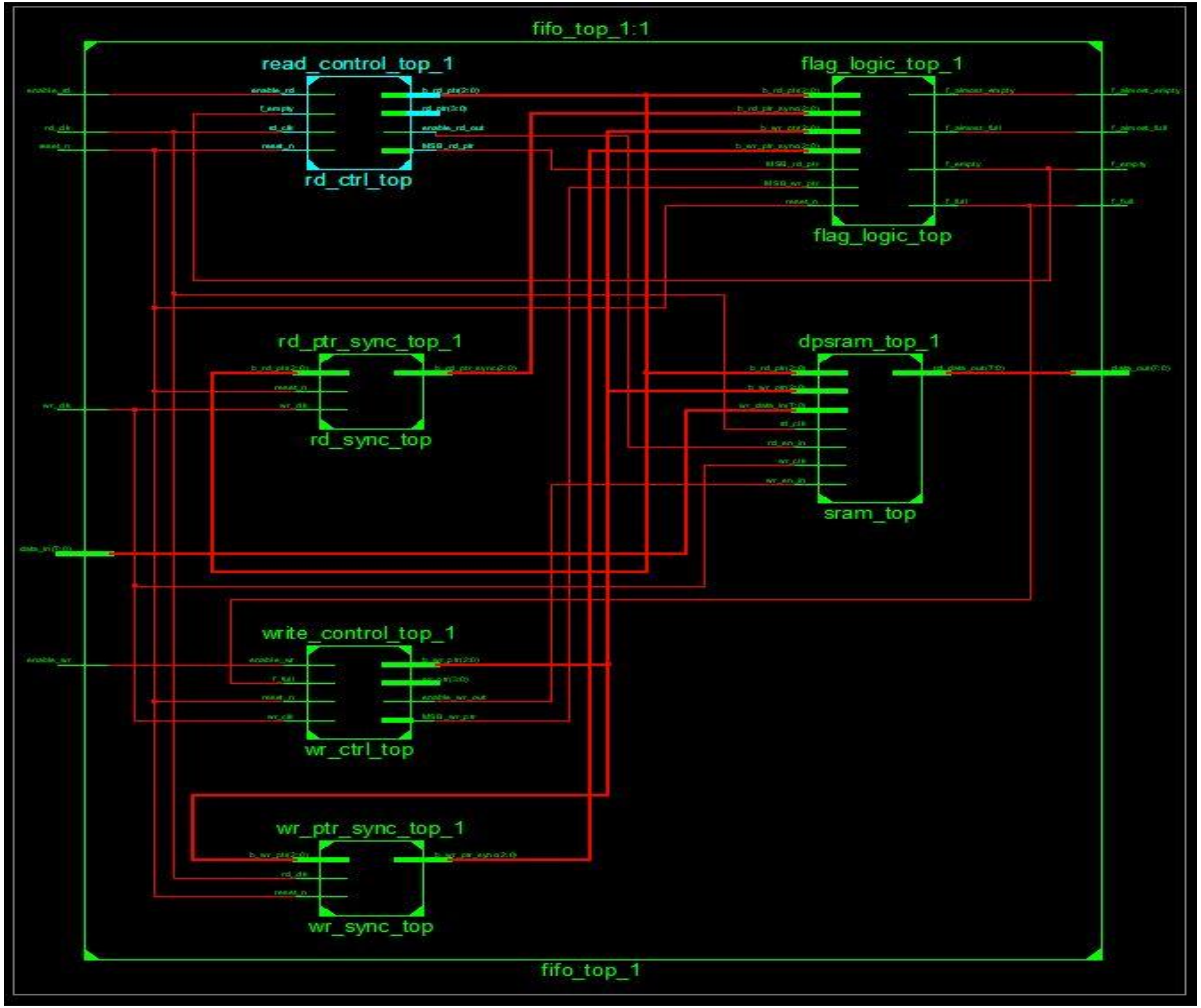
Top Level Diagram:



Micro Architecture



Top Level RTL



Pin Description – Top Level

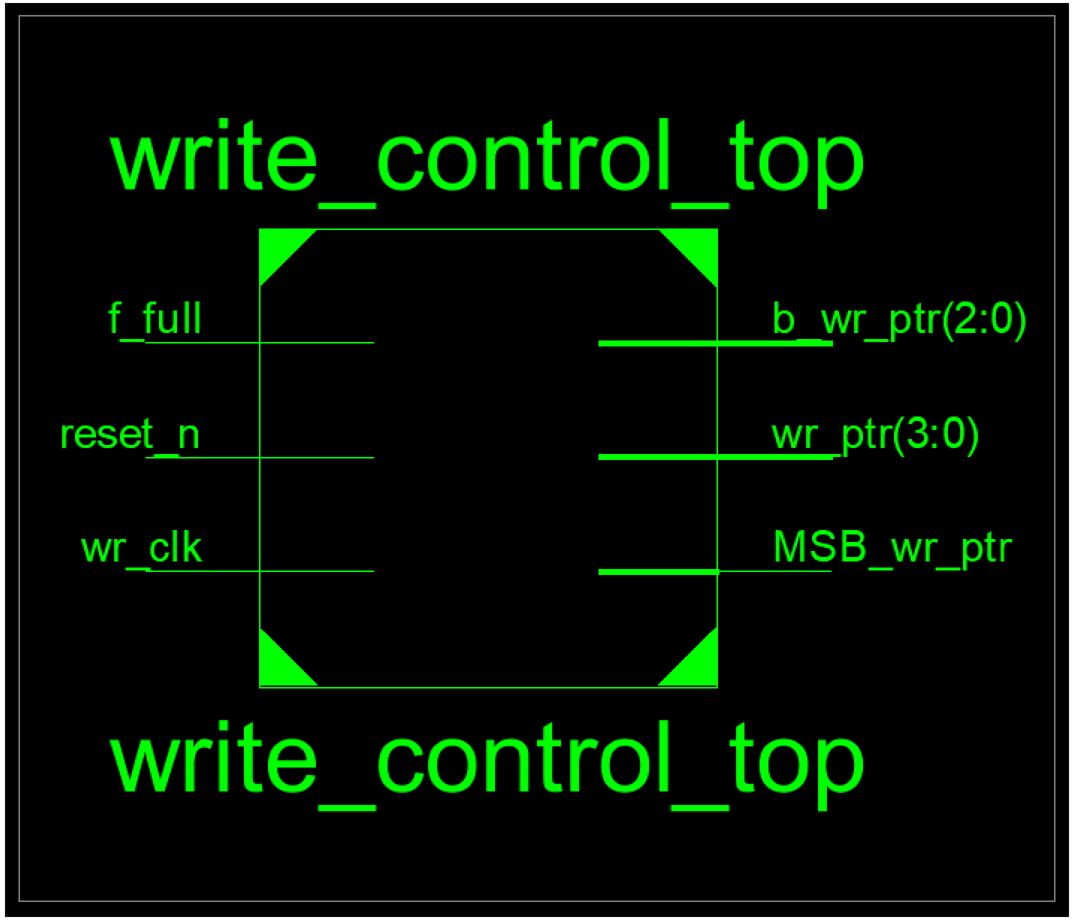
No.	Name of Pin	No. of Pins	Direction	Description
1.	Write Data	8	Input	Input data to SRAM
2.	Read Data	8	Output	Output data from SRAM
3.	Write Enable	1	Input	Enable write operation
4.	Read Enable	1	Input	Enable read operation
5.	Write Clock	1	Input	Clock for write operation
6.	Read Clock	1	Input	Clock for read operation
7.	Reset	1	Input	To reset FIFO memory
8.	Flag Logic	4	Output	Show status of FIFO

Important consideration in design

- Overflow:
  - Occurs when the FIFO is full and writer attempts to insert a new data element

- Underflow:
  - Occurs when the FIFO is empty and reader attempts to retrieves data from buffer

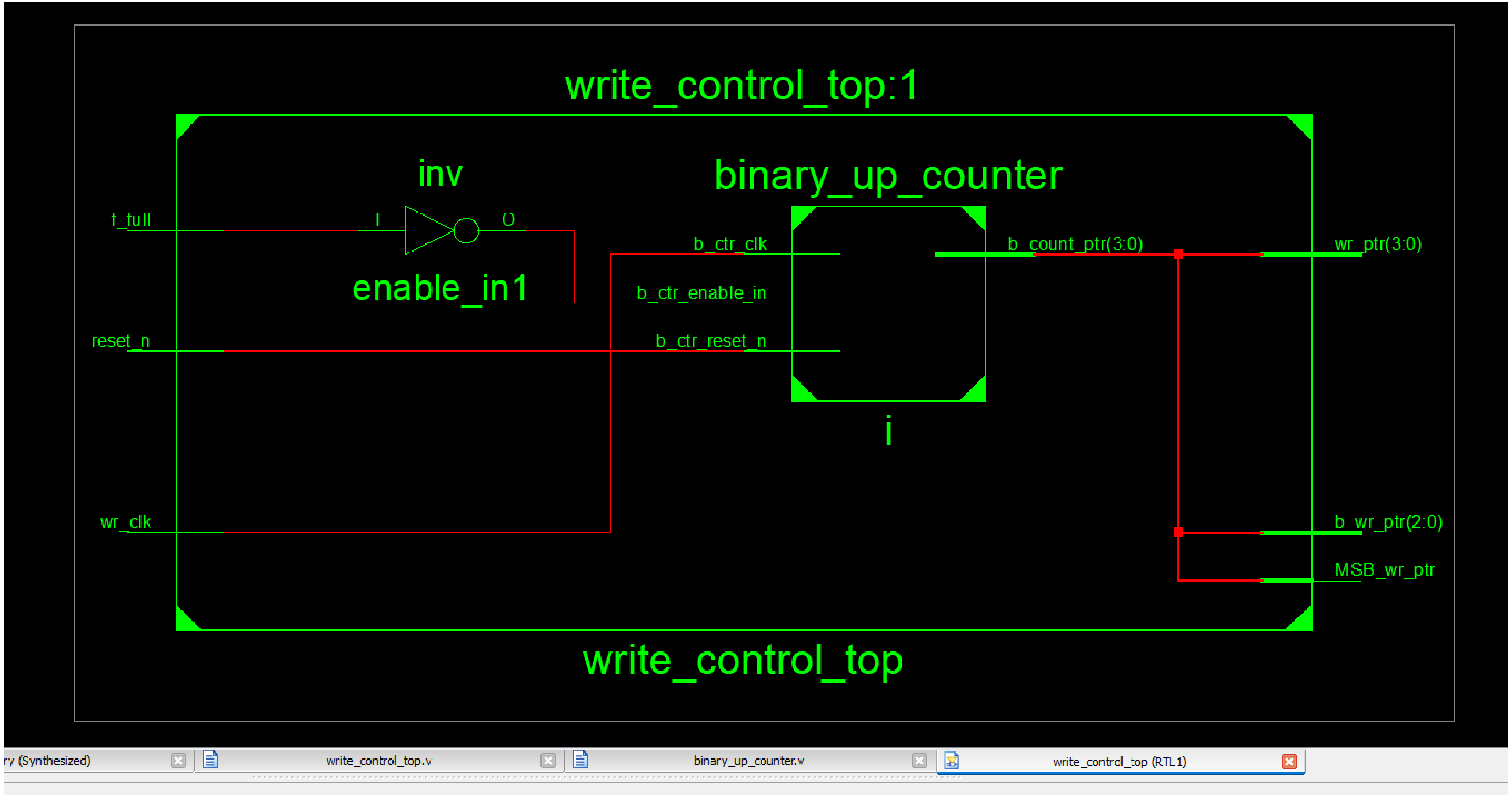
Write Control Block



Pin Functions – Write Control Block

No.	Name of Pin	No. of Pins	Direction	Function
1.	f_full	1	Input	Disable signal to write operation
2.	reset_n	1	Input	To reset write Pointer
3.	wr_clk	1	Input	Clock for write operation
4.	b_wr_ptr	10	Output	Output address to SRAM
5.	MSB_wr_ptr	1	Output	To generate flag logic

