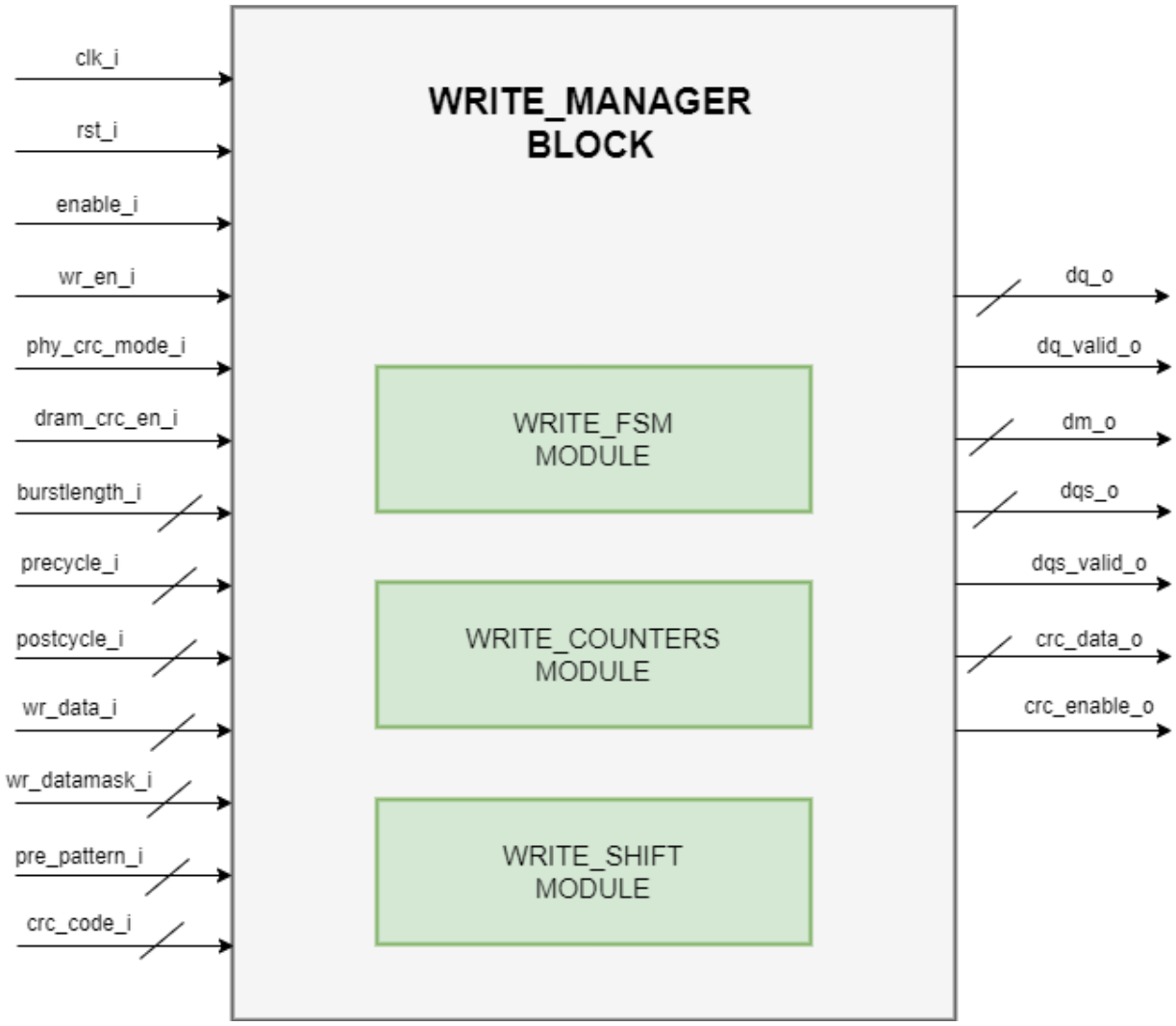
Write data Block

Sponsored by: Si-Vision

Contents

1. Block diagram.



2. Block functionality:

This block is responsible for multiple functions:

1- Checking for CRC generation through 2 input signals phy_crc_mode_i and dram_crc_en_i.
   - **(phy_crc_mode_i = 0 and** dram_crc_en_i **= 1)** → MC crc support.
   - **(phy_crc_mode_i = 1 and** dram_crc_en_i **= 1)** → phy crc support.
   - **(phy_crc_mode_i = 0 and** dram_crc_en_i **= 0)** → mask operation.

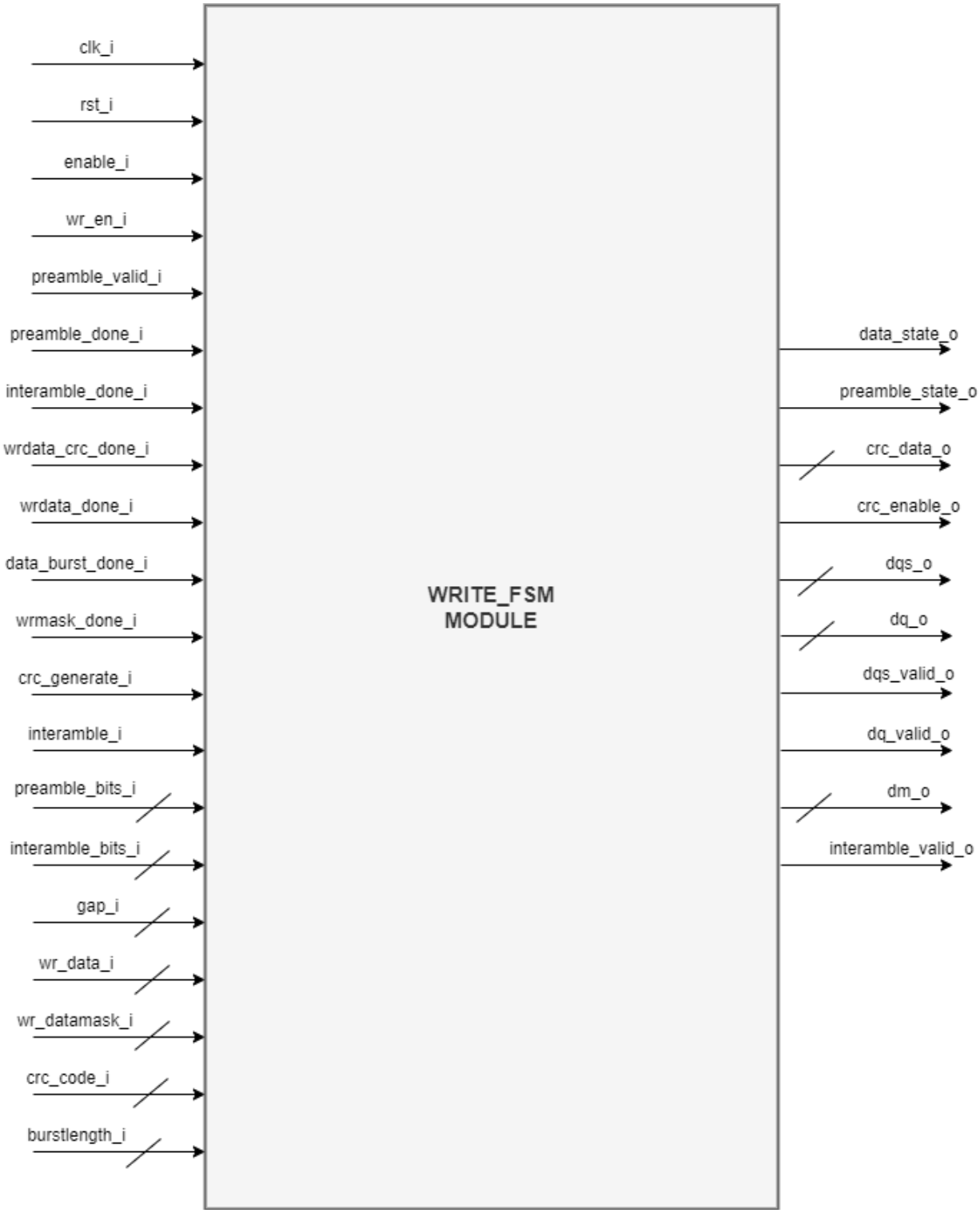
2- According to different cases , the block will perform different operations to send write data on DQ bus to DRAM interface.
   a- Phy CRC support : in this case the block is responsible for
      - sending write data on DQ bus to DRAM interface with DQ valid signal .
      - sending CRC_enable and write data to CRC block to generate crc code.
      - Taking crc code generated from CRC block and sending it on DQ bus after data.
      - In case burst_length = 8 , this block is responsible for completing the rest of data with ones and sending it on DQ bus and to CRC block .
   b- MC CRC support or Mask operatin : in this case the block is responsible for
      - sending write data on DQ bus to DRAM interface with DQ valid signal .

