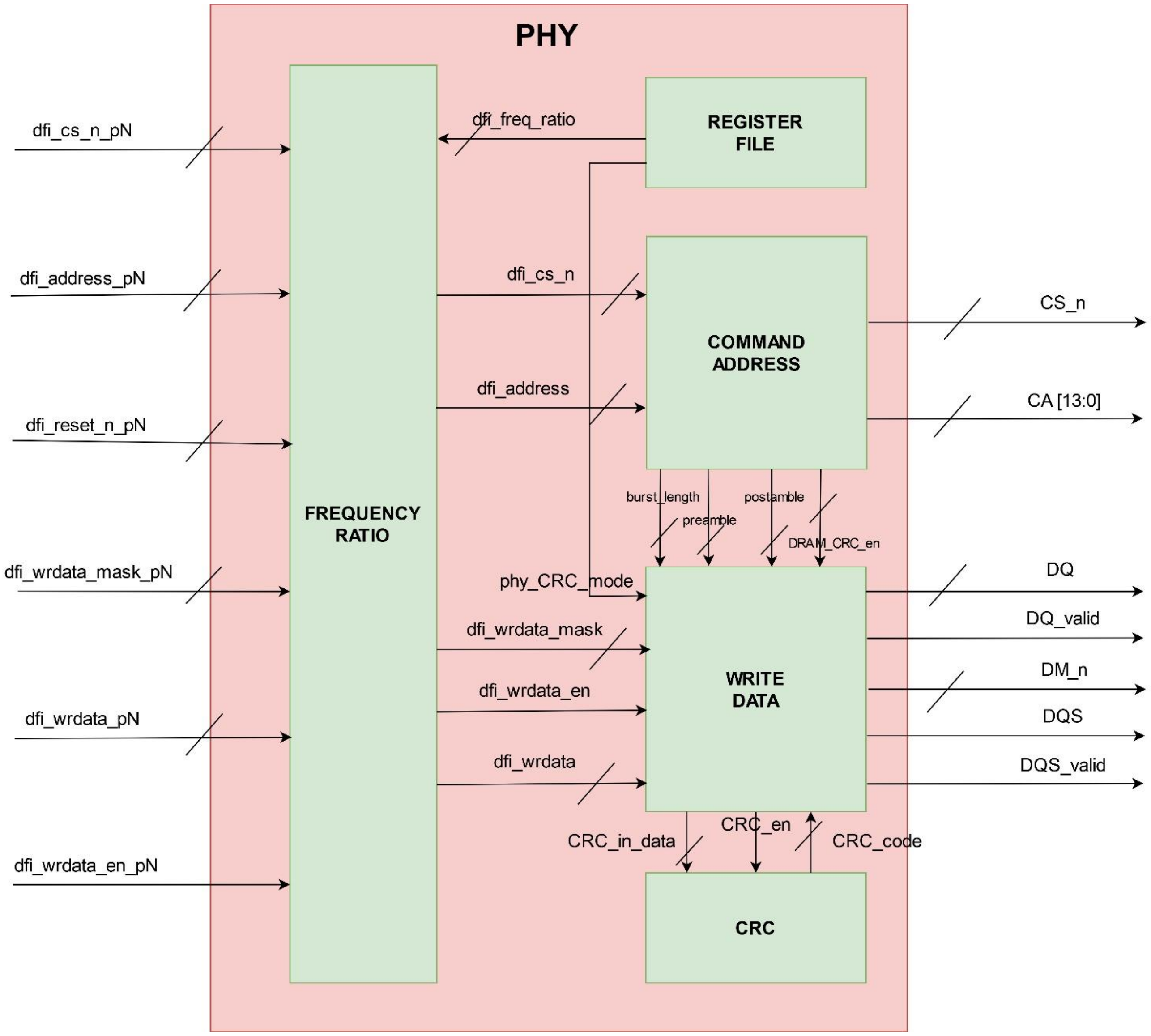
PHY Architecture Design

Sponsored by: Si-Vision

Contents

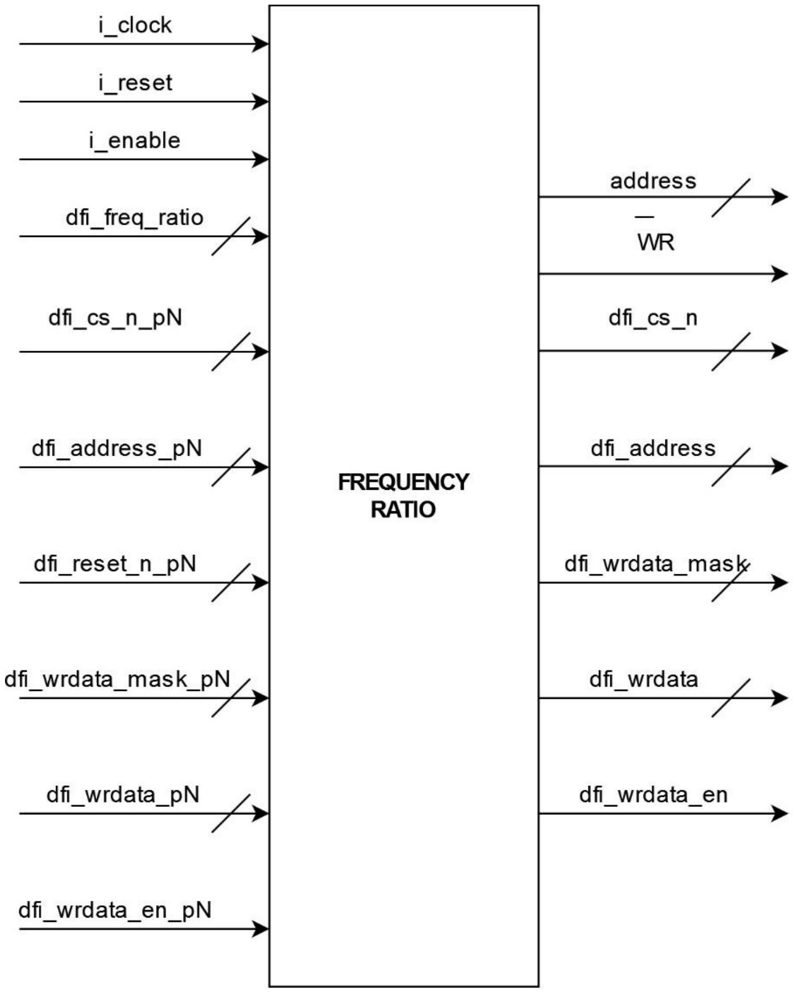
PHY Block Diagram



- ❖ PHY Block consists of 5 main blocks
  - FREQUENCY RATIO
  - COMMAND ADDRESS
  - REGISTER FILE
  - WRITE DATA
  - CRC