Write Control Block - RTL



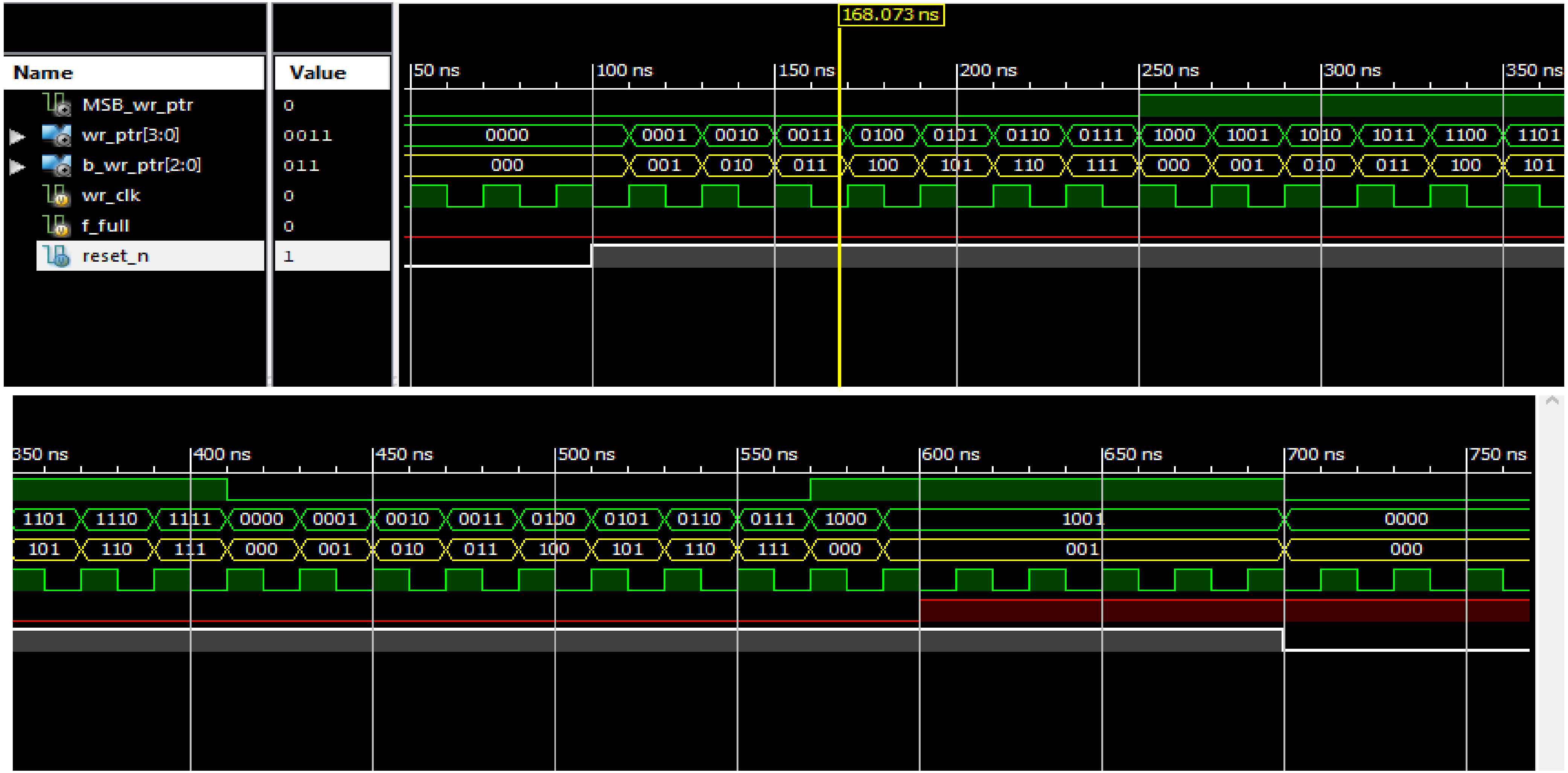
Test Cases- Writing data

1. reset_n = 0, wr_clk= x, f_full=x : **O/P wr_ptr = 3’b000;**

2. reset_n = 1, wr_clk= pos , f_full=1 : **O/P wr_ptr = Prev_value;**

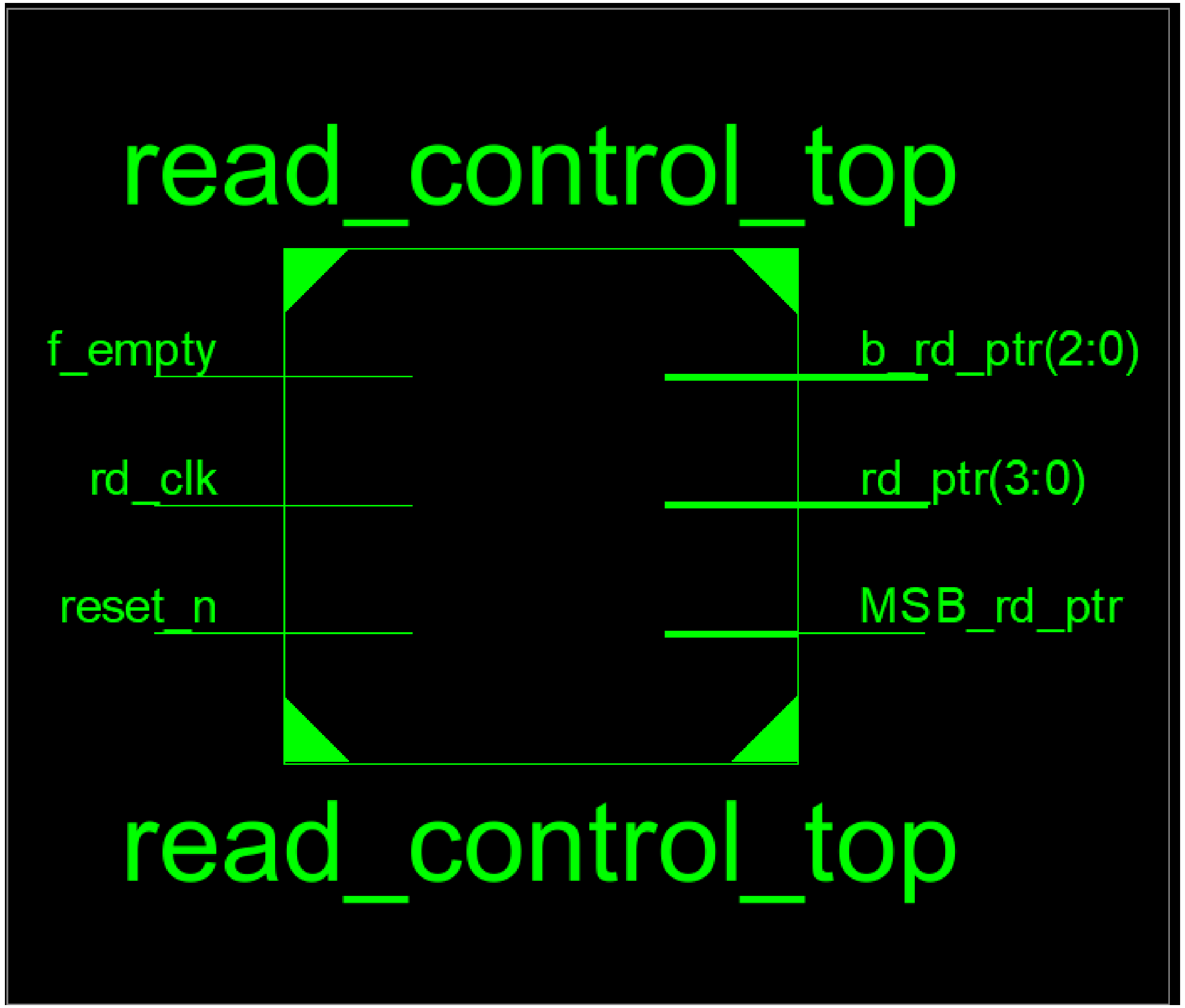
3. reset_n = 1, wr_clk= pos, f_full=0 : **O/P wr_ptr = +1’b1;**

Write Control Block – Simulation Results



Name	Value
MSB_wr_ptr	0
wr_ptr[3:0]	0011
b_wr_ptr[2:0]	011
wr_clk	0
f_full	0
reset_n	1