3- Shifting pre_ pattern_i (2 bits for each clock cycle) and sending it on DQS bus when write enable is activated and before coming of data with DQS valid signal.
4- while sending write data on DQ bus , DQS will be phy clock (DQS =10) with DQS valid signal.
5- Calculating gap between different write operations (no of cycles write enable is low) .
6- checking if ther is interamble through 2 input signals pre_ cycle_i and post_cycle_i (From command address block) when (gap < pre_ cycle_i + post_cycle_i) there will be interamble and According to gap value , interamble pattern will be detected from pre_ pattern_i .
7- After whole data is sent on DQ bus , interamble pattern will be shifted (2 bits for each clock cycle) and sent on DQS bus (for no of cycles = gap value) with DQS valid signal.
8- if there is no interamble , After whole data is sent on DQ bus , postamble pattern will be shifted (2 bits for each clock cycle) and sent on DQS bus (for no of cycles = post cycle) with DQS valid signal.

For performing this functions , this block will be divided into 3 modules as the following.

1- Write_FSM module:

1. Block Diagram



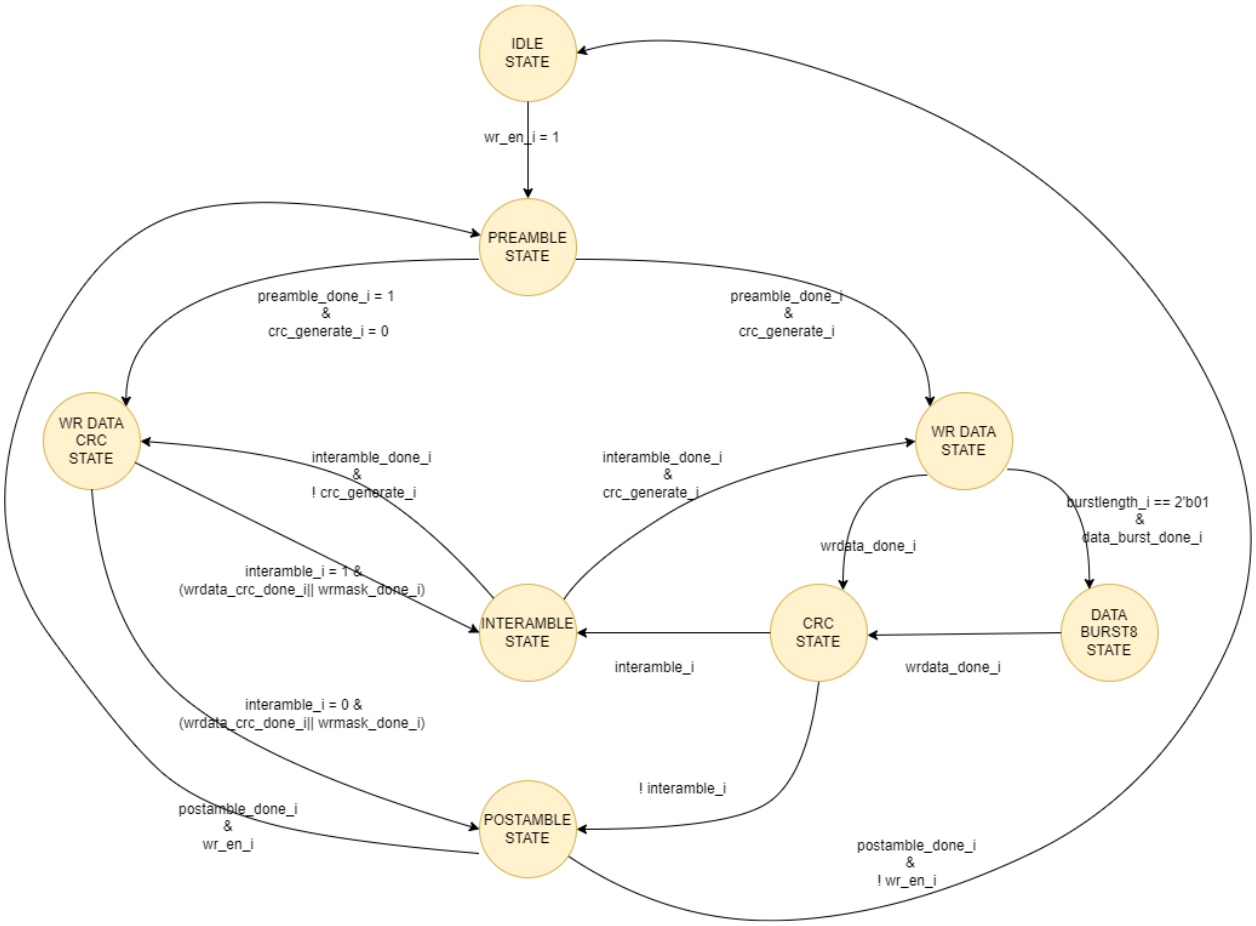
2. I/O ports description:

port	Direction	Size	Description
clk_i	input	1bit	clock signal. (From system)
rst_i	input	1 bit	active low asynchronous reset. (From system)
wr_en_i	input	1 bit	write enable signal. (From frequency Ratio)
enable_i	input	1 bit	signal to enable FSM. (From system)
preamble_done_i	input	1 bit	signal indicates that whole preamble pattern is sent on DQS bus. (From write_counter)
postamble_done_i	input	1 bit	signal represents that whole postamble pattern is sent on DQS bus. (From write_counter)
interamble_done_i	input	1 bit	signal represents that whole interamble pattern is sent on DQS bus. (From write_counter)
preamble_valid_i	input	1 bit	valid signal represents that correct preamble pattern is sent on DQS bus. (From write_counter)
wrdata_crc_done_i	input	1 bit	signal represents that whole data is sent on DQ bus (MC CRC support). (From write_counter)
wrdata_done_i	input	1 bit	signal represents that whole data is sent on DQ bus (phy crc support). (From write_counter)
data_burst_done_i	input	1 bit	signal represents that whole data is sent on DQ bus (burst length = 8). (From write_counter)
wrmask_done_i	input	1 bit	input signal that indicates whole data is sent on DQ bus (data mask). (From write_counter)
crc_generate_i	input	1 bit	signal indicates that if phy will generate crc or not. (From write_counter)
interamble_i	input	1 bit	signal indicates that if there is interamble or not. (From write_counter)