1- FREQUENCY RATIO

1. Block Diagram



2. Input and Output ports

Port	Type	Size
dfi_cs_n_pN	Input	NUM_RANK bits
dfi_address_pN	Input	14 bits
dfi_wrdata_pN	Input	2 * DEVICE_TYPE bits
dfi_wrdata_mask_pN	Input	dfi_wrdata_pN / 8 bits
dfi_reset_n_pN	Input	NUM_RANK bits
dfi_wrdata_enable_pN	Input	1 bit
dfi_freq_ratio	Input	3 bits
dfi_address	Output	14 bits
dfi_cs	Output	NUM_RANK bits
dfi_wrdata_enable	Output	1 bit
dfi_wrdata_mask	Output	dfi_wrdata_pN / 8 bits
dfi_wrdata	Output	2 * DEVICE_TYPE bits

- **Parameters Description**:
  - **NUM_RANK**: Number of DRAM Ranks.
  - **DEVICE_TYPE**: Either X4 or X8 or X16.

3. Block Functionality

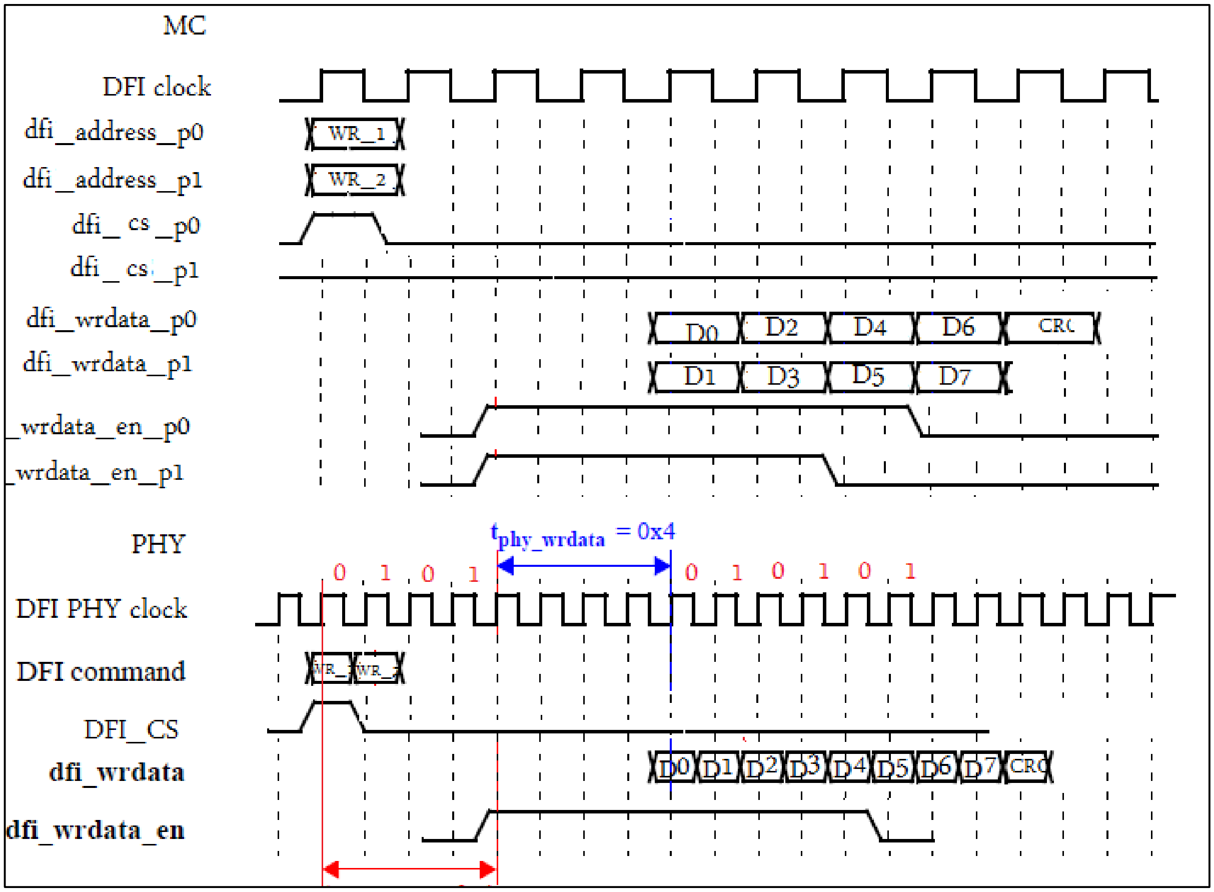
- FREQUENCY RATIO Block will be used to convert the signals from the Memory controller interface to the PHY interface according to the dfi_freq_ratio signal.
- The dfi_freq_ratio signal is determined during initialization and stored in the REGISTER FILE block.
- It will map the signals with multiple phases into one phase.
- This block works on the DFI PHY Clock.
- In case of the matched system, the DFI PHY clock will be the same speed as the MC clock. While, in frequency ratio system, the DFI PHY clock will be faster than the MC clock.