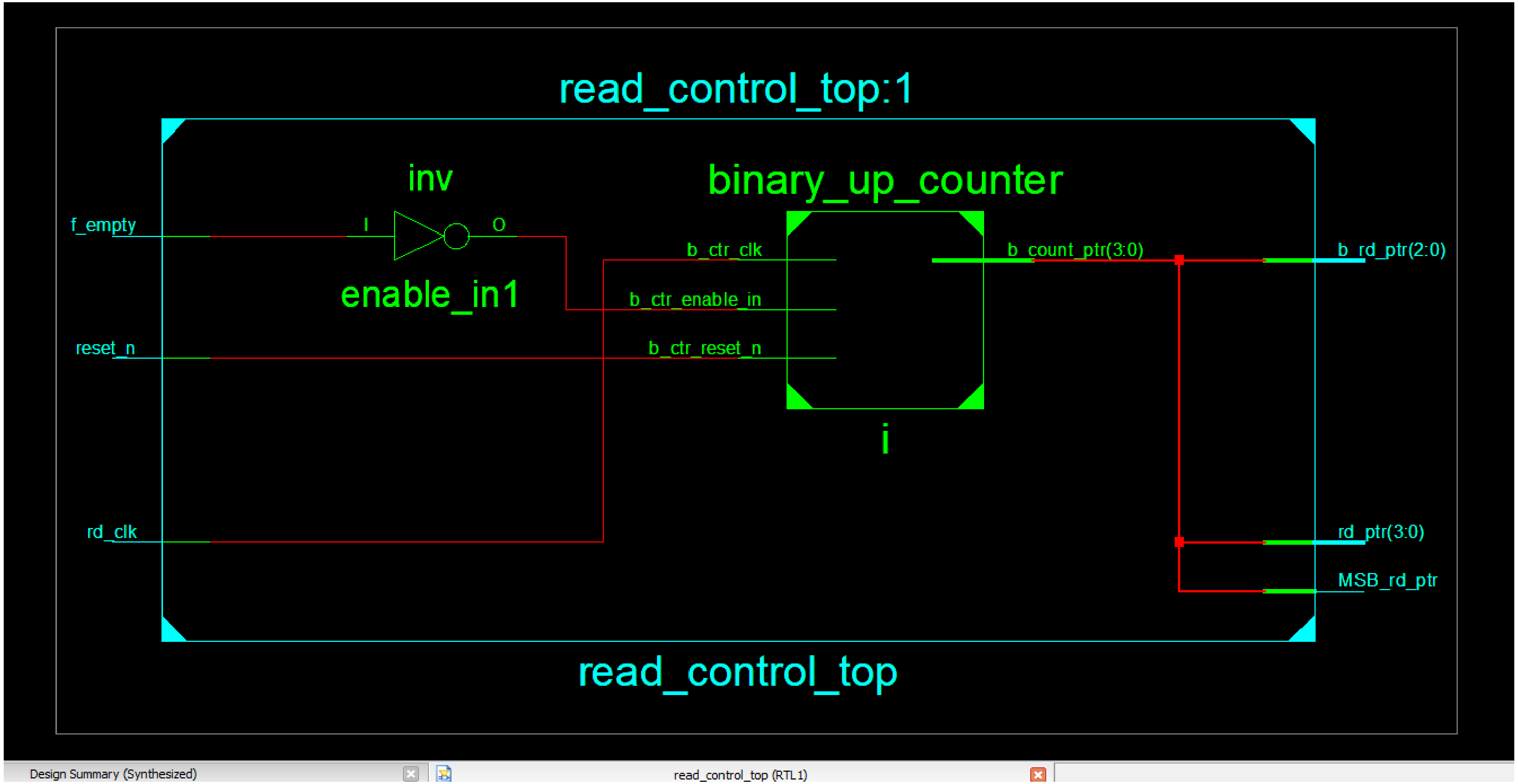
Read Control Block



Pin Functions – Read Control Block

No.	Name of Pin	No. of Pins	Direction	Function
1.	f_empty	1	Input	Show status of FIFO as empty
2.	reset_n	1	Input	To reset read Pointer
3.	rd_clk	1	Input	Clock for read operation
4.	b_rd_ptr	10	Output	Output address to SRAM
5.	MSB_rd_ptr	1	Output	To generate flag logic

Read Control Block – RTL



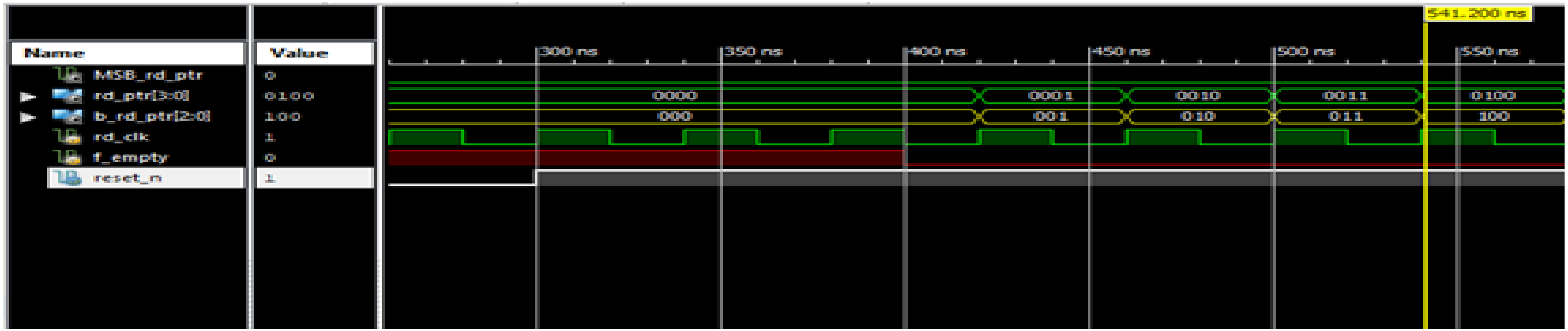
Test Cases – Reading data

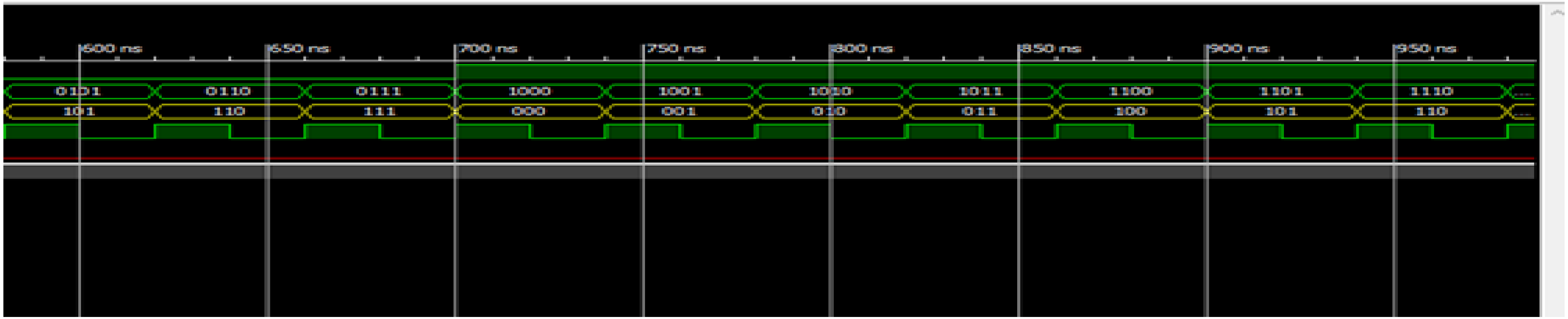
1. reset_n = 0, rd_clk= x, f_empty=x : **O/P read_ptr = 3’b000;**

2. reset_n = 1, rd_clk= pos , f_empty=1 : **O/P read_ptr = Prev_value;**

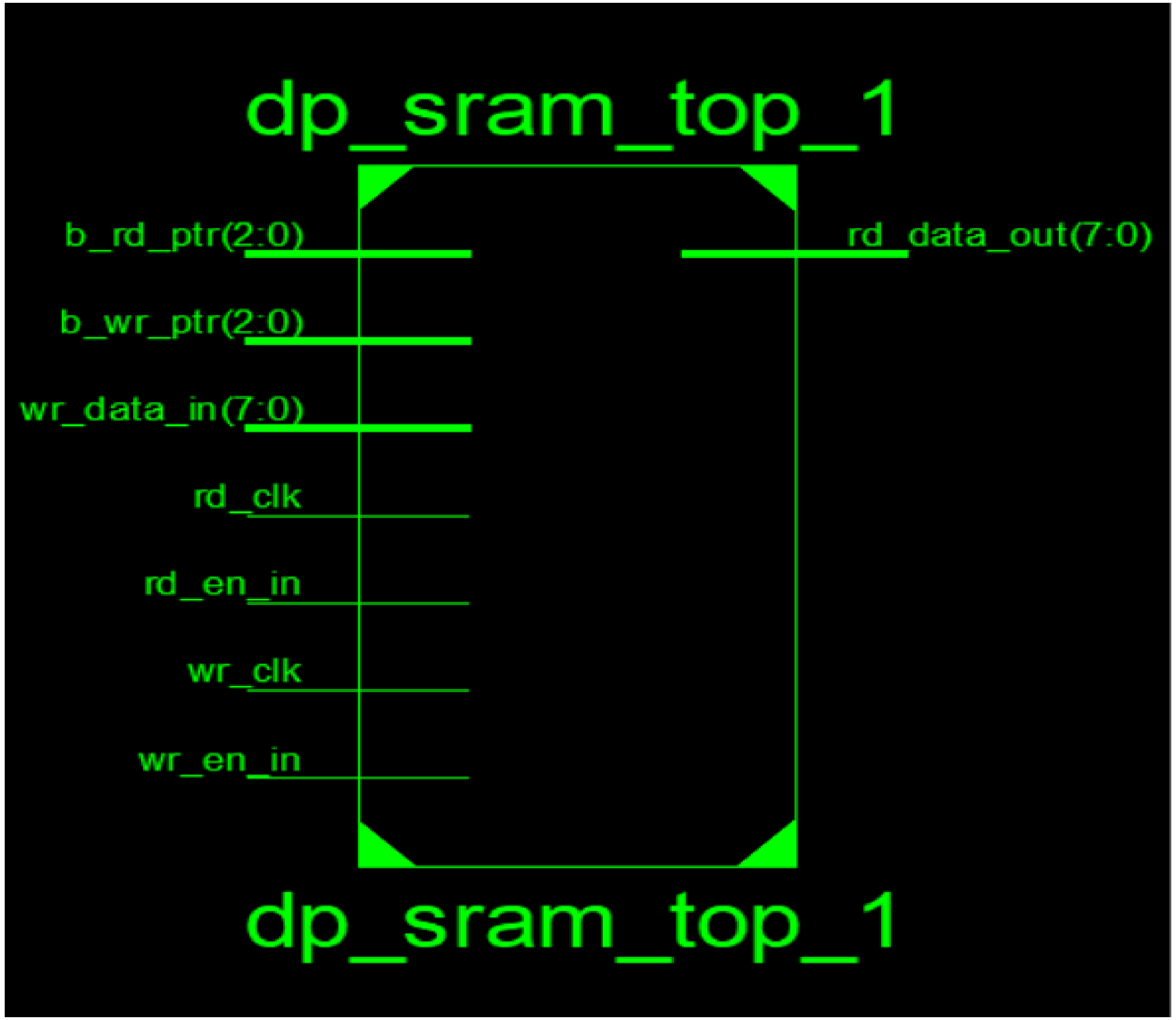
3. reset_n = 1, rd_clk= pos, f_empty=0 : **O/P read_ptr = +1’b1;**