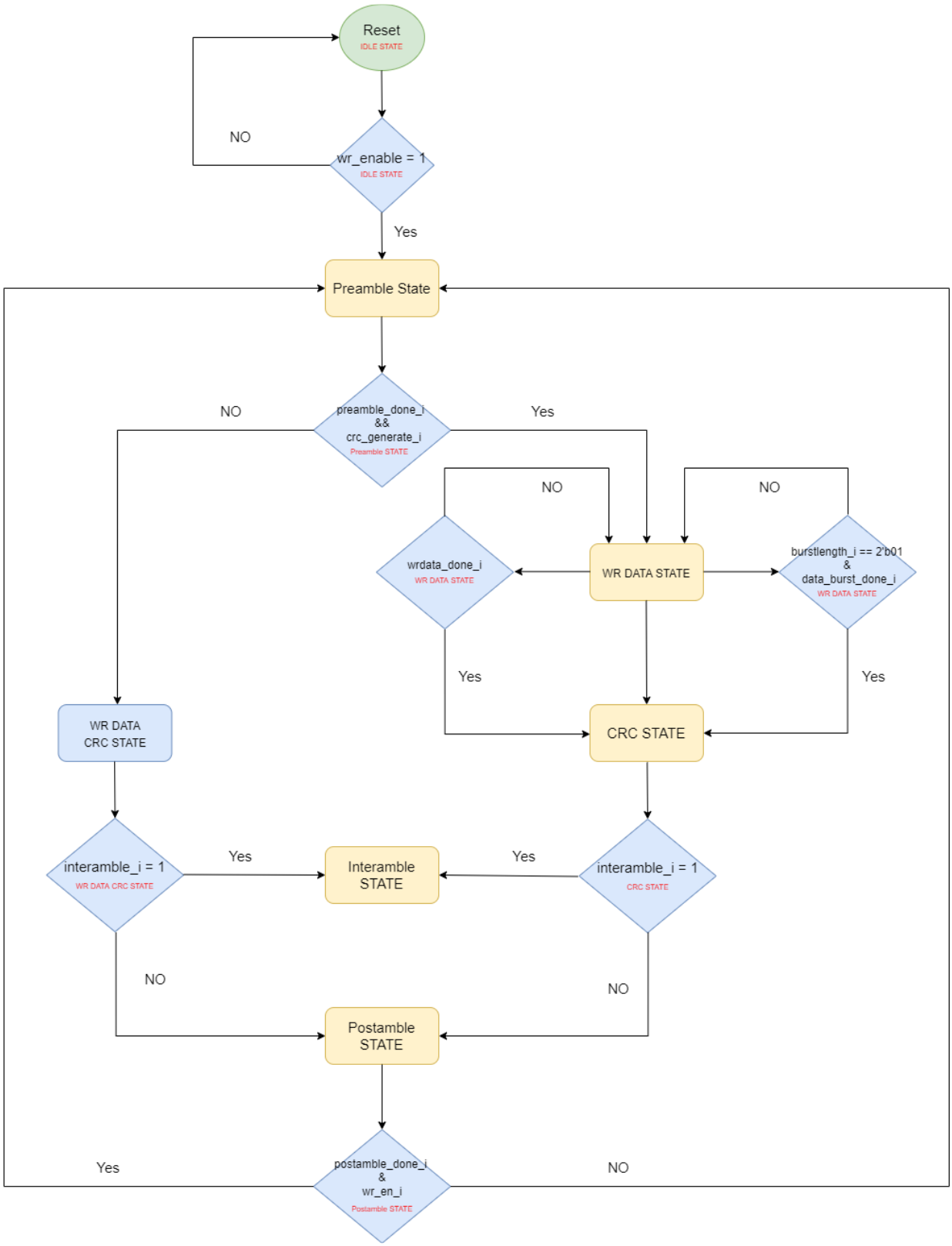
preamble_bits_i	input	2 bits	preamble bits result from shifting preamble pattern to be sent on DQS bus. (From write_shift)
interamble_bits_i	input	2 bits	bus signal results from shifting interamble pattern. (From write_shift)
gap_i	input	4 bits	signal detects number of cycles at which write enable is low. (From write_shift)
Wr_data_i	input	2*DRAM_SIZE	input bus signal holds wr data (From freq ratio block)
Wr_datamask_i	input	DRAM_SIZE/4	input bus signal hold wr data mask (From freq ratio block)
crc_code_i	input	2*DRAM_SIZE	input bus signal holds crc code (From crc block).
burstlength_i	input	2 bits	input bus signal holds the code of the burst length (From command_address block).
crc_data_o	output	2*DRAM_SIZE	bus signal holds data to CRC block to generate CRC code for it.
crc_enable_o	output	1 bit	enable signal to enable CRC block.
DQS_o	output	2 bits	output bus signal holds different patterns of data strobe to be sent to DRAM interface.
DQ_o	output	2*DRAM_SIZE	output bus signal holds write data to be sent to DRAM interface.
DQS_valid_o	output	1 bit	signal indicates that correct pattern of data strobe is sent to DRAM interface when activated high.
DQ_valid_o	output	1 bit	signal indicates that write data is sent to DRAM interface when activated high.
DM_o	output	DRAM_SIZE/4	output bus signal holds write data mask to be sent to DRAM interface.
interamble_valid_o	output	1 bit	signal indicates that that controller in interamble_state. (To write_counter and write_shift)
preamble_state_o	output	1 bit	signal indicates that controller in preamble_state. (To write_counter)
data_state_o	output	1 bit	output signal indicates that controller in write data_states. (To write_counter)

3. Block implantation:

State diagram:



Algorithm Chart



Operation

The write_FSM block consist of 7 states:

1- idle:

- no operation in this state.
- When wr_en_i is activated high, we will move to preamble_state.

2- Preamble:

- in this state, preamble pattern will be sent on DQS bus.
- After whole preamble_pattern is sent on DQS bus, we will move to wr_data state (crc_generate = 1) or wr_data_crc (crc_generate = 0).
  - DQS_valid will be high when correct preamble_pattern is sent on DQS.

3- wr_data_crc (MC_crc_support or mask operation):

- In this state, wr_data will be sent on DQ bus and wr_data_mask will be sent on DM in case of mask operation.
- DQ_valid will be high when data is sent on DQ bus.
- DQS will be the PHY clock (DQS = 10) and DQS_valid will be high.
- After whole data is sent on DQ bus, we will move to interamble_state if there is interamble (interamble_i is activated high) otherwise, we will move to postamble_state.

4- Wr_data (phy_crc_support)

- In this state, wr_data will be sent on DQ bus.
- Wr_data and crc_enable will be sent to crc_block to generate crc_code.
- DQ_valid will be high when data is sent on DQ bus.
- DQS will be the PHY clock (DQS = 10) and DQS_valid will be high.
- After whole data is sent on DQ bus, we will move to crc state to take crc code generated from CRC block and send it on DQ bus.
- After whole data is sent on DQ bus in case burst_length = 8, we will move first to data_burst8 state to complete rest of wr_data with ones then move to crc state to generate crc code.

5- data_burst8 (burst-length = 8)

- In this state, wr_data will be completed by ones and sent it on DQ bus.
- The rest of Wr_data and crc_enable will be sent to crc_block to generate crc_code.
- DQ_valid will be high when completing data with ones on DQ bus.
- DQS will be the PHY clock (DQS = 10) and DQS_valid will be high.
- After whole data is sent on DQ bus, we will move to crc state to take crc code generated from CRC block and send it on DQ bus.

6- CRC

- In this state, crc_code will be sent on DQ bus.
- DQ_valid will be high when crc_code is sent on DQ bus.
- DQS will be the PHY clock (DQS = 10) and DQS_valid will be high.
- After crc_code is sent on DQ bus, we will move to interamble_state if there is interamble (interamble_i is activated high) otherwise, we will move to postamble_state.

7- postamble (interamble_i =0)

- In this state, postamble_pattern will be sent on DQS bus.
- DQS_valid will be high when postamble_pattern is sent on DQS.
- After whole pattern is sent on DQS bus, we will move to preamble_state if wr_en is activated high to start another operation otherwise, we will move to idle.

8- Interamble (interamble_i =1)

- In this state, interamble_pattern according to gap value will be sent on DQS bus.
- DQS_valid will be high when interamble_pattern is sent on DQS.
- After whole pattern is sent on DQS bus, we will move to wr_data state if crc_generate = 1 or wr_data_crc if crc_generate = 0.