  If dfi_freq_ratio is : 000 the ratio is 1:1

  001 the ratio is 1:2

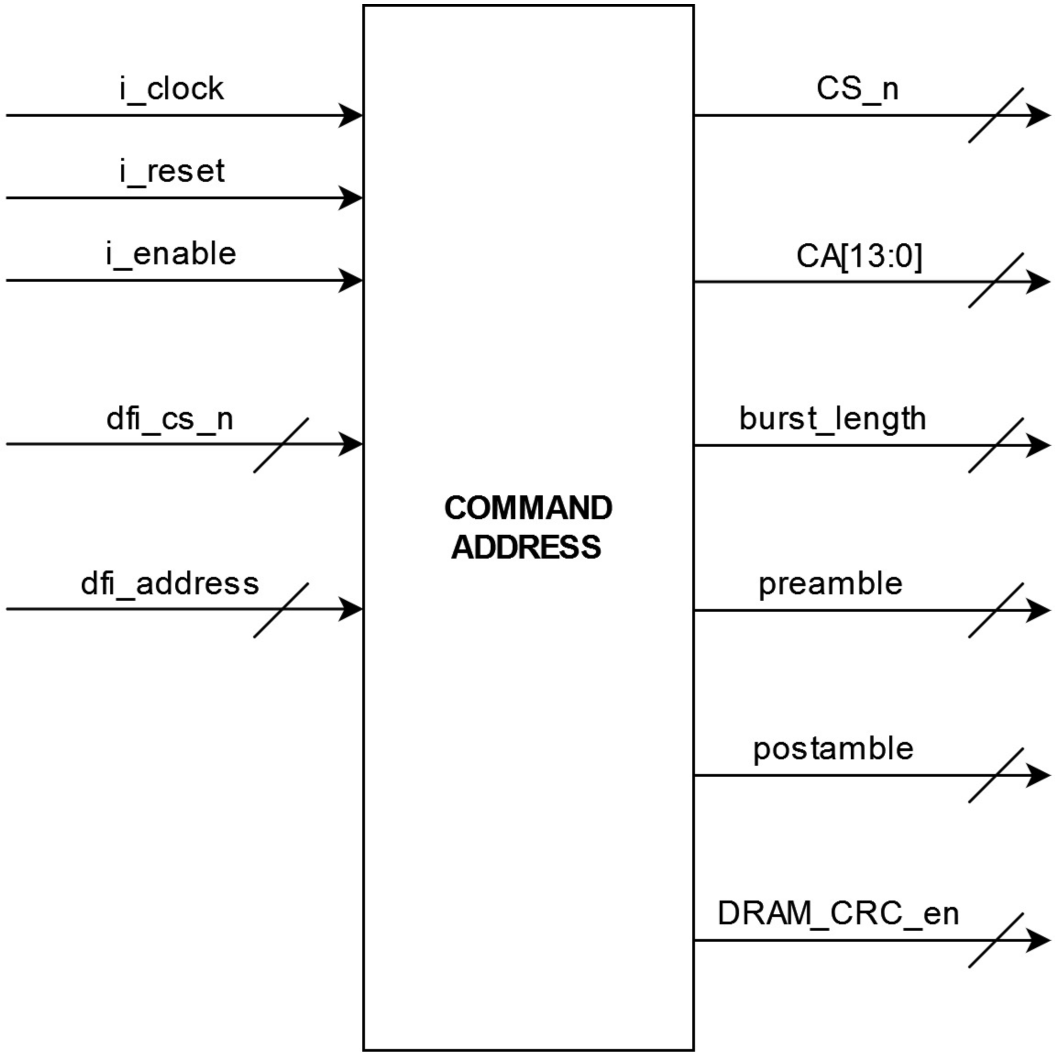
  010 the ratio is 1:4

- The following figure briefly describes the function of the block



2- COMMAND ADDRESS

1. Block Diagram

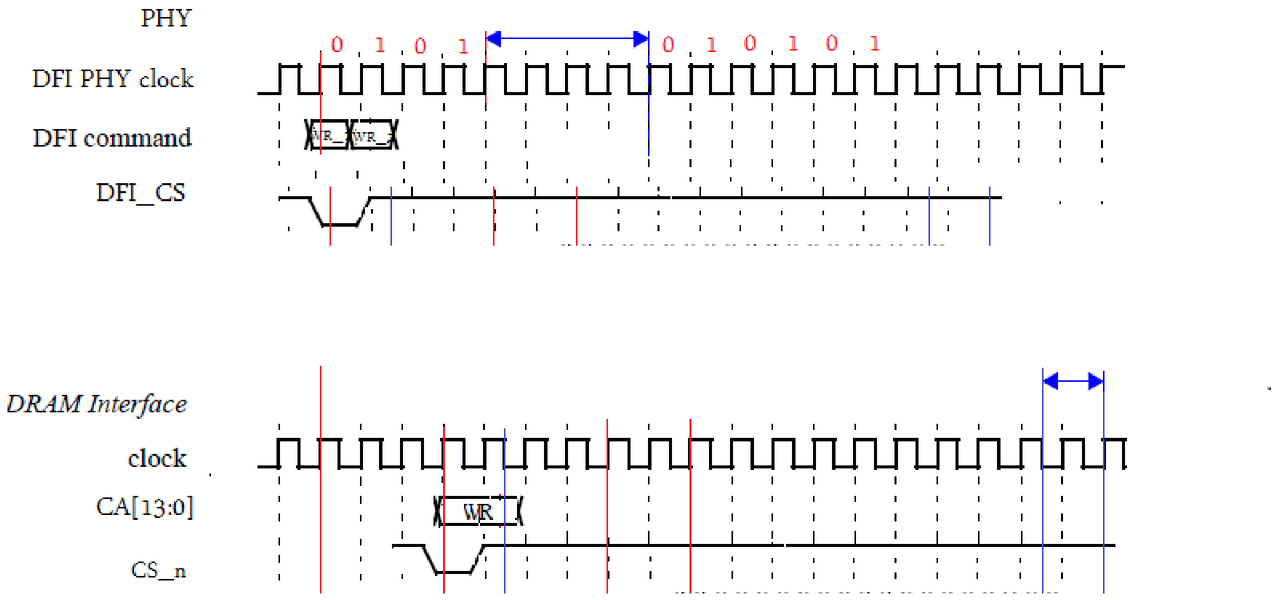


2. Input and Output ports

Port	Type	Size
dfi_address	Input	14 bits
dfi_cs	Input	NUM_RANK bits
CA	Output	14 bits
CS_n	Output	NUM_RANK bits
burst_length	Output	2 bits
preamble	Output	2 bits
postamble	Output	1 bit
DRAM_CRC_en	Output	1 bit