Read Control Block – Simulation Results





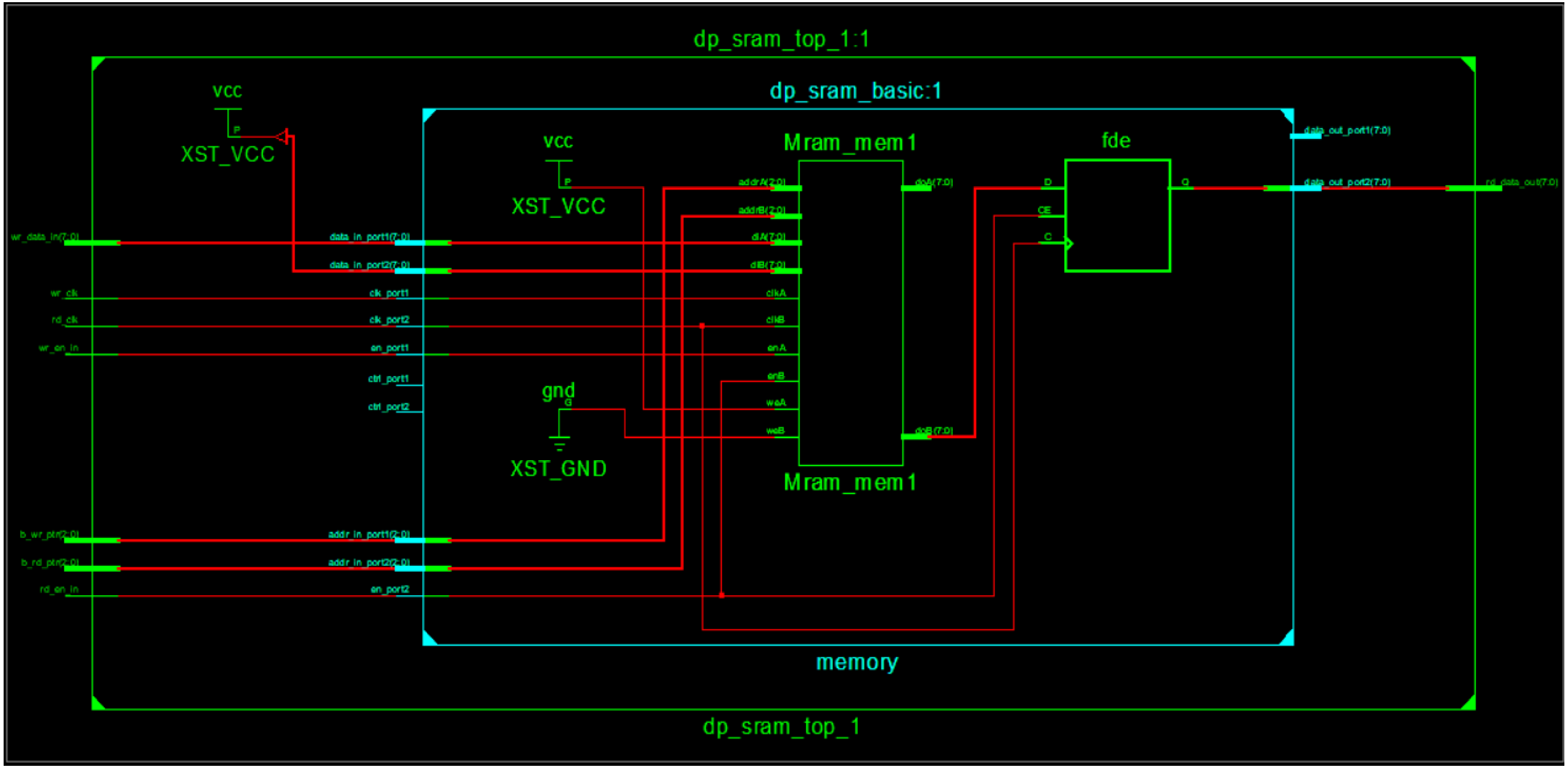
Dual Port SRAM – Top Level Diagram



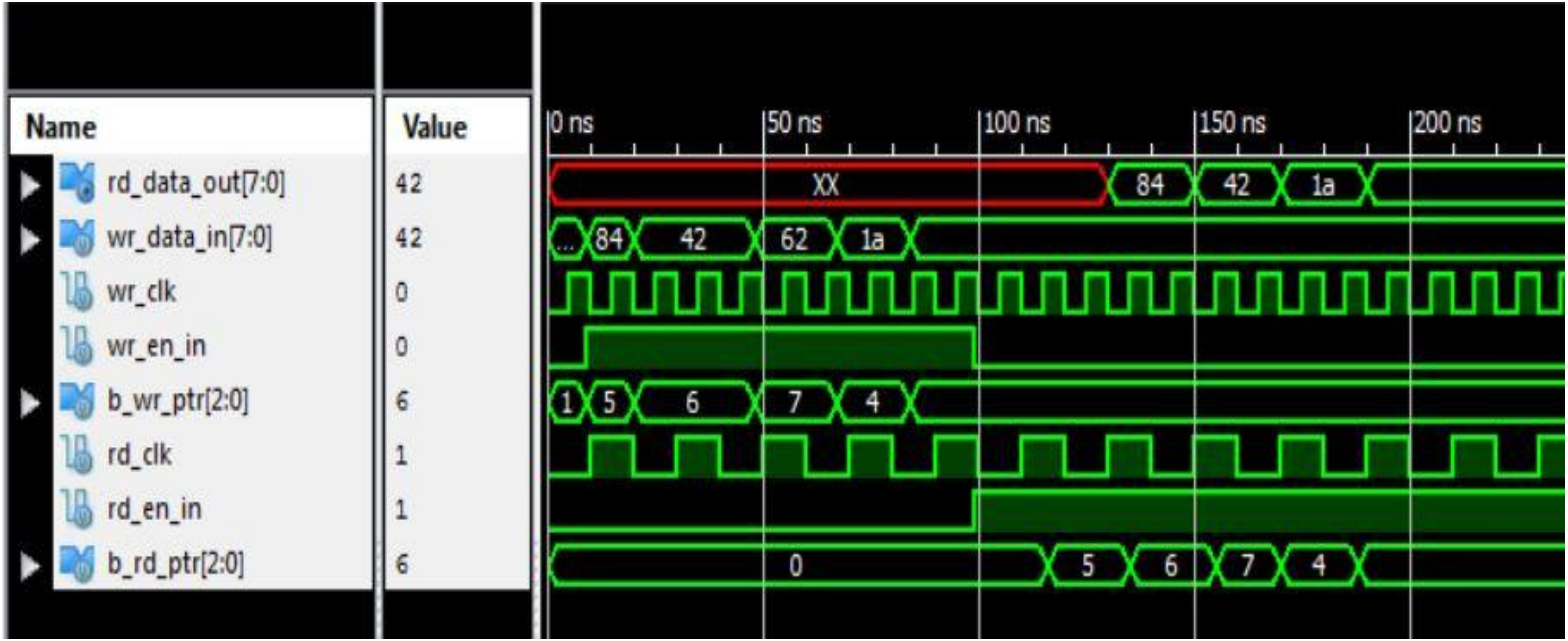
Pin Description – Dual Port SRAM

No.	Name of Pin	No. of Pins	Direction	Function
1	b_rd_ptr	10	Input	Port specifying the read address
2	b_wr_ptr	10	Input	Port specifying the write address
3	wr_data_in	8	Input	Input data port
4	rd_clk	1	Input	Read clock signal for operation
5	wr_clk	1	Input	Write clock signal for operation
6	rd_en_in	1	Input	Enable pin for the read operation
7	wr_en_in	1	Input	Enable pin for the write operation
8	rd_data_out	8	Output	Output data port