2- Write_shift module

1. Block Diagram

2. I/O ports description:

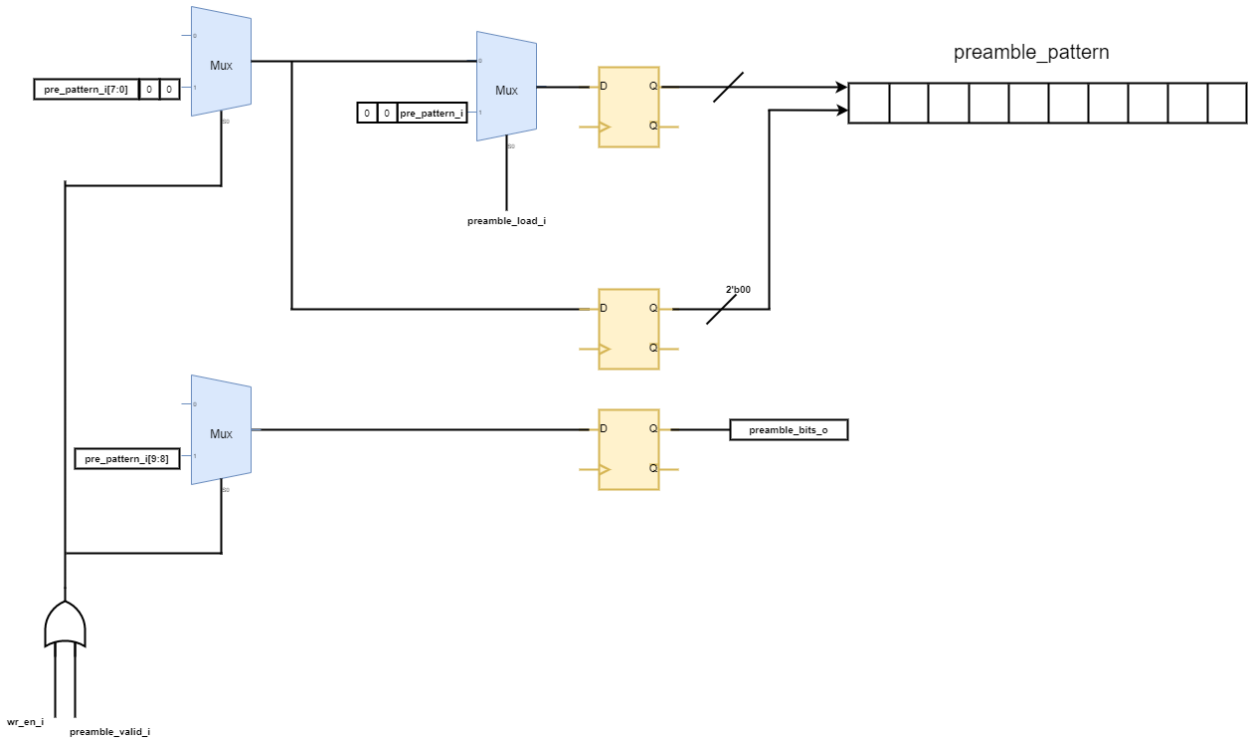
port	Direction	Size	Description
clk_i	input	1bit	clock signal. (From system)
rst_i	input	1 bit	active low asynchronous reset. (From system)
wr_en_i	input	1 bit	write enable signal. (From frequency Ratio)
pre_pattern_i	input	8 bit	bus signal holds the preamble pattern which will be shifted to be sent on DQS bus. (From command address)
interamble_valid_i	input	1 bit	signal indicates that that controller in interamble_state. (From write_fsm)
interamble_shift_i	input	3 bit	bus signal for shifting the interamble pattern to be out on the DQS bus. (From write_counter)
Preamble_valid_i	input	1 bit	valid signal represents that correct preamble pattern is sent on DQS bus. (From write_counter)
Preamble_load_i	input	1 bit	signal to load preamble pattern in shift register when activated high. (From write_counter)
gap_burst_eight_i	input	1 bit	signal is activated high when burst length = 8 and phy crc support to decrement gap value by 4. (From write_counter)
interamble_bits_o	output	2 bits	bus signal results from shifting interamble pattern to be sent on DQS bus. (To write_fsm)
Preamble_bits_o	output	2 bits	bus signal results from shifting preamble pattern to be sent on DQS bus. (To write_fsm)
gap_o	output	4 bits	bus signal represents the number of cycles at which write enable is low. (To write_fsm and write_counter)

3. Block implantation:

This block is responsible for 3 main operations:

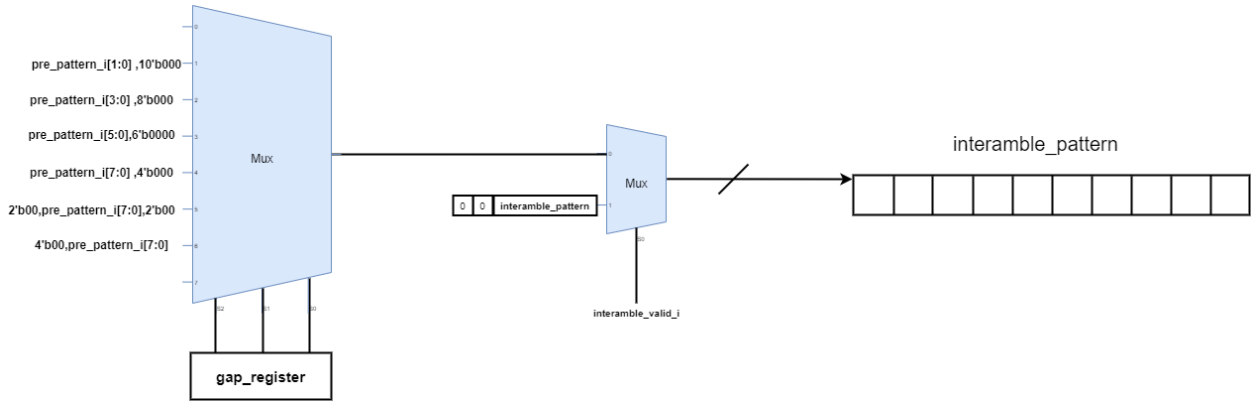
1- Shifting preamble pattern:

- When preamble_load_i is activated high the preamble_pattern will be loaded in to shift_ register and ready to be shifted.
- When wr_en_i is activated high, preamble_pattern will be shifted for 4 clock cycles (2 bits for each cycle) and will be sent on DQS bus.
- When preamble_valid_i is activated high it means that the correct pattern is sent, and the shifting operation will be continued.
- After wr_en_i and preamble_valid_i become low, the shifting operation will be stopped and preamble_pattern will be loaded into shift_register again to be used it in another operation.



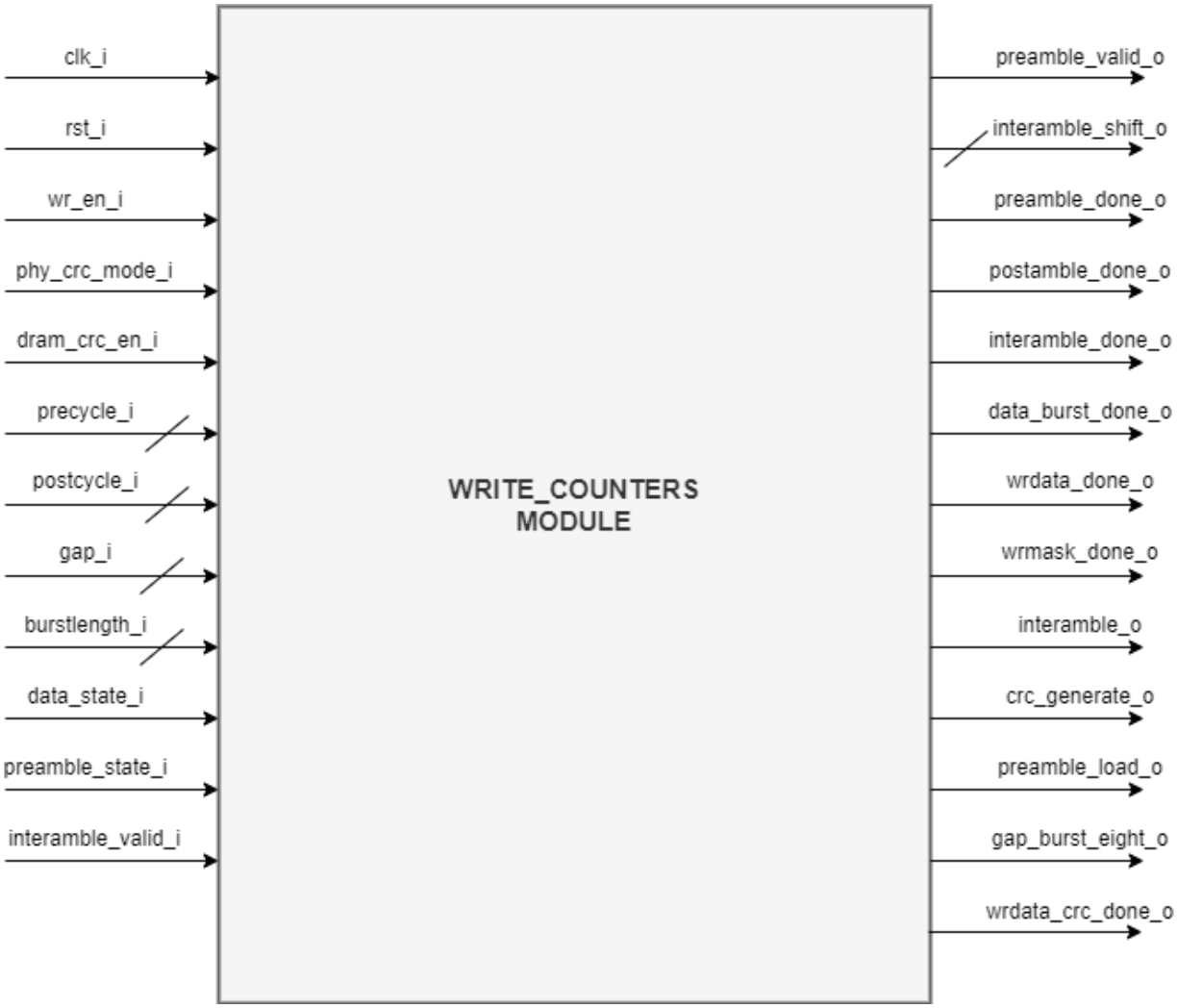
2- Calculating gap (no of cycles at which write enable is low):
    - When wr_en_i becomes low, counter will start to count, the value of counter will be stored in gap_register.
    - After wr_en_i becomes high, we will reset the counter and check the value stored in register to detect the correct interamble_pattern according to the gap value.