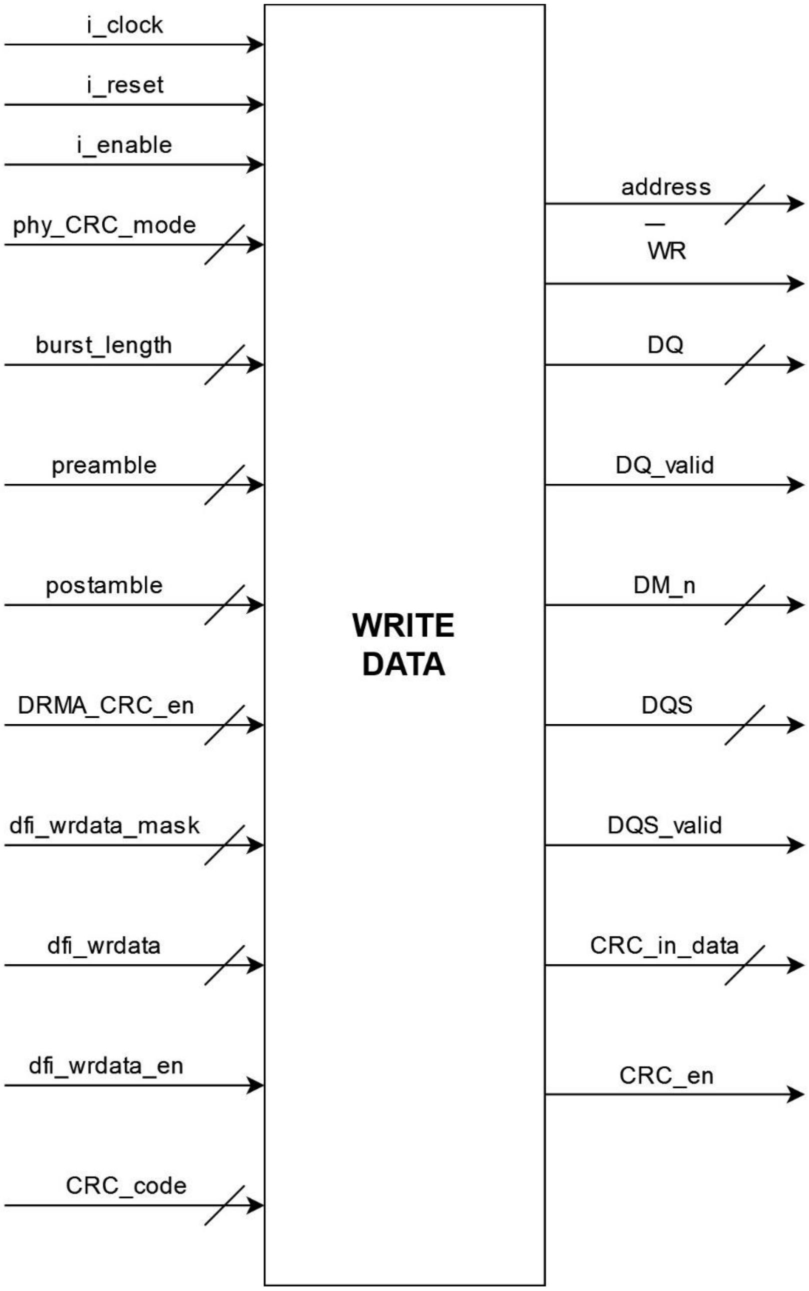
3. Block Functionality

- COMMAND ADDRESS Block will receive the dfi_address and dfi_cs from the FREQUENCY RATIO Block then sends the dfi_address on the CA bus and the dfi_cs on CS signal.
- This block determines whether the command is 1-cycle command or 2-cycle command by checking the CA [1].
    - If CA [1] = 0 → the command is 2-cycle command.
    - If CA [1] = 1 → the command is 1-cycle command.
- In case of the command is Mode Register Write Command, this block will extract the burst length, preamble, postamble and DRAM_CRC_en information from the MRW command.