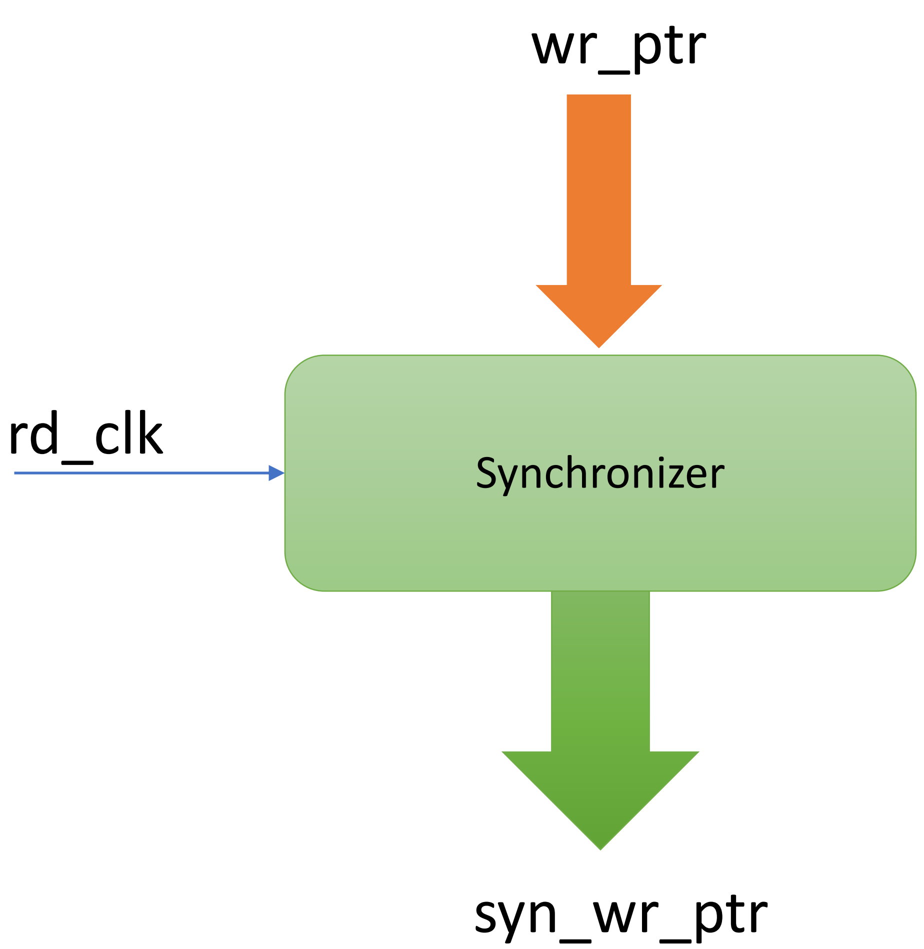
Dual Port SRAM - RTL

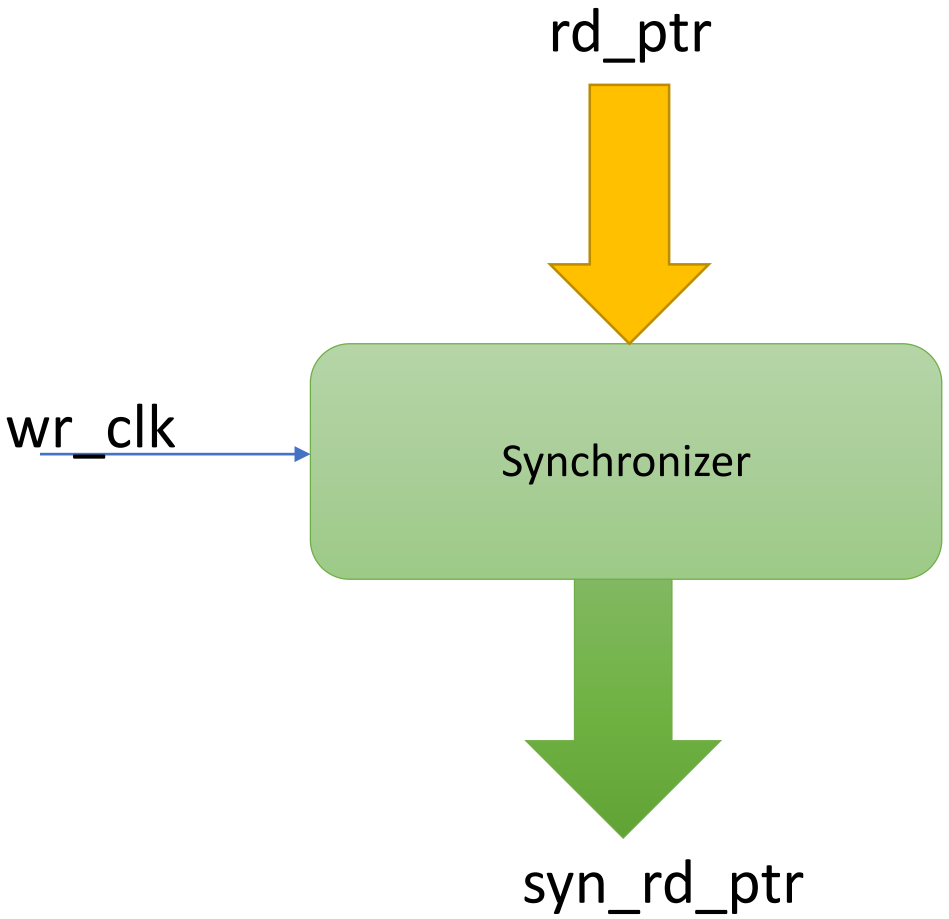


Dual Port SRAM – Simulation Results

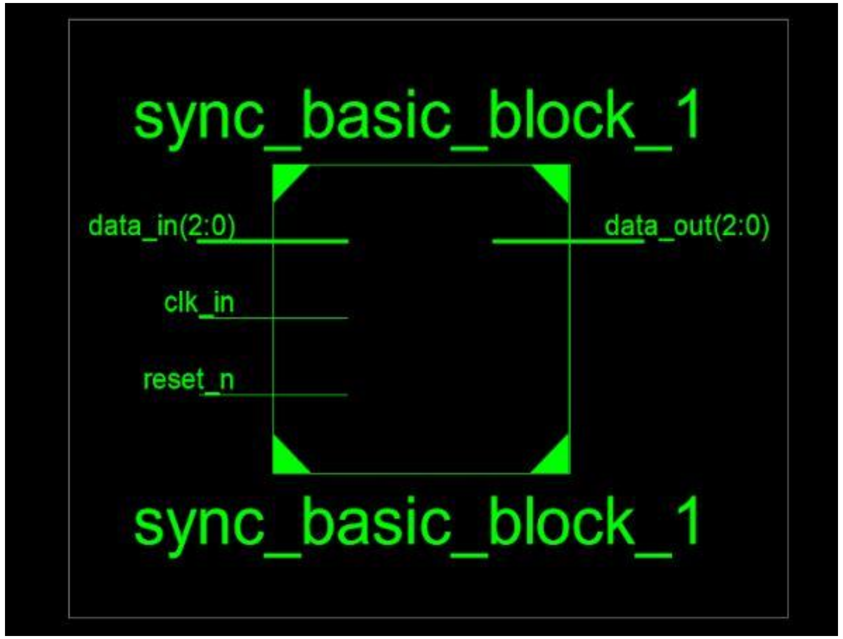


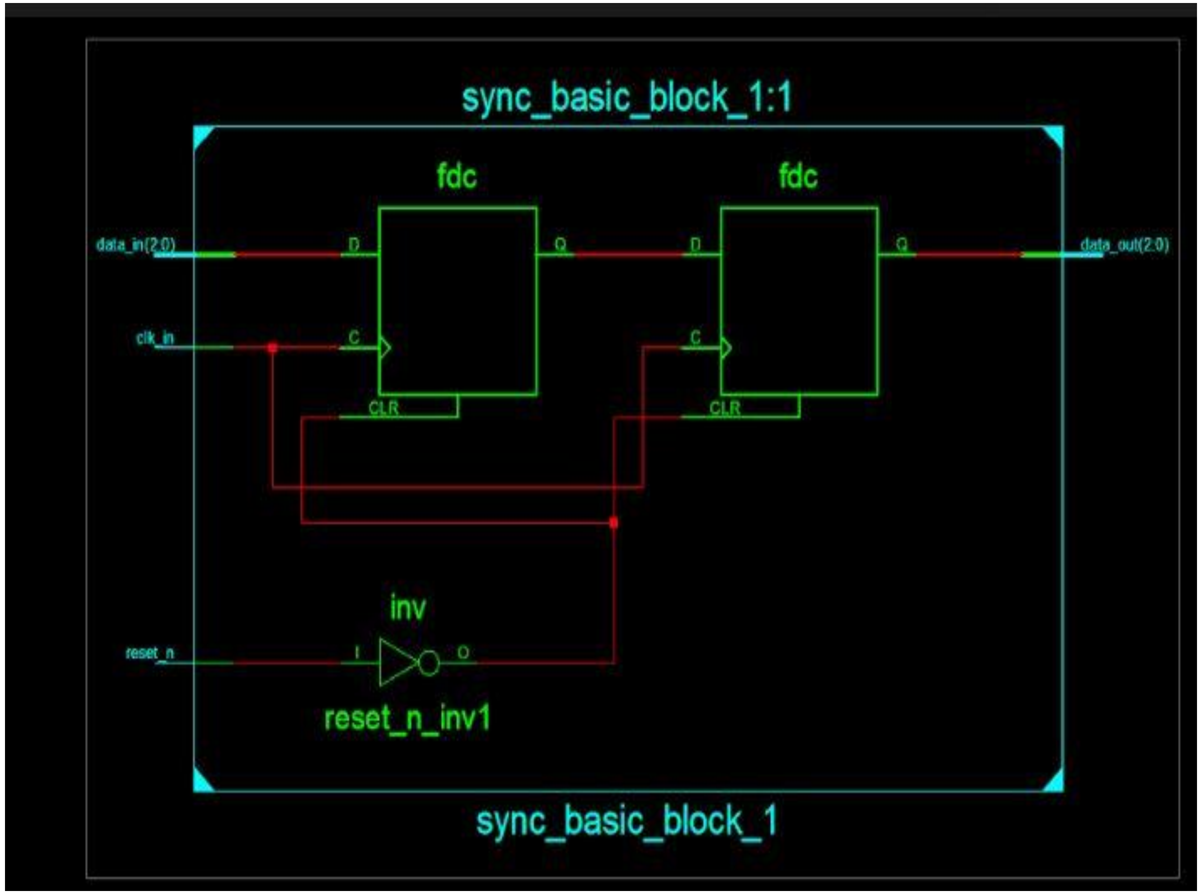
Synchronizers





Synchronizer – Top Level and RTL Diagram

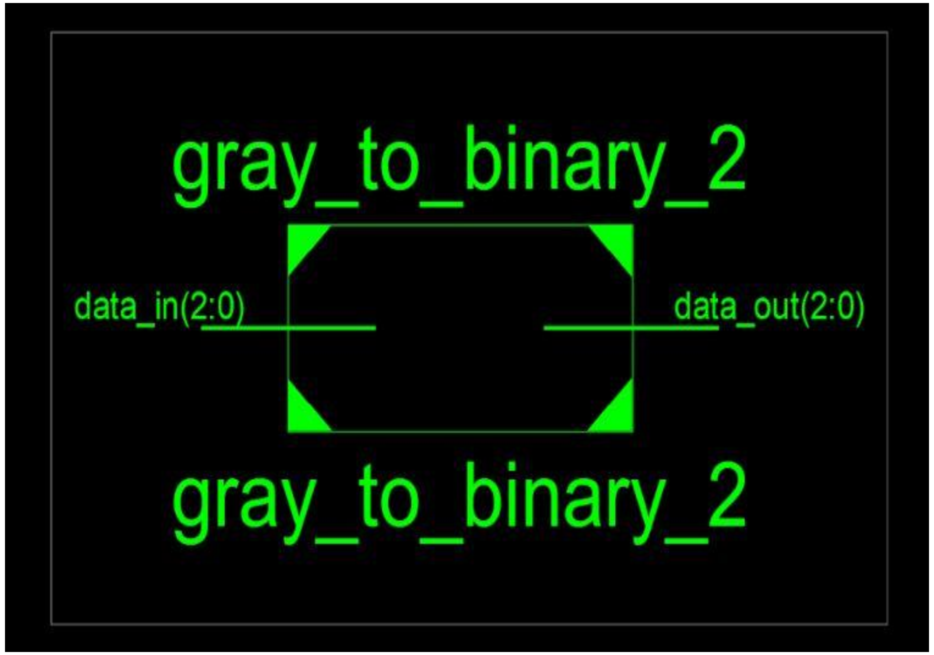


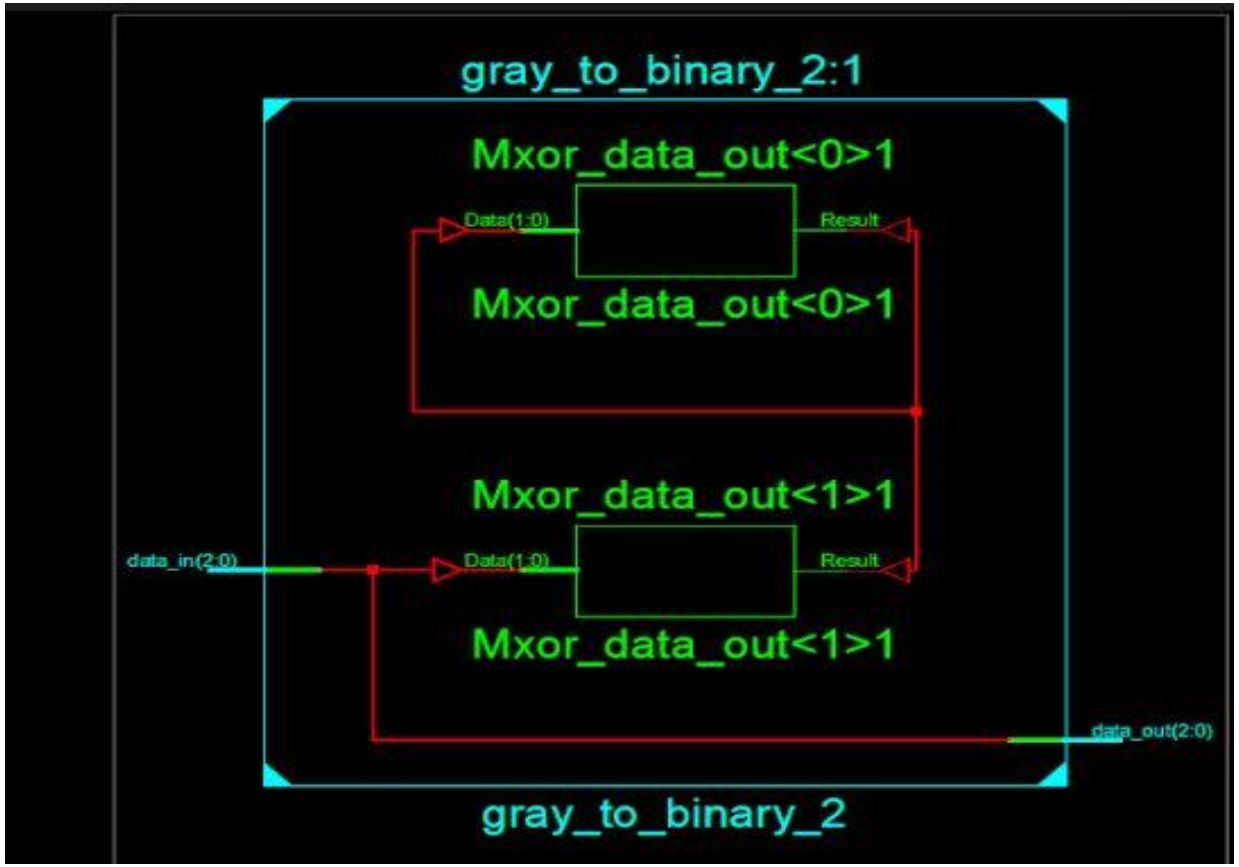