3- Shifting interamble pattern:
    - According to the gap value, we will detect the interamble pattern from **pre_pattern_i** and store it in a shift_ register and it will be ready to be shifted.
    - When interamble_valid_i is activated high, interamble pattern will be shifted according to gap value (2 bits for each cycle) and will be sent on DQS bus.
    - interamble_shift_i will take different value for each clock cycle during interamble state so the shifting operation will be continued.
    - After interamble_valid_i becomes low, it means that interamble state is finished and shifting operation will be stopped.



3- Write_counters module

1. Block Diagram



2. I/O ports description:

Port	Direction	Size	Description
clk_i	Input	1 bit	input clock signal. (From system)
rst_i	Input	1 bit	input active low asynchronous reset. (From system).
wr_en_i	Input	1 bit	input write enable signal.
phy_crc_mode_i	Input	1 bit	signal indicates either PHY or MC will generate the crc code for the data. (From Regfile)
dram_crc_en_i	Input	1 bit	Signal indicates whether the dram wants to receive a crc code for the data or not. (From Regfile)
precycle_i	Input	3 bit	Input represents the number of preamble cycle. (From command address block)
postcycle_i	Input	2 bit	Input represents the number of postamble cycles. (From command address block).
gap_i	Input	4 bit	Input represents the gap value. (From write_shift block)
burstlength_i	Input	2 bit	Signal represents the burst length of the data. (From command address block)
data_state_i	Input	1 bit	signal indicates that the controller is in the data states and used to start the write data counter and to stop the preamble counter. (From write_FSM block)
preamble_state_i	Input	1 bit	signal indicates that controller is in preamble_state. (From write_FSM block)
interamble_valid_i	Input	1 bit	signal indicates that that controller is in interamble_state and interamble bits is sent on DQS bus. (From write_FSM block)
preamble_valid_o	Output	1 bit	valid signal represents that correct preamble pattern is sent on DQS bus. (To write_FSM and write_shift block)
interamble_shift_o	Output	3 bit	bus signal that takes different values in interamble state for shifting the interamble pattern to be out on the DQS bus. (To write_shift)
preamble_done_o	Output	1 bit	signal indicates that whole preamble pattern is sent on DQS bus and preamble_state is finished. (To write_FSM)
postamble_done_o	Output	1 bit	signal represents that whole postamble pattern is sent on DQS bus and postamble_state is finished. (To write_FSM)
interamble_done_o	Output	1 bit	signal represents that whole interamble pattern is

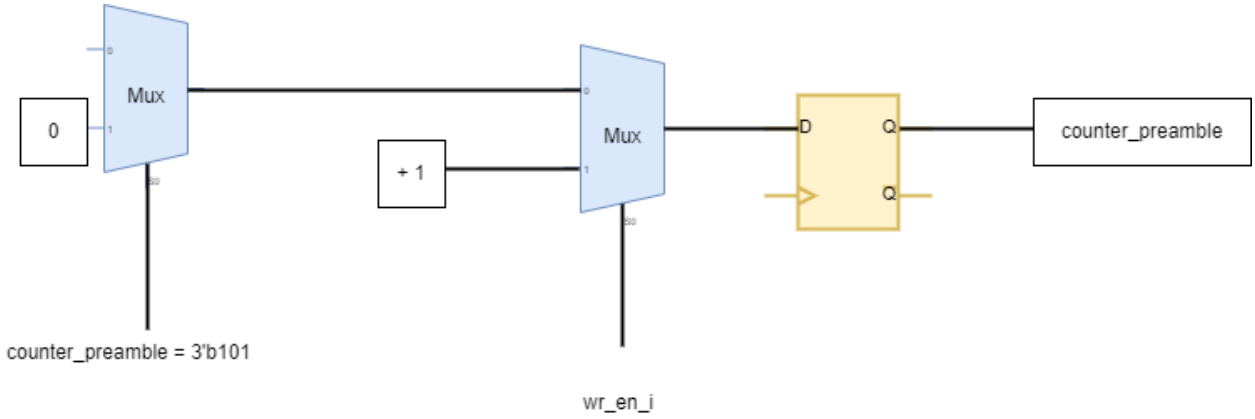
			sent on DQS bus and interamble_state is finished. (To write_FSM)
data_burst_done_o	Output	1 bit	signal represents that whole data of BL8 is sent on DQ bus. (To write_FSM)
wrdata_done_o	Output	1 bit	signal represents that whole data is sent on DQ bus. (To write_FSM)
wrmask_done_o	Output	1 bit	signal that indicates that whole data mask is sent on DQ bus. (To write_FSM)
interamble_o	Output	1 bit	signal indicates that if there is interamble or not. (To write_FSM)
crc_generate_o	Output	1 bit	signal indicates that if phy will generate crc or not. (To write_FSM)
preamble_load_o	Output	1 bit	Signal used to load preamble pattern in shift register when activated high. (To write_shift)
gap_burst_eight_o	Output	1 bit	signal used to decrement gap value by 4. (To write_shift)
wrdata_crc_done_o	Output	1 bit	signal represents that whole data is sent on DQ bus (MC CRC support). (To write_FSM)

3. Block implantation:

- Write_counters module is responsible for many functions but its main one is outputting done signals which will be used in transitions between different states in Controller module. Other marginal operations can be specified in the following outlines:
    - Determining either if there will be an interamble or not by comparing the precycle added to the postcycle with the gap value.
    - Determining whether there will be a CRC code generation and transmission to the DRAM or not.
    - Adjusting the DQS valid during preamble strobe by outputting a signal works as a valid signal to the controller to start output the DQS valid in preamble state.

- This block consists of 3 counters each one is responsible for multiple operation

    - **Counter1: Preamble counter**
        - Preamble counter works simultaneously with write enable signal independent on the state.
        - Activating write enable signal means there is a data coming after 5 cycles. So, preamble counter start counting
            - when the counter reaches (5 – precycle) → preamble valid signal is activated to tell the controller to start the DQS valid signal with the pre DQS strobe.
            - when the counter reaches five, preamble done signal is activated telling the controller to move from preamble state.
        - After counting five high write enable cycles, this counter will be initiated to zero to be ready for another operation.



- **Counter2: Write Data counter**
  - Unlike preamble counter, write data counter only counts in data state especially when write enable is de-activated.
  - After 5 cycles from de-activating write enable in data state, the full data burst will finish, and a done signal will be activated high telling the controller to move from data state. Accordingly, there are different cases:
    - Case (1): PHY will generate a CRC to a data with burst length = 16.

      In this case, write data done signal will be activated after five cycles from disabling the write enable telling the controller to move to crc state.

    - Case (2): PHY will generate a CRC to a data with burst length = 8.

      In this case, data burst done signal will be activated after four cycles telling the controller to move to data_burst_8 state and after another four cycles, write data done will be activated to move from this state to crc state.

    - Case (3): MC generates a CRC to a data with burst length = 16.

      In this case, write data crc done signal will be activated after five cycles plus one more cycle from disabling the write enable telling the controller to move from write data crc state.

      The delay of the write data crc done signal with one more cycle to postpone this state with one more cycle to allow the CRC code coming from MC to be sent on DQ bus.

    - Case (4): MC generates a CRC to a data with burst length = 8.

      MC will complete the burst with 1's and generate a CRC for the full data and send the full burst appended to it the CRC.

      In this case, write data counter works exactly like case (3).

    - Case (5): Data mask with burst length = 16.