3- WRITE DATA

1. Block Diagram



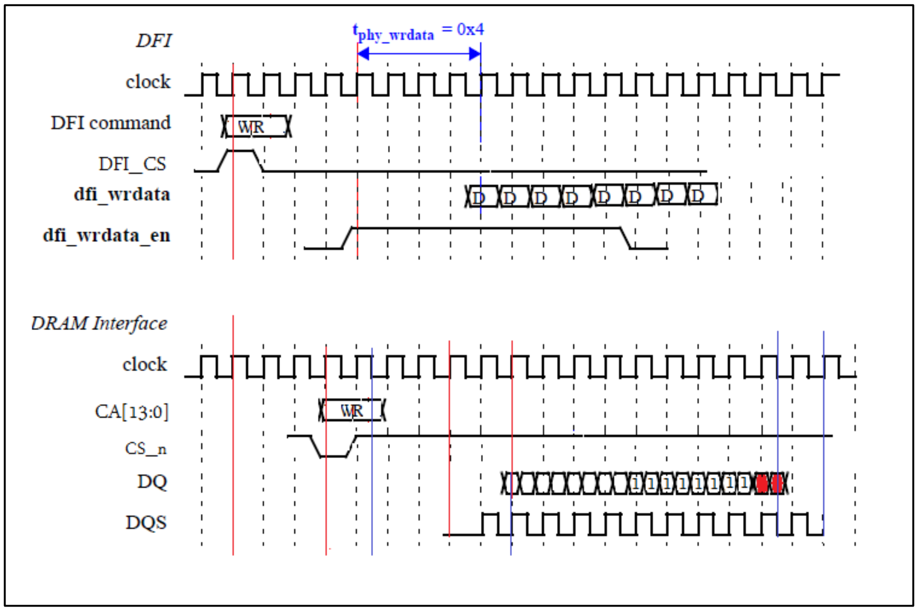
2. Input and Output Ports

Port	Type	Size
burst_length	Input	2 bits
preamble	Input	2 bits
postamble	Input	1 bit
DRAM_CRC_en	Input	1 bit
dfi_wrdata_en	Input	1 bit
dfi_wrdata	Input	2 * DEVICE_TYPE bits
dfi_wrdata_mask	Input	(2 * DEVICE_TYPE) / 8 bits
phy_CRC_mode	Input	1 bit
CRC_code	Input	8* (DEVICE_TYPE/4) bits
CRC_in_data	Output	8* (DEVICE_TYPE/4) bits
CRC_en	Output	1 bit
DQ	Output	DEVICE_TYPE bits
DQ_valid	Output	1 bit
DQS	Output	1 bit
DQS_valid	Output	1 bit
DM_n	Output	DEVICE_TYPE / 8 bits

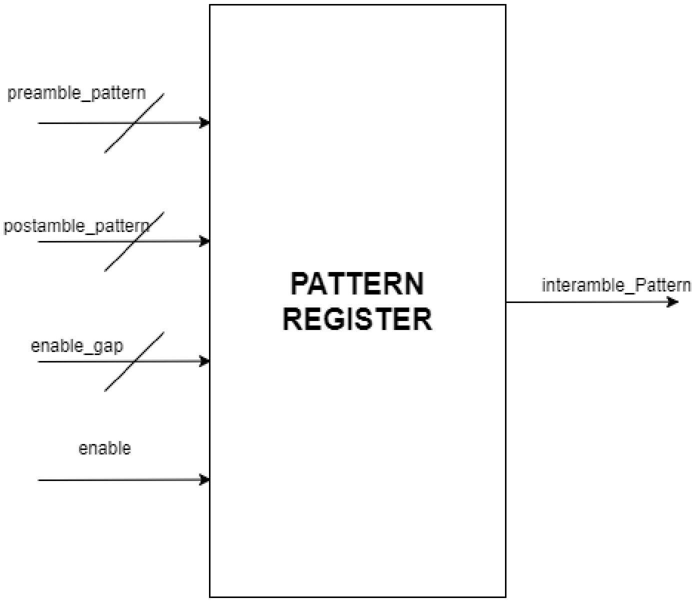
3. Block Functionality

- WRITE DATA is responsible for transmitting the data from the frequency ratio block to the DQ bus.
- Receive an inputs form command address block:
  - Burst length
  - Pre-amble
  - Postamble
  - DRAM CRC enable
- It is responsible for transmitting the crc code with the data transmitted through 2 cases
  - Case (1) phy_crc_mode = 0: Sends the data with its CRC code to the DQ bus.
  - Case (2) phy_crc_mode = 1: Sends the data to the CRC Block to generate its CRC code then receive the CRC bits from the CRC Block and sends it at the end of the burst on the DQ bus to the DRAM interface.
- It should check the burst length value and according to this value it should send the data to the CRC Block as follows
  - If burst length = 8:
    - It should take the 8-bit data.
    - Complete the rest of burst with 1s.
    - Sends it to CRC Block to generate crc code.
    - Receive the crc code from the CRC Block
    - Sends the data and its crc code on DQ bus.

- If burst length = 16:
  - It should send the whole data to the CRC Block to generate the crc code.
  - Receive the crc bits from the CRC Block.
  - Sends the data and its crc code on DQ bus.
- If burst length = 32:
  - It should divide the data into 2 halves, each is 16 bits
  - Sends the 1st half of data into the CRC Block.
  - Receives the crc bits for the 1st half.
  - Sends the 2nd half of data into the CRC Block.
  - Receives the crc bits for the 2nd half.
  - Sends the 1st half of data and their crc bits on the 17th and 18th UI then sends the 2nd half of data with their crc bits on the 35th and 36th UI.



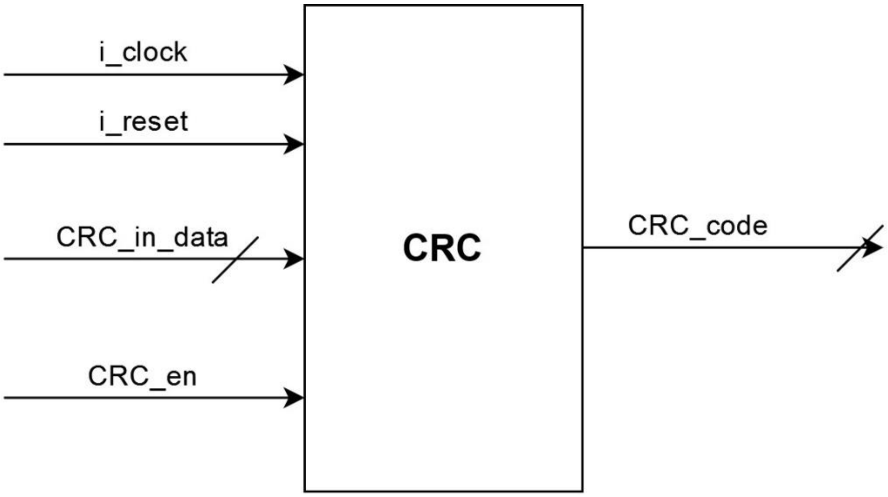
- It should generate the DQS pattern.
  - The following block illustrates how WRITE DATA Block handles and generate the DQS pattern.