Pin Functions – Synchronizers

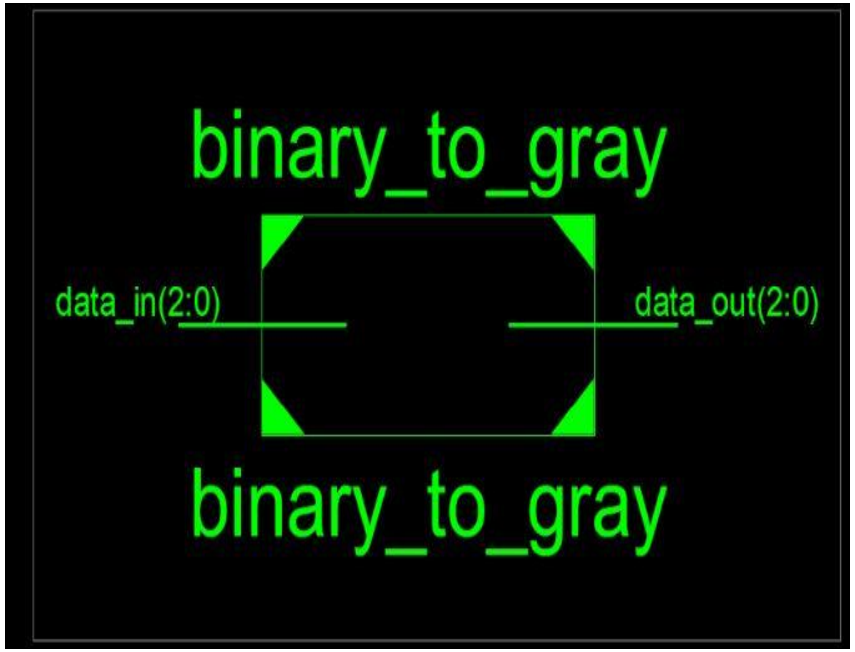
No.	Name of Pin	No. of Pins	Direction	Function
1.	wr_ptr	10	Input	Address to be synchronized
2	rd_ptr	10	Input	Address to be synchronized
3	wr_clk	1	Input	Synchronizing clock
4	rd_clk	1	Input	Synchronizing clock
5	sync_wr_ptr	10	Output	Synchronized address
6	sync_rd_ptr	10	Output	Synchronized address