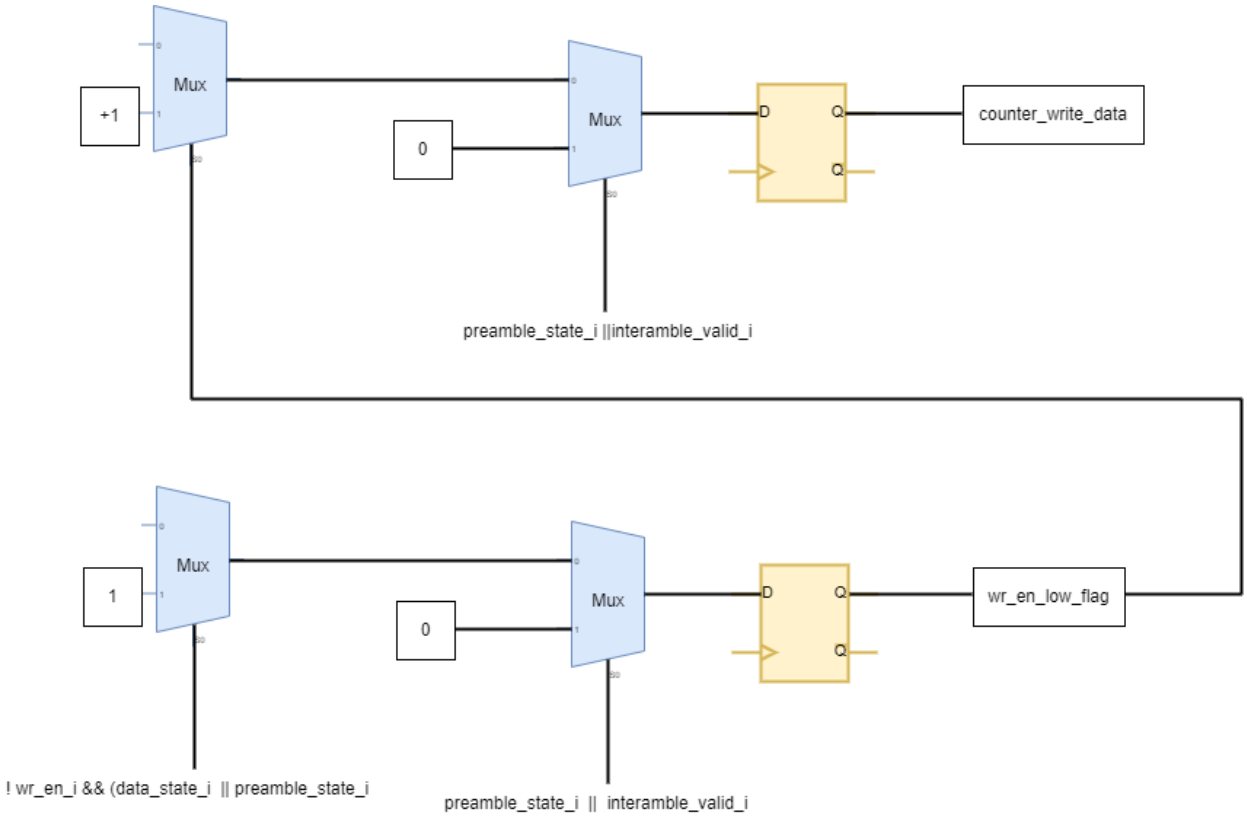
      In this case, write mask done signal will be activated after five cycles from disabling the write enable telling the controller to move from data state.

- Case (6): Data mask with burst length = 8.

  In this case, write mask done signal will be activated after four data cycles without completing the burst with 1's.

- wr_en_low_flag is activated high when write enable signal is de-activated in data state to allow write data counter to count on wr_en_low_flag. The reason of using this flag and not counting on the de-activation of write enable signal is to allow the counter to count even in the case of multiple write operations.

- At the end of data state, this counter will be initiated to zero to be ready for another operation.



- **Counter3: inter post counter**
  - This counter counts only during interamble state and postamble state.
  - Postamble and Interamble states cannot occur at same time. This illustrates the reason for using one counter for them both.
  - According to the state, we will have 2 different cases
    - Case (1): Postamble State.

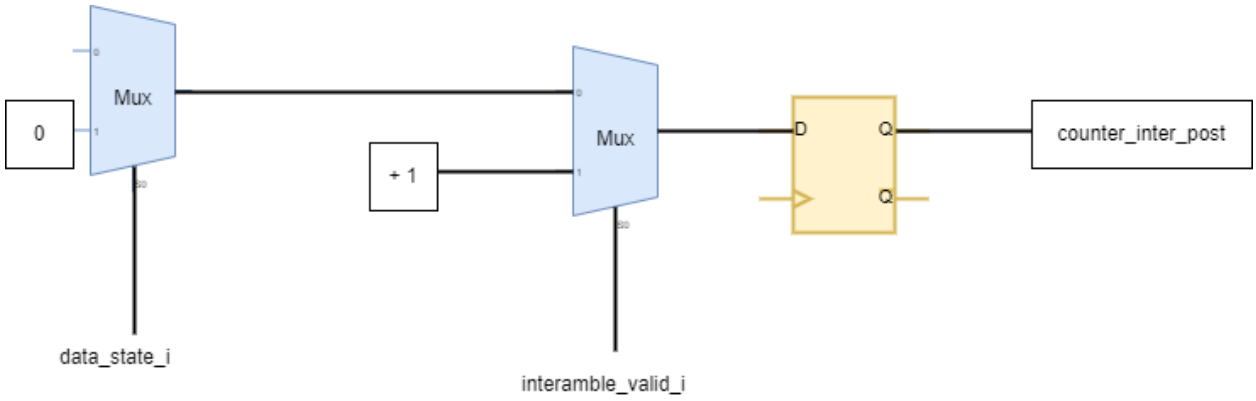
      In this case, postamble done signal will be activated when inter post counter reaches the value of postcycle.

    - Case (2): Interamble State.

      Interamble means that there is another coming data too nearly that cannot have its own preamble. This is determined by comparing the gap value with precycle needed for coming data and postcycle required for previous data.

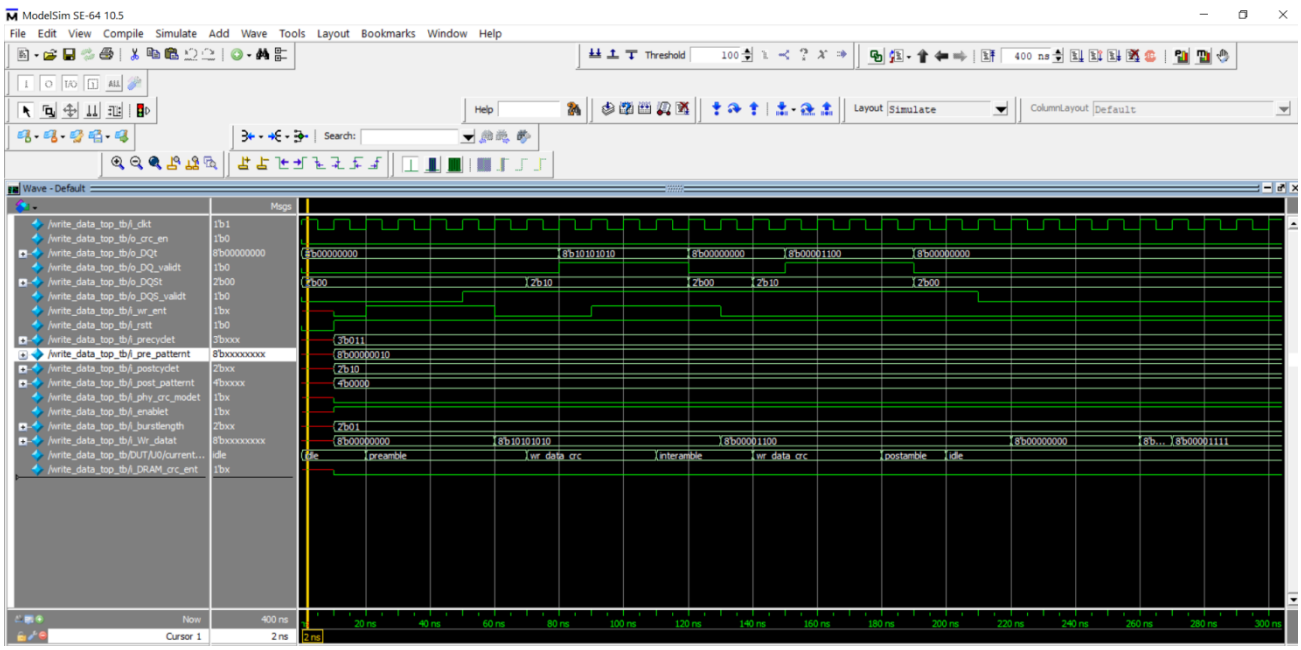
      Accordingly, write fsm module should remain in interamble state for a number of cycles equal to the gap value. Therefore, interamble done signal will be activated when inter post counter reaches gap value.

  - As soon as moving from postamble or interamble states, interpost counter is initiated to zero to be ready for another operation.