  - WRITE DATA Block should check the write data enable gap between two consecutive write data and check the sum of preamble and postamble pattern bits.
    - If write data enable gap greater than 5 clock cycles, → then no back-to-back write and this block will be dis-abled (enable = 0).
    - If write data enable gap smaller than 5 clock cycles and the sum of pattern bits greater than double the enable gap clock cycles, → then it is back-to-back write and this block should be enabled.
  - PATTERN REGISTER Block
    - Takes the post-amble pattern and save it in the register (from left to write).
    - Takes the preamble pattern and save it in the register (from right to left).
    - While saving the preamble pattern, if preamble pattern collides with the post amble pattern, we should give priority to preamble. (Preambles overwrite post amble if collided)
  - The output of this block is the interamble Pattern.

4- CRC

1. Block Diagram



2. Input and Output Ports

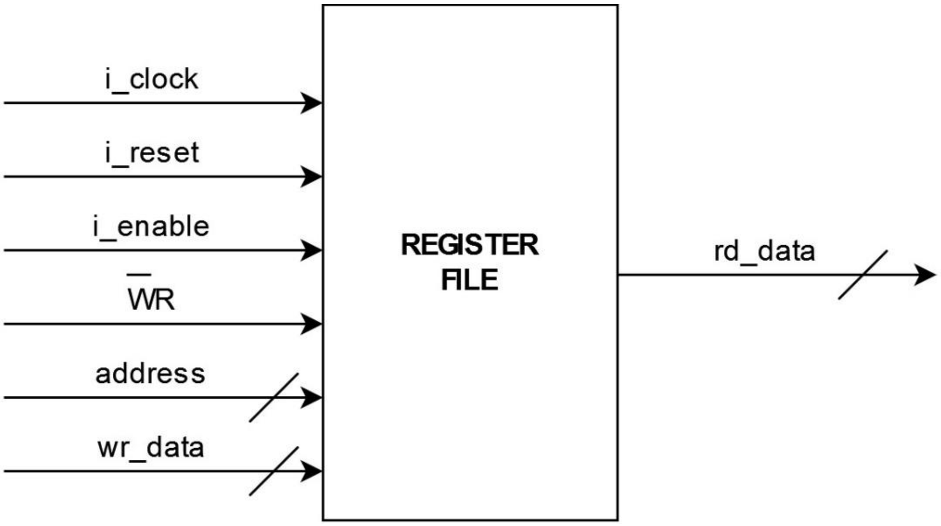
Port	Type	Size
CRC_in_data	Input	8* (DEVICE_TYPE/4) bits
CRC_en	Input	1 bit
CRC_code	Output	8* (DEVICE_TYPE/4) bits

3. Block Functionality

- CRC Block receives an input (CRC_en), this input decides either this block will work or not.
- It contains memory to store the data and send it bit by bit to the LFSR to generate the CRC code then send the CRC code to write data block.
- This block is duplicated according to the device type for:
  - X4 device, one CRC block will be used.
  - X8 device, two CRC blocks will be used.
  - X16 device, four CRC blocks will be used.

5- REGISTER FILE

1. Block Diagram



2. Input and Output Ports

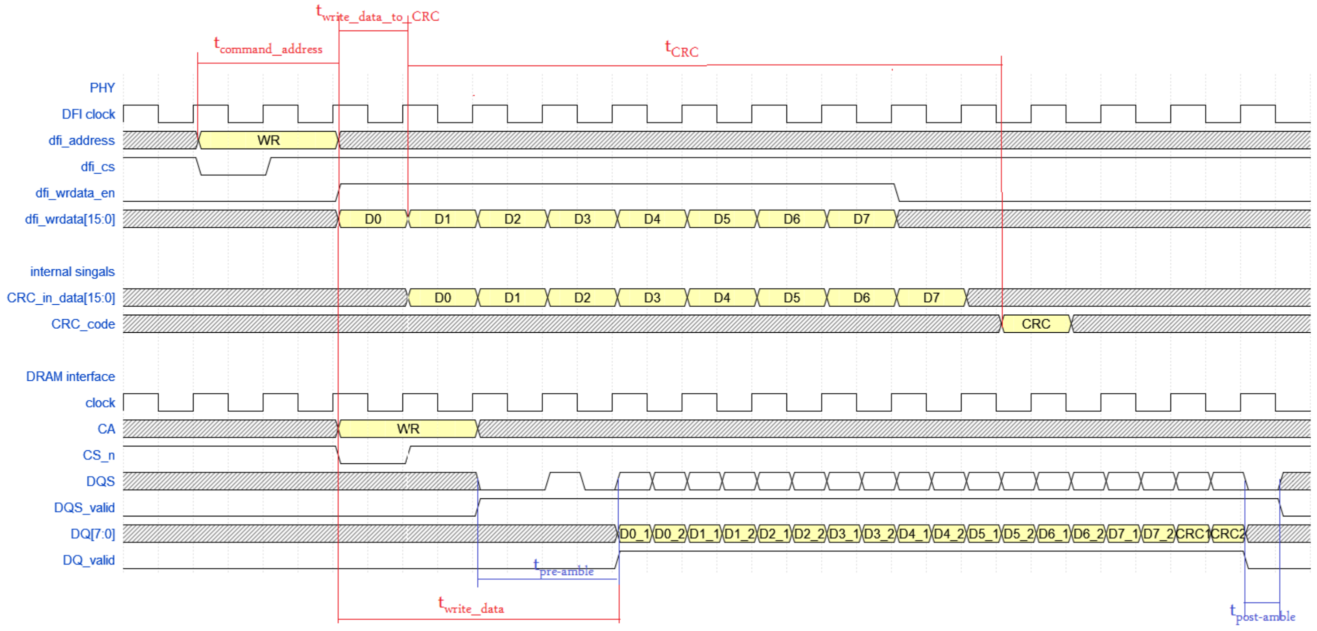
Port	Type	Size
WR	input	1 bit
address	input	Log2 (reg_file length)
Wr_data	input	Reg_file width
Rd_data	output	Reg_file width