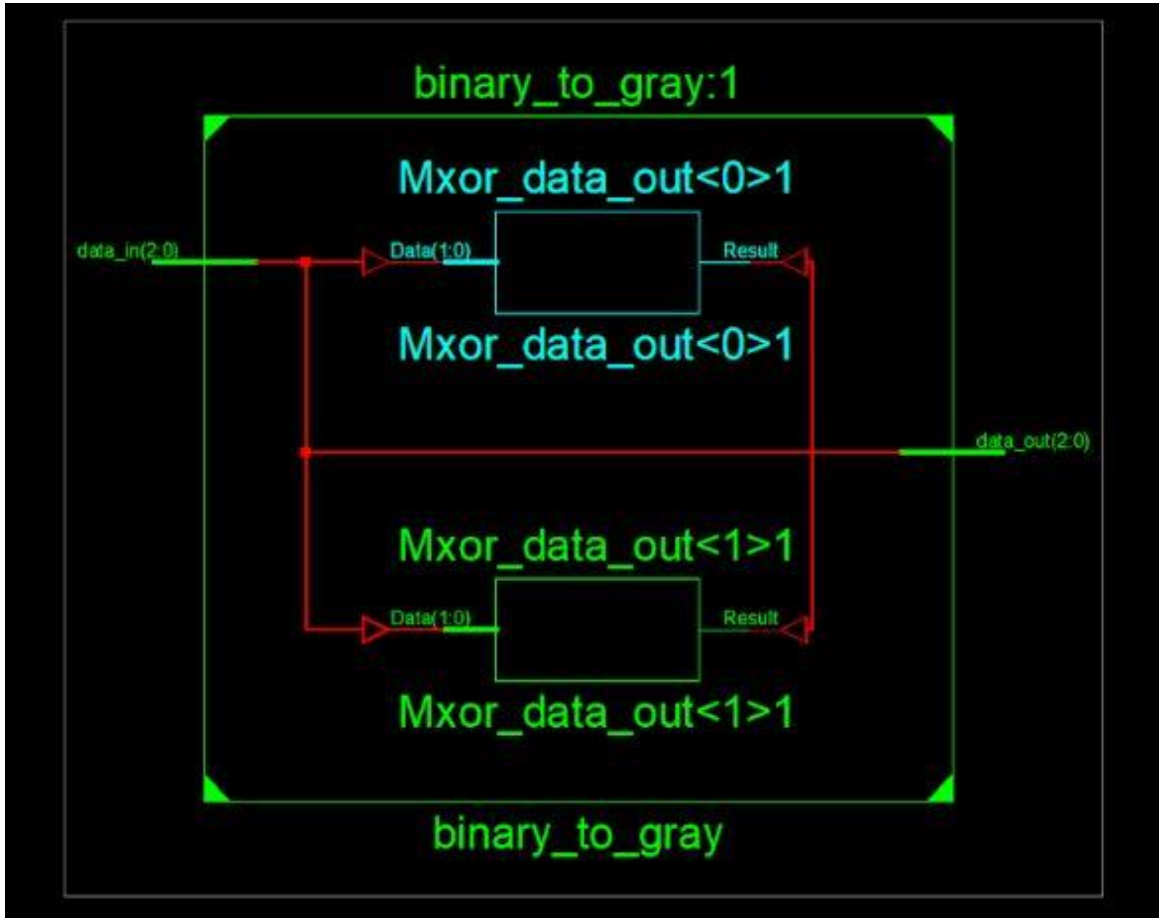
Gray to Binary Converter - RTL



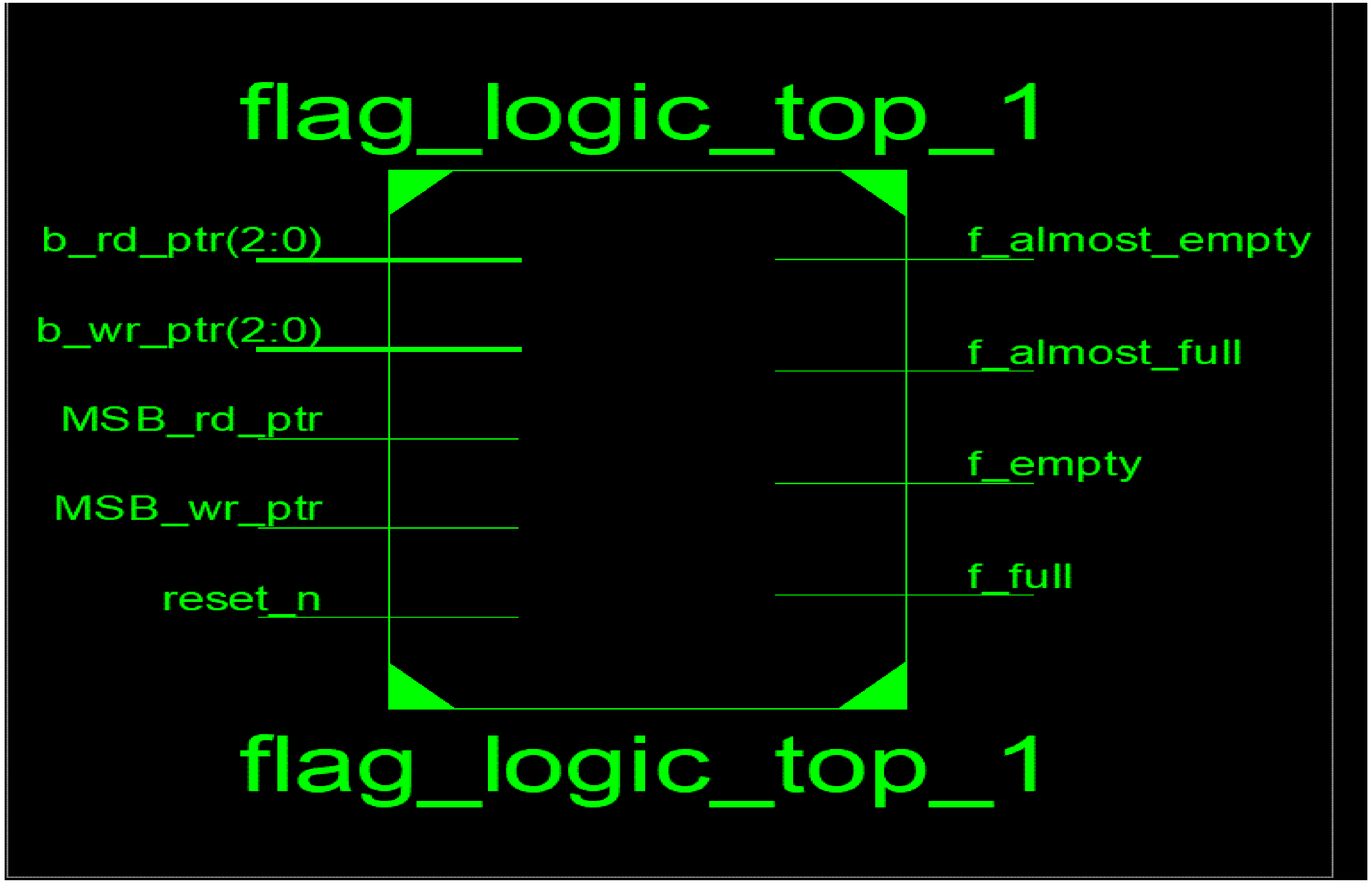


Binary to Gray Converter - RTL





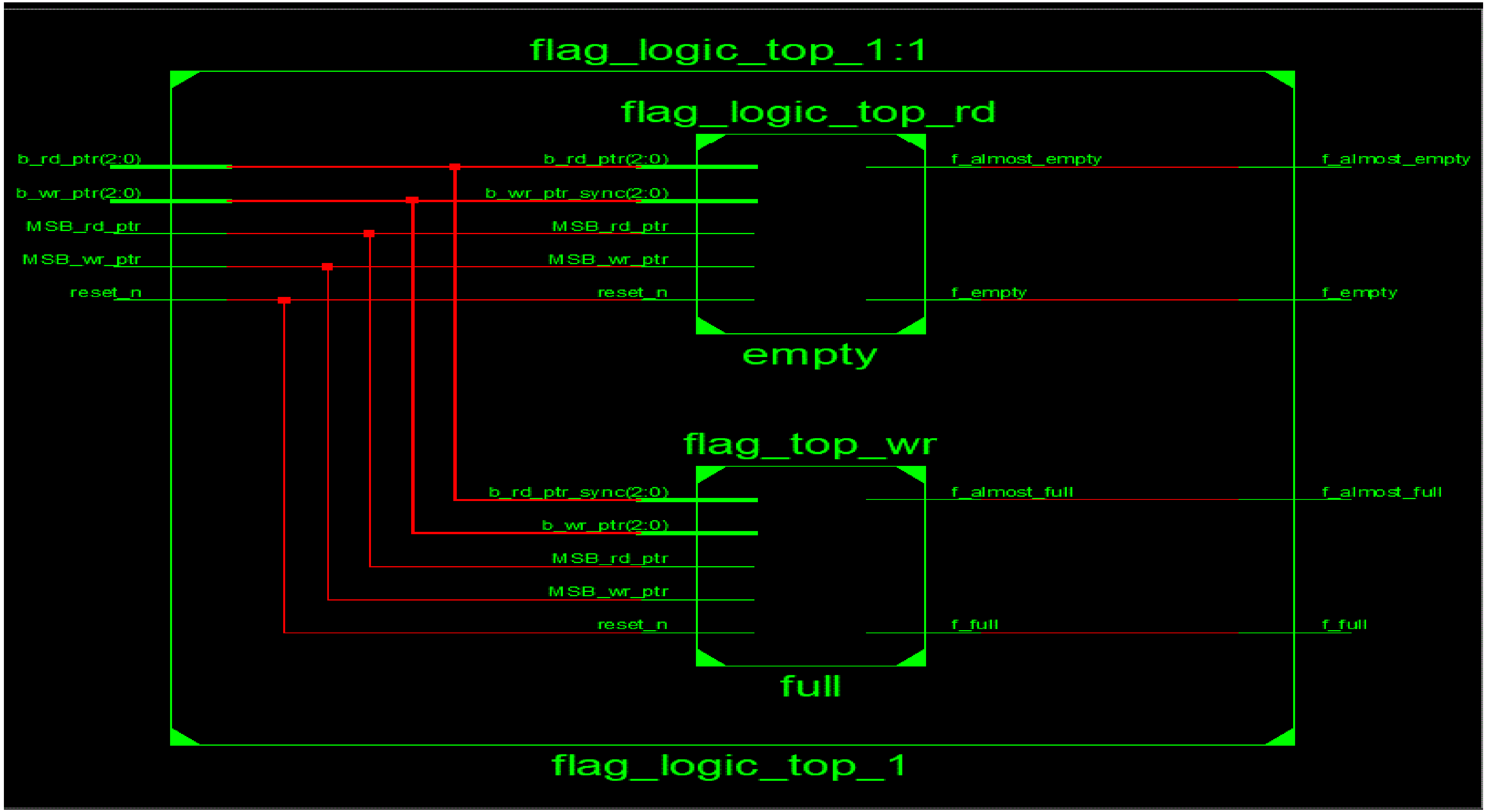
Flag logic block



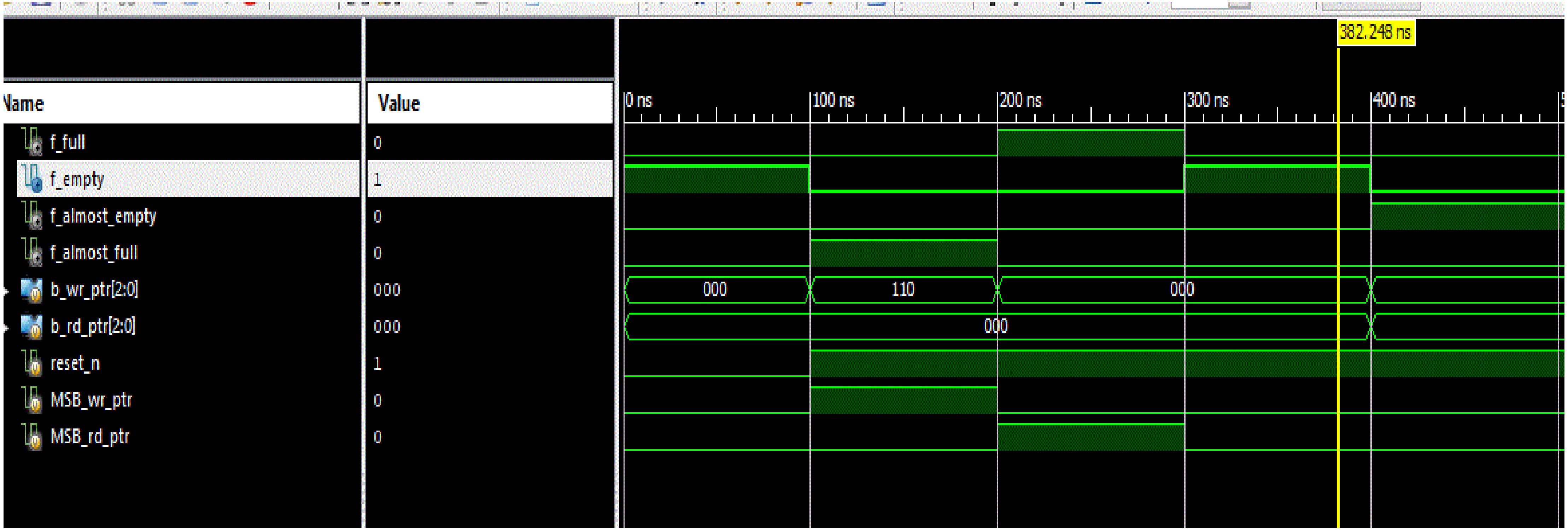
Pin Functions – Flag Logic

No.	Name of Pin	No. of Pins	Direction	Function
1.	b_wr_ptr	10	Input	Address input to comparator
2.	b_rd_ptr	10	Input	Address input to comparator
3.	Config_depth	10	Input	Limiting value of memory location
4.	Reset_n	1	Input	To set empty flag
5.	wr_clk	1	Input	To synchronize Read pointer
6.	rd_clk	1	Input	To synchronize Write pointer
7.	f_full	1	Output	Show status as FIFO is Full
8.	f_empty	1	Output	Show status as FIFO is empty
9.	f_almost _full	1	Output	FIFO is Almost empty
10.	f_almost_empty	1	Output	FIFO is Almost empty

Flag logic block –RTL



Flag logic block-Simulation



Name	Value
f_full	0
f_empty	1
f_almost_empty	0
f_almost_full	0
b_wr_ptr[2:0]	000
b_rd_ptr[2:0]	000
reset_n	1
MSB_wr_ptr	0
MSB_rd_ptr	0

Test Cases- Top Level RTL

1. reset_n =0 ; wr_en=x , rd_en = x : FIFO reset

2. reset_n =1 ; wr_en=1 & f_full = 0 , rd_en = 0

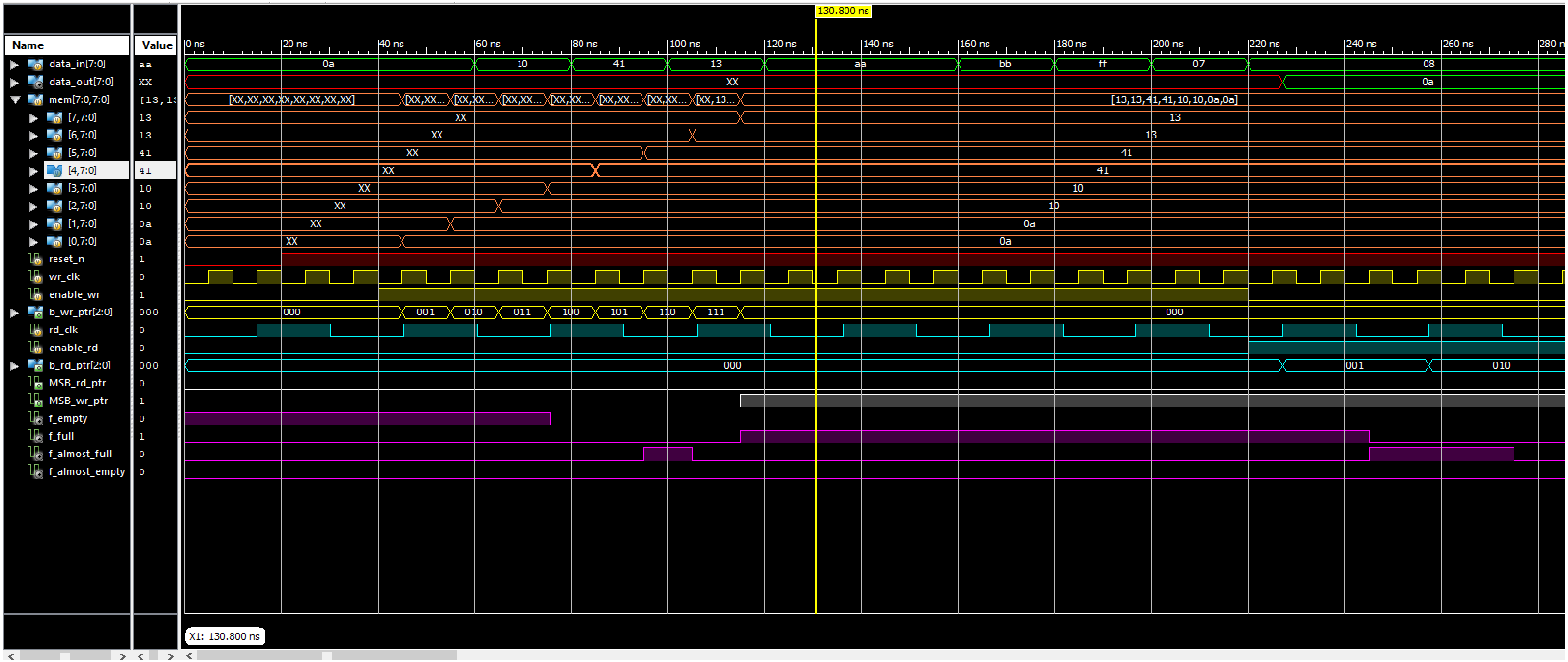
   : FIFO write operation (b_wr_ptr= +1’b1)