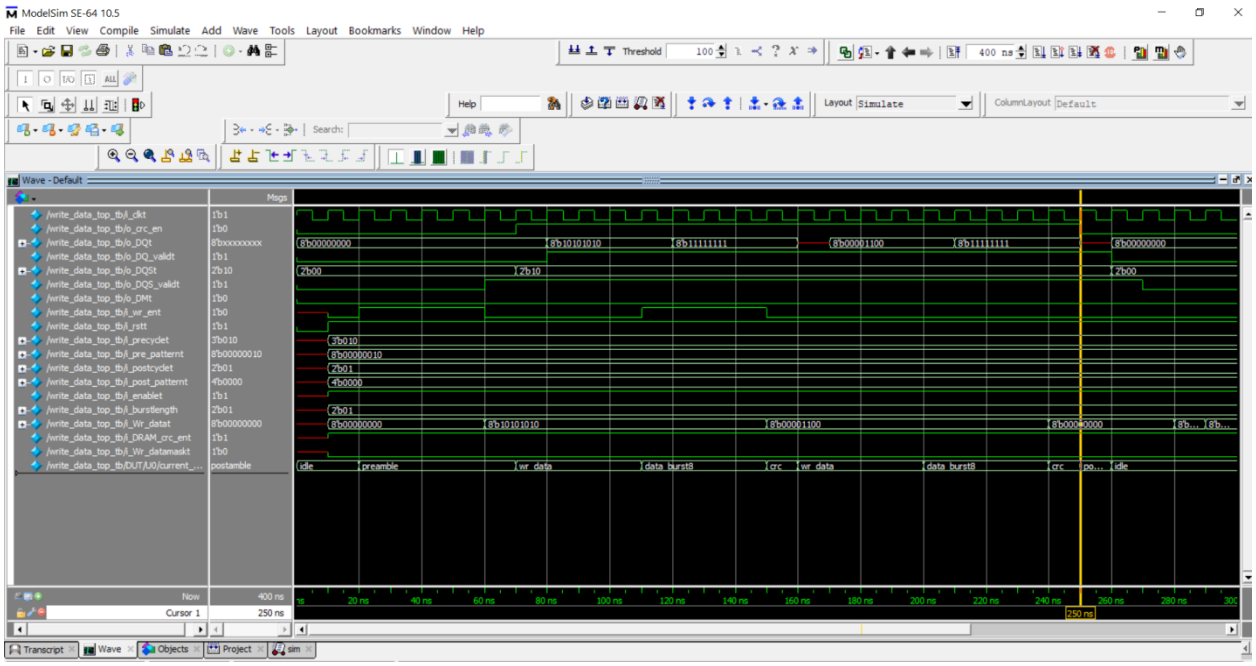


4. Simulation Results:

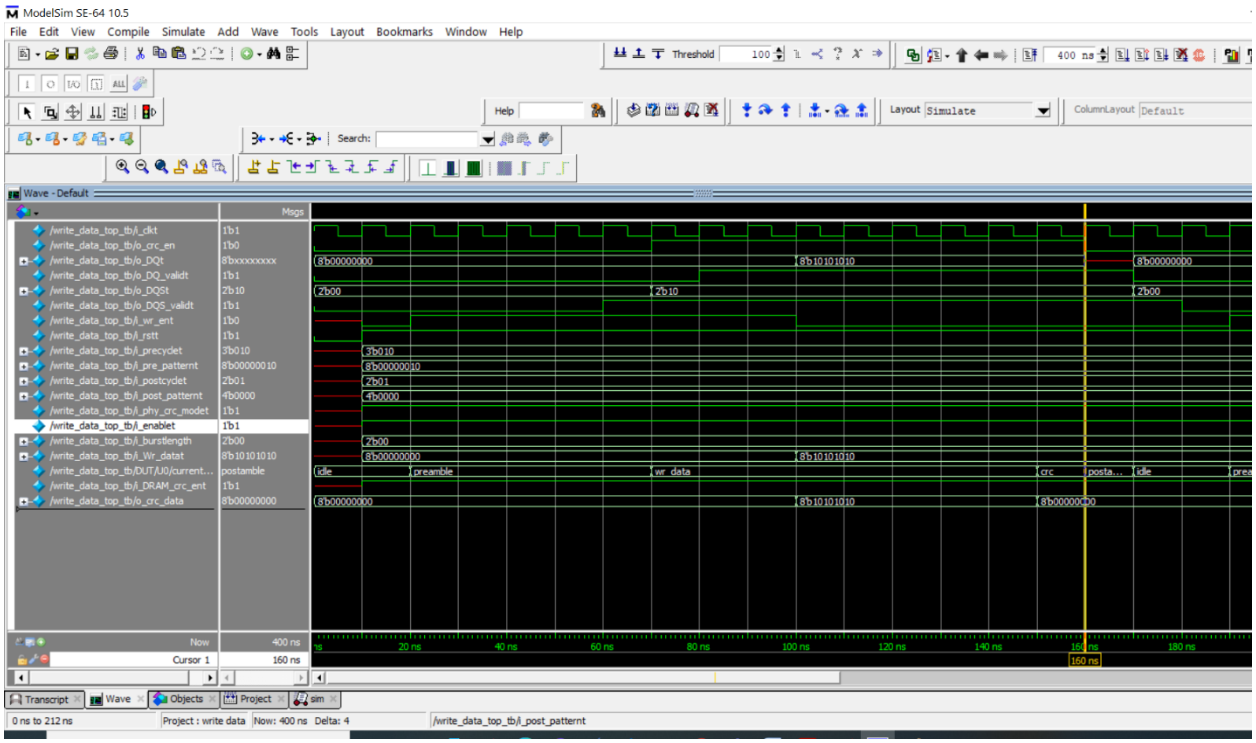
1- Burst length 16 , MC crc support



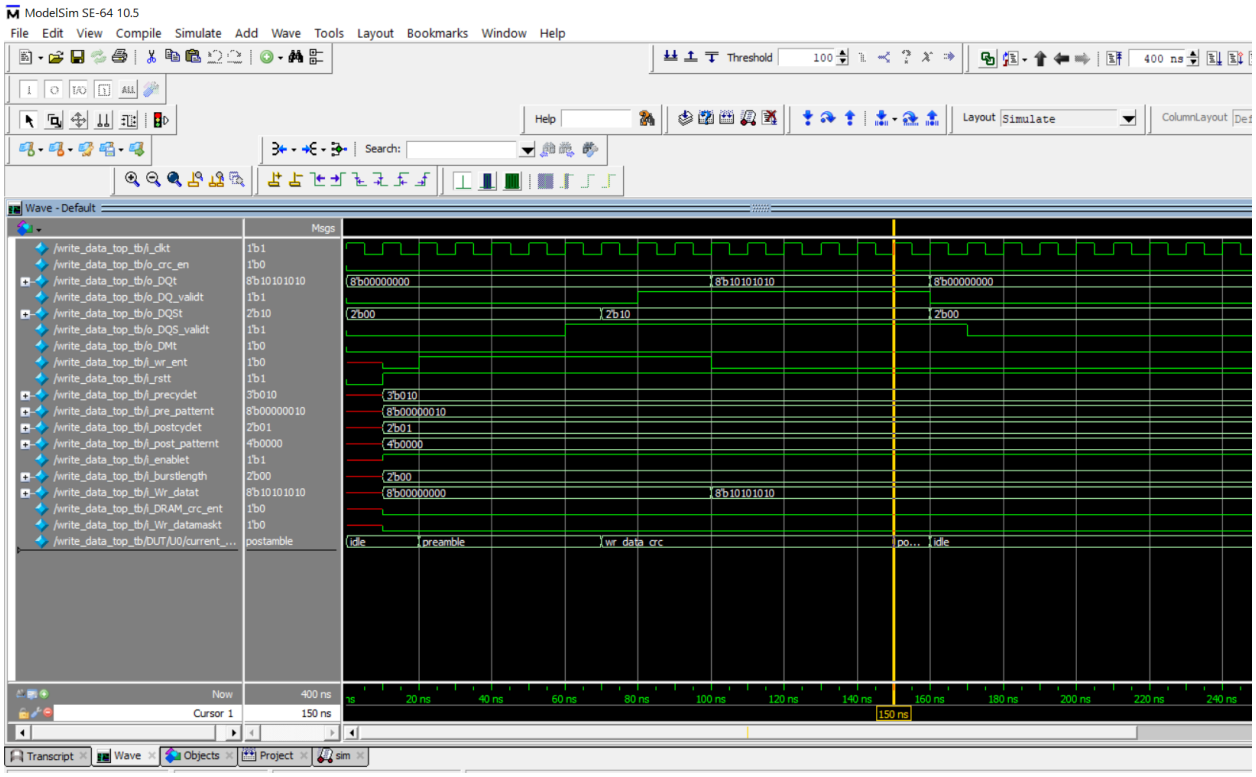
2- Burst length = 8 ,phy crc support



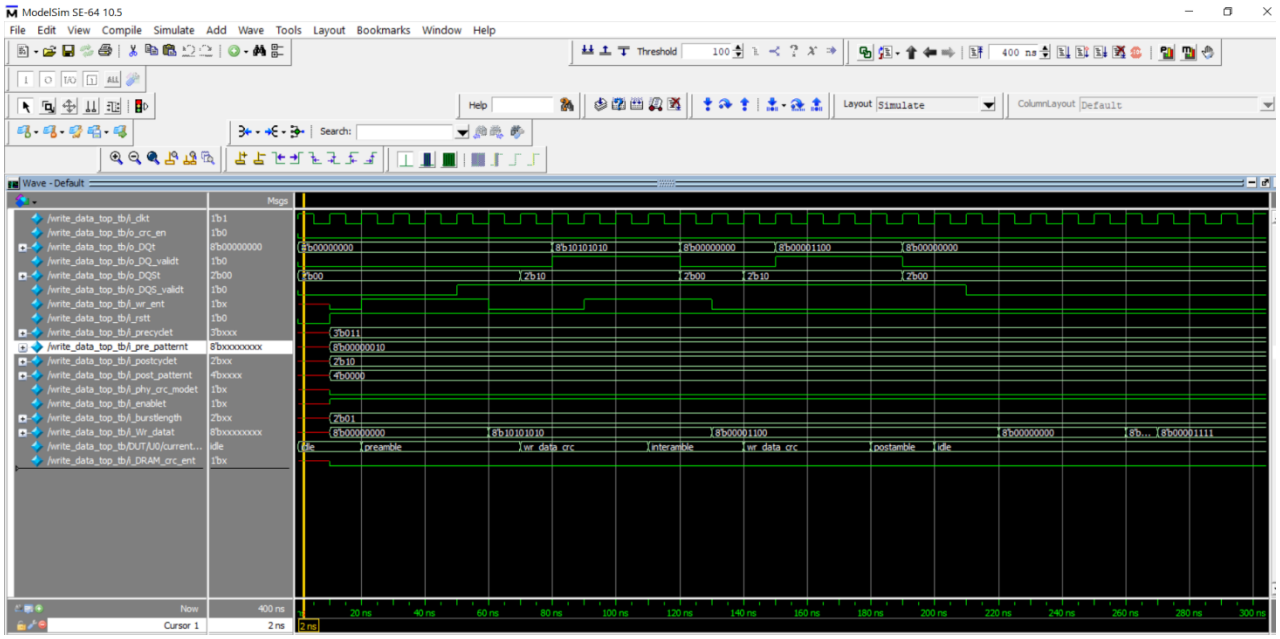
3- Burst length =16 .phy crc support



4- Burst length =16 .Mask operation



5- Burst length = 8 .Mask operation



5. Synthesis results

1. Area report

```
1
2 ****************************************
3 Report : area
4 Design : write_manager
5 Version: K-2015.06
6 Date   : Sun Apr 17 10:55:50 2022
7 ****************************************
8
9 Library(s) Used:
10
11    scmetro_tsmc_cl013g_rvt_ss_1p08v_125c (File: /home/IC/tsmc_fb_cl013g_sc/aci/sc-m/synopsys/scmetro_tsmc_cl013g_rvt_ss_1p08v_125c.db)
12
13 Number of ports:                          190
14 Number of nets:                           637
15 Number of cells:                          474
16 Number of combinational cells:            408
17 Number of sequential cells:                63
18 Number of macros/black boxes:               0
19 Number of buf/inv:                         95
20 Number of references:                       3
21
22 Combinational area:               3427.727142
23 Buf/Inv area:                      575.406309
24 Noncombinational area:            1672.090689
25 Macro/Black Box area:                0.000000
26 Net Interconnect area:          165480.824341
27
28 Total cell area:                  5099.817831
29 Total area:                     170580.642172
30 1
```

2. Power report

```
19
20 Operating Conditions: scmetro_tsmc_cl013g_rvt_ss_1p08v_125c   Library: scmetro_tsmc_cl013g_rvt_ss_1p08v_125c
21 Wire Load Model Mode: top
22
23 Design        Wire Load Model            Library
24 ------------------------------------------------
25 write_manager          tsmc13_wl30       scmetro_tsmc_cl013g_rvt_ss_1p08v_125c
26
27
28 Global Operating Voltage = 1.08
29 Power-specific unit information :
30     Voltage Units = 1V
31     Capacitance Units = 1.000000pf
32     Time Units = 1ns
33     Dynamic Power Units = 1mW    (derived from V,C,T units)
34     Leakage Power Units = 1pW
35
36
37   Cell Internal Power  = 300.8719 uW   (54%)
38   Net Switching Power  = 255.0824 uW   (46%)
39                          ---------
40 Total Dynamic Power    = 555.9543 uW  (100%)
41
42 Cell Leakage Power     =   3.9562 uW
43
44
45                 Internal         Switching           Leakage            Total
46 Power Group      Power            Power               Power              Power   (   %    )  Attrs
47 --------------------------------------------------------------------------------------------------
48 io_pad             0.0000           0.0000              0.0000           0.0000  (   0.00%)
49 memory             0.0000           0.0000              0.0000           0.0000  (   0.00%)
50 black_box          0.0000           0.0000              0.0000           0.0000  (   0.00%)
51 clock_network      0.0000           0.0000              0.0000           0.0000  (   0.00%)