3. Block Functionality

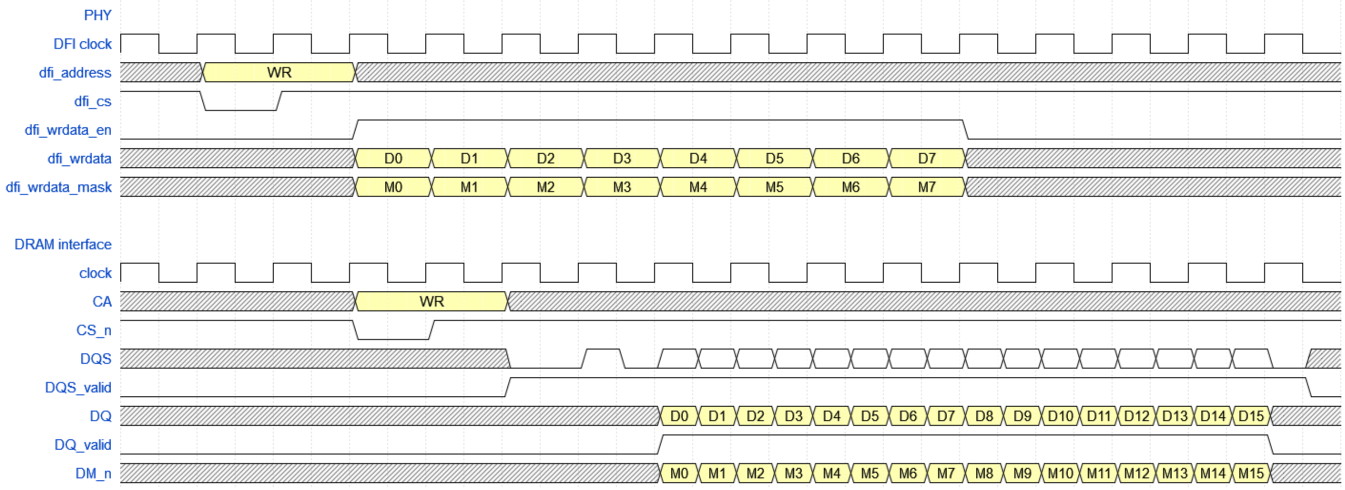
- REGISTER FILE is a block that will be used to store the programmable parameters (phy_CRC_mode, dfi_freq_ratio).
- During the initialization:
  - The Frequency ratio block will read the dfi_freq_ratio parameter.
  - The Write data block will read the phy_CRC_mode parameter.

6- Timing Diagram (DRAM interface)

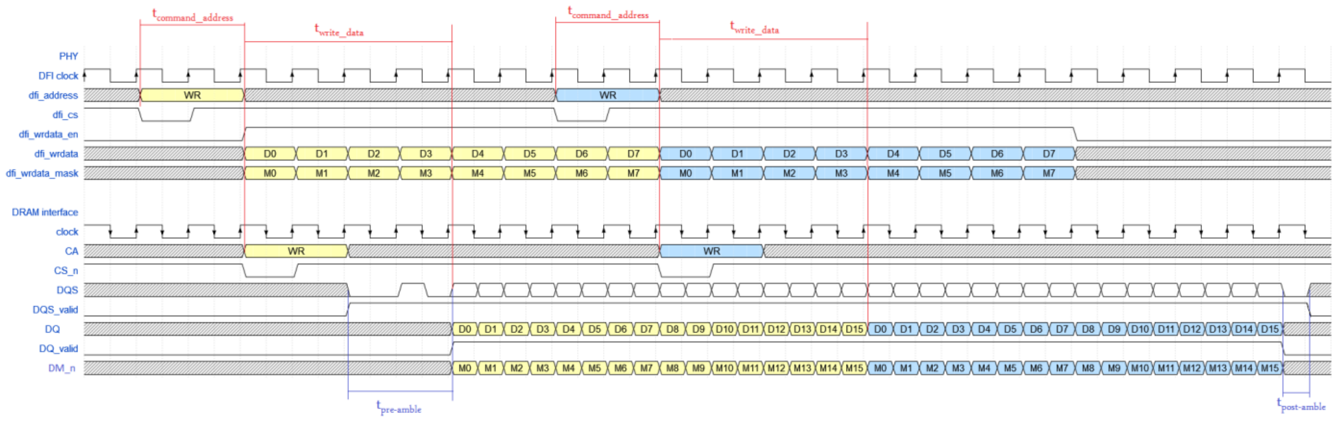
1. Single write operation with CRC



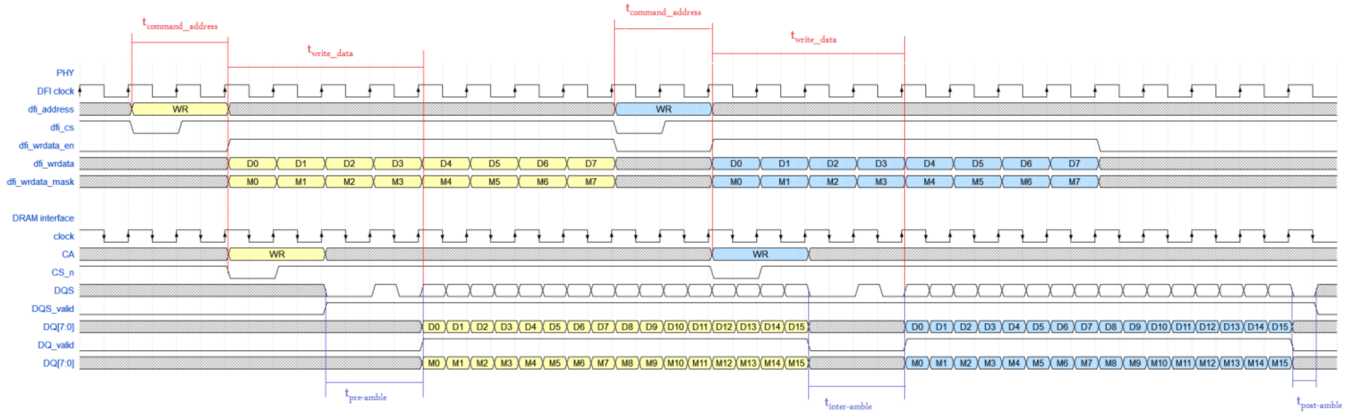
2. Single write operation with write data mask