3. reset_n =1 ; wr_en=0 , rd_en = 1 & f_empty = 0

   : FIFO read operation (b_rd_ptr= +1’b1)

4. reset_n =1 ; wr_en=1 , rd_en = 1 : Async read & write

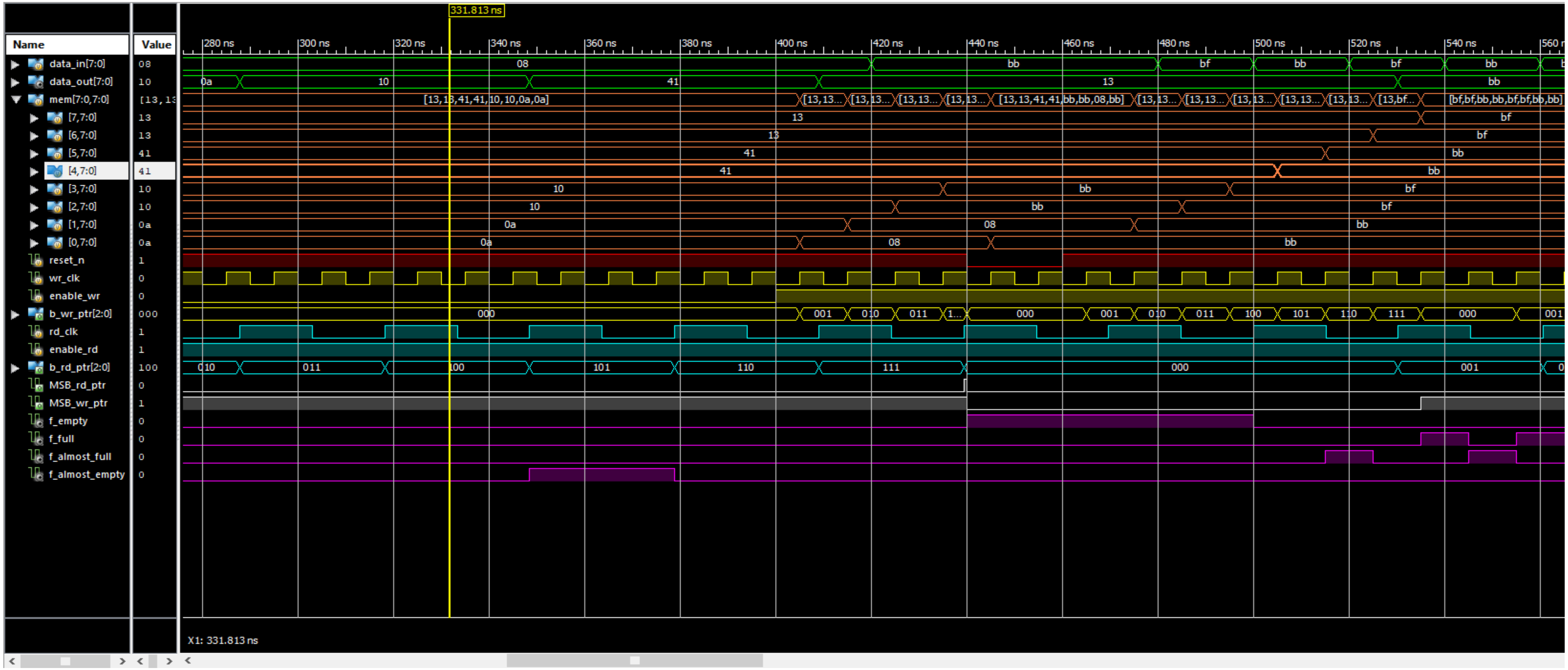
Top Level– Simulation Results



Name	Value
data_in[7:0]	aa
data_out[7:0]	XX
mem[7:0,7:0]	[13,13
[7,7:0]	13
[6,7:0]	13
[5,7:0]	41
[4,7:0]	41
[3,7:0]	10
[2,7:0]	10
[1,7:0]	0a
[0,7:0]	0a
reset_n	1
wr_clk	0
enable_wr	1
b_wr_ptr[2:0]	000
rd_clk	0
enable_rd	0
b_rd_ptr[2:0]	000
MSB_rd_ptr	0
MSB_wr_ptr	1
f_empty	0
f_full	1
f_almost_full	0
f_almost_empty	0

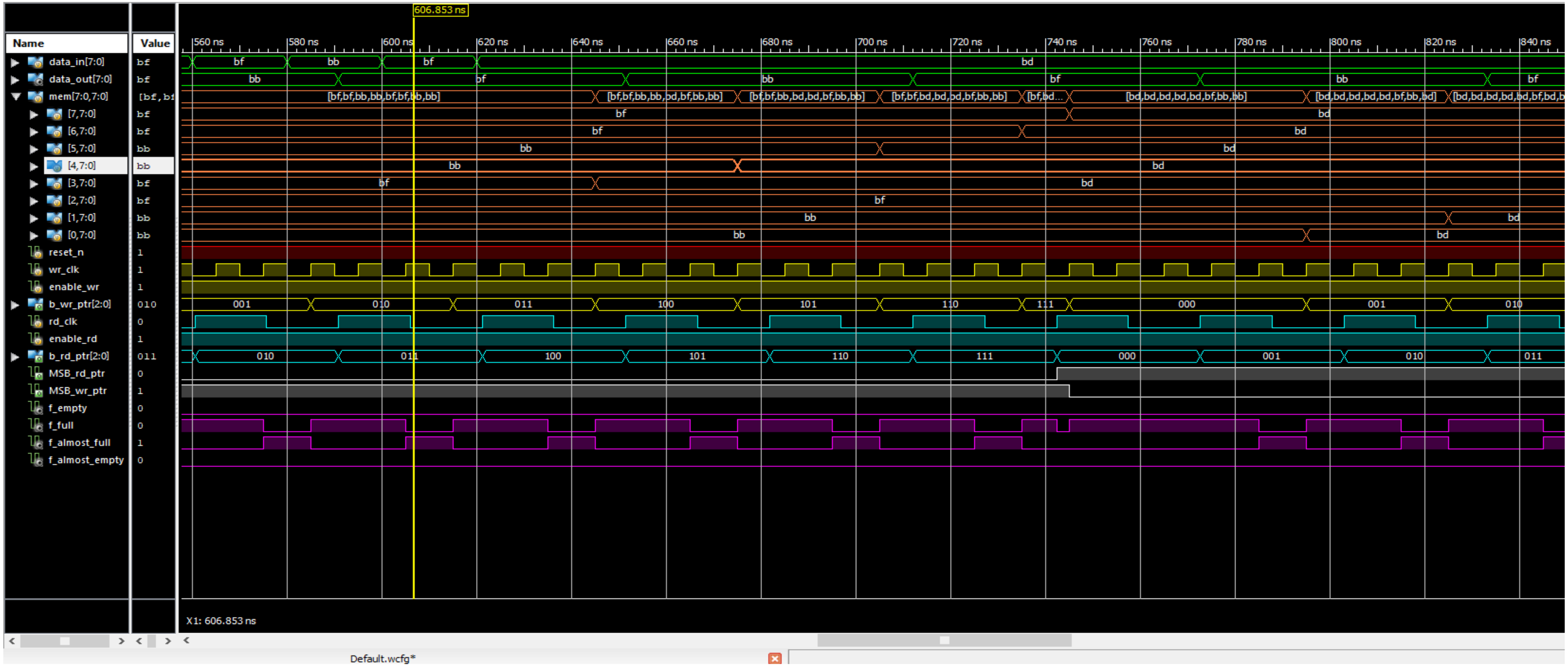
X1: 130.800 ns

Top Level– Simulation Results



Name	Value
data_in[7:0]	08
data_out[7:0]	10
mem[7:0,7:0]	[13,13
[7,7:0]	13
[6,7:0]	13
[5,7:0]	41
[4,7:0]	41
[3,7:0]	10
[2,7:0]	10
[1,7:0]	0a
[0,7:0]	0a
reset_n	1
wr_clk	0
enable_wr	0
b_wr_ptr[2:0]	000
rd_clk	1
enable_rd	1
b_rd_ptr[2:0]	100
MSB_rd_ptr	0
MSB_wr_ptr	1
f_empty	0
f_full	0
f_almost_full	0
f_almost_empty	0

Top Level– Simulation Results



Name	Value
data_in[7:0]	bf
data_out[7:0]	bf
mem[7:0,7:0]	[bf,bf
[7,7:0]	bf
[6,7:0]	bf
[5,7:0]	bb
[4,7:0]	bb
[3,7:0]	bf
[2,7:0]	bf
[1,7:0]	bb
[0,7:0]	bb
reset_n	1
wr_clk	1
enable_wr	1
b_wr_ptr[2:0]	010
rd_clk	0
enable_rd	1
b_rd_ptr[2:0]	011
MSB_rd_ptr	0
MSB_wr_ptr	1
f_empty	0
f_full	0
f_almost_full	1
f_almost_empty	0

X1: 606.853 ns

Default.wcfg*

Applications

- Packet Buffering :
  - Data written into the FIFO can be stored until the system on the output of the FIFO is ready to accept the data.
- Frequency coupling:
  - FIFO provides frequency coupling, taking data in at one rate and output the data at another data rate. The input and output data rates of the FIFO being controlled by the discrete Read and Write clock signals.
- Bus Matching :
  - Data transfer may need to take place between separate digital domains with different bus widths. Here the FIFO acts as a bridge between the domains, channeling the data from the input of a particular bus width , to the output with another bus width.

Advantages

- Data Rate Matching.
- Buffering and bus matching parallel FIFO structure allow formulation of any word size while serial FIFO structure provide a rapid and simple like other structure.
- Interchip communication protocol problem is solved by FIFO.

Disadvantages

- Noise in signals is caused by switching operation and alteration of memory content.

- Synchronization circuit is required to synchronize WRITE CLOCK and READ CLOCK

Thank you…!