52 register           0.2533        4.0035e-02          8.5093e+05           0.2942  (  52.54%)
53 sequential     6.5090e-05           0.0000          2.6267e+04        9.1357e-05  (   0.02%)
54 combinational  4.7539e-02           0.2150          3.0790e+06           0.2657  (  47.45%)
55 --------------------------------------------------------------------------------------------------
56 Total              0.3009 mW         0.2551 mW     3.9562e+06 pW         0.5599 mW
57 1
```

3. Timing report

```
476  Path Group: i_clk
477  Path Type: max
478
479  Des/Clust/Port     Wire Load Model       Library
480  ------------------------------------------------
481  write_manager      tsmc13_wl30           scmetro_tsmc_cl013g_rvt_ss_1p08v_125c
482
483  Point                                                   Incr       Path
484  ---------------------------------------------------------------------------
485  clock i_clk (rise edge)                                 0.00       0.00
486  clock network delay (ideal)                             0.00       0.00
487  U0/current_state_reg[2]/CK (DFFRHQX8M)                  0.00       0.00 r
488  U0/current_state_reg[2]/Q (DFFRHQX8M)                   0.49       0.49 f
489  U0/U18/Y (INVX6M)                                       0.45       0.94 r
490  U0/U48/Y (OR2X2M)                                       0.58       1.52 r
491  U0/U4/Y (OAI21X8M)                                      0.23       1.74 f
492  U0/U26/Y (BUFX32M)                                      0.27       2.01 f
493  U0/o_interamble_valid (write_fsm)                       0.00       2.01 f
494  U1/i_interamble_valid (write_counters)                  0.00       2.01 f
495  U1/U61/Y (INVXLM)                                       0.50       2.52 r
496  U1/U3/Y (BUFX5M)                                        0.63       3.15 r
497  U1/U54/Y (NAND2X4M)                                     0.48       3.63 f
498  U1/U165/Y (OAI2BB2X1M)                                  0.73       4.36 r
499  U1/gap_value_reg[2]/D (DFFRQX2M)                        0.00       4.36 r
500  data arrival time                                                  4.36
501
502  clock i_clk (rise edge)                                 5.00       5.00
503  clock network delay (ideal)                             0.00       5.00
504  clock uncertainty                                      -0.20       4.80
505  U1/gap_value_reg[2]/CK (DFFRQX2M)                       0.00       4.80 r
506  library setup time                                     -0.41       4.39
507  data required time                                                 4.39
508  ---------------------------------------------------------------------------
509  data required time                                                 4.39
510  data arrival time                                                 -4.36
511  ---------------------------------------------------------------------------
512  slack (MET)                                                        0.03
513
514
515  Startpoint: U0/current_state_reg[0]
```