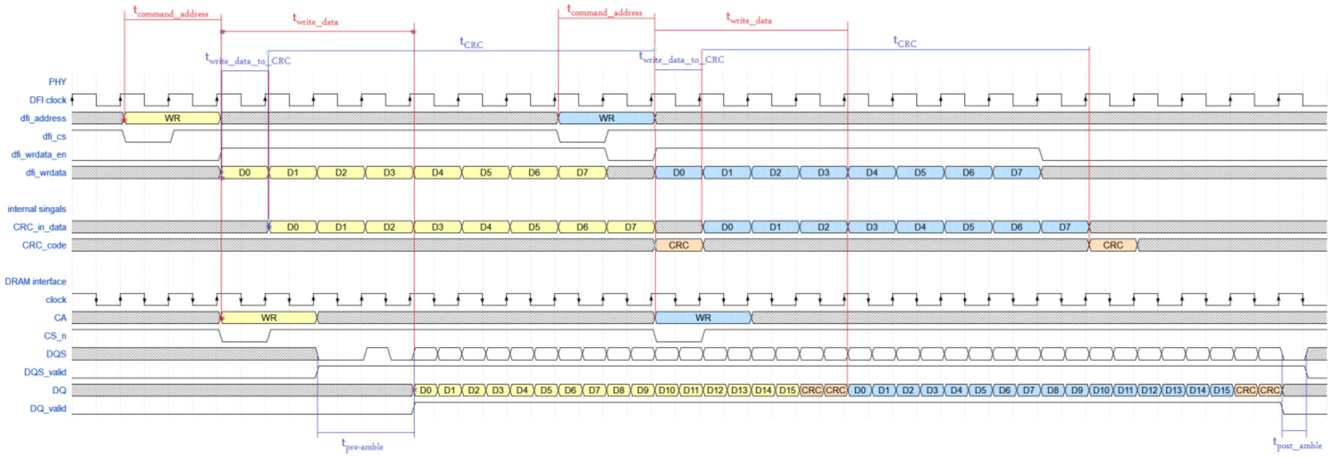
3. Back to back write with data mask



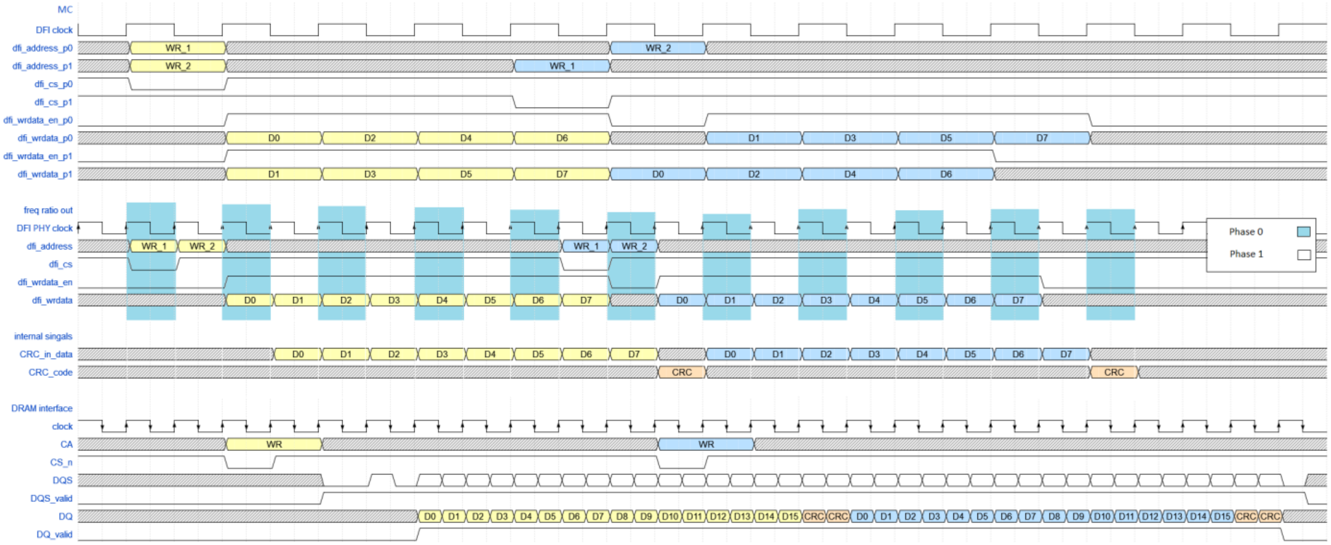
4. Two independent writes with data mask and gap 2 clock cycles



5. Back to back write with CRC



6. Back to back write with frequency ratio 2:1 with CRC



Important timing parameters

parameter	Defined by	Description
tphy_wrdata	Frequency ratio Block	This parameter specifies the number of DFI PHY clock cycles between the dfi_wrdata_en signal is asserted and the associated write data is driven on the dfi_wrdata to write data Block interface
tphy_wrlat	Frequency ratio Block	This parameter specifies the number of DFI PHY clock cycles between write commands is sent and the dfi_wrdata_en signal is asserted.

Questions

1. How the register file will be written with the desired values, and how the blocks will communicate with the register file?
2. When will the dfi_reset_n_pN signal will be used, and will it be mapped to RESET_n in JEDEC?
3. For the section 4.38.5 in JEDEC, how the CRC will be enabled and disabled?