

L_MIF specification

Rev.0.10

Tetsuo Yoshida <Tetsuo.Yoshida@jp.sony.com>

Dept.3, IP Design Division
Sony LSI Design Inc.

Table of contents

1. Definition of words	4
2. Overview.....	5
2.1. <i>Function overview</i>	5
2.2. <i>Normal function</i>	5
2.3. <i>Test function</i>	7
2.4. <i>Other function</i>	7
2.5. <i>Target device</i>	7
3. Basic configuration	8
3.1. <i>Block diagram</i>	8
3.2. <i>Clock specification</i>	9
3.3. <i>Reset specification</i>	10
4. IO pin specification	11
4.1. <i>Clock, reset</i>	11
4.2. <i>System IF</i>	11
4.3. <i>System setting IF</i>	11
4.4. <i>REG IF</i>	11
4.5. <i>PHY IF</i>	11
4.6. <i>PHY register IF</i>	12
4.7. <i>UFT register IF</i>	13
5. Register map	14
5.1. <i>SYSIF register</i>	14
5.2. <i>COM register</i>	15
5.3. <i>PHY register</i>	16
5.4. <i>UFT register</i>	17
6. IF specification.....	19
6.1. <i>System IF</i>	19
6.2. <i>Internal Bus (COM IF)</i>	22
6.3. <i>PHY IF</i>	25
6.4. <i>REG IF</i>	26
6.5. <i>SRAM IF</i>	27
6.6. <i>SYSIF register IF</i>	27
6.7. <i>COM register IF</i>	28
6.8. <i>PHY register IF</i>	29
6.9. <i>UFT register IF</i>	29
7. Normal operation mode specification	31
7.1. <i>Start-up sequence</i>	31
7.2. <i>Write and read operation</i>	33
7.3. <i>Arbitration</i>	36
7.4. <i>recovery from illegal case</i>	36
7.5. <i>frame format change</i>	37
8. Test mode specification	38
8.1. <i>Memory write test</i>	38
8.2. <i>Memory read test</i>	39
8.3. <i>Parity check mode</i>	41
8.4. <i>tst_mode_force function</i>	42
9. COM block specification	44
9.1. <i>Overview</i>	44
9.2. <i>Basic configuration</i>	45
9.3. <i>IO pin specification</i>	47
9.4. <i>Register map</i>	49
9.5. <i>IF specification</i>	49
9.6. <i>Function specification</i>	51
10. Appendix.....	58

10.1. Clock domain change	58
10.2. Sysmte IF port restriction	59
10.3. PHY IF port restriction.....	59
10.4. COM IF address and data mapping image	60
10.5. MIF, PHY, DRAM configuration	60
10.6. COM block difference between F_MIF and L_MIF.....	61
10.7. SRAM wrapper	61
10.8. PHY wrapper	62
11. Revision history	64

1. Definition of words

- Frame
Picture data written to or read from memory through MIF. A frame consists of multiple lines.
- Line
Row data element consists of multiple dots.
- Dot
Data element consists of R/G/B data.
- v_size
Vertical size (line number) of a frame.
- h_size
Horizontal size (dot number) of a line.
- DDR initialization
To send command sequence to memory for initialization.
- Controller start-up
Sequence of reset and control signals for L_MIF and PHY, and DDR initialization.

2. Overview

2.1. Function overview

L_MIF はシステム IF からの要求により、入力された frame data を DDR2 SDRAM へ書き込み、また DDR2 SDRAM から読み出した frame data をシステム IF へ出力する機能を持つメモリ IF モジュールである。L_MIF はシステム IF と DDR PHY IF の間でデータビット幅の変換を行う。またシステム IF からの要求に応じて適切な DRAM 領域にアクセスするために Address 生成機能を持つ。

L_MIF は内部にレジスタブロックを持ち、レジスタ IF を通して各種パラメータ設定、DRAM 制御、テストモードの制御などを行うことができる。

2.2. Normal function

2.2.1. Frame format

下の図は L_MIF を通してメモリに write/read する frame のイメージである。

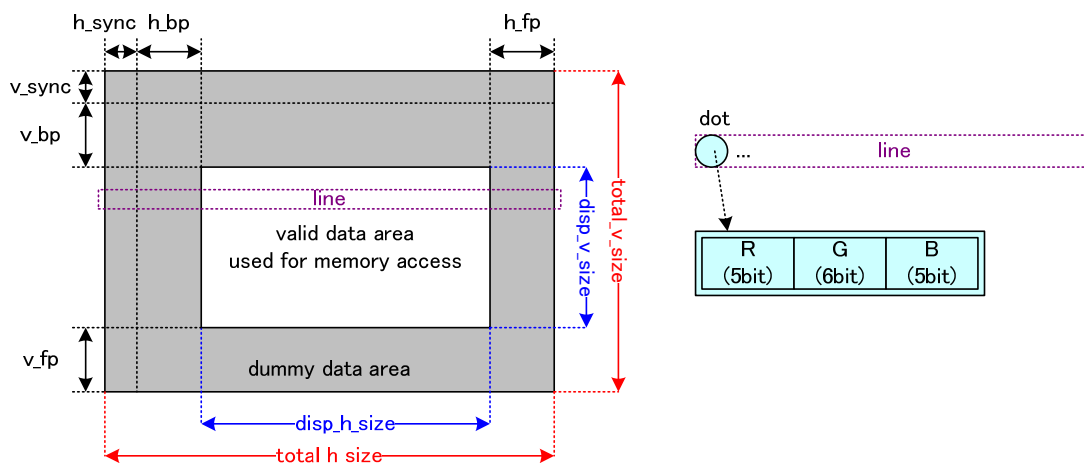


Fig. 1: Frame image

1 frame は複数の line から構成され、1 line は複数の dot から構成される。1 dot は R(5bit)/G(6bit)/B(5bit)、total 16bit のデータから構成される。

L_MIF は以下の frame format に対応可能である。

Table 1: Frame format

frame format name	total_h_size [dot]	disp_h_size [dot]	total_v_size [line]	disp_v_size [line]
QXGA	2448	2064	1562	1552
FHD	2280	1920	1090	1080
FHD(HFR)	2256	1968	1114	1104
WUXGA(HFR)	2208	1936	1226	1216
4K2.4K	2320	2080	1226	1216
2K1K	2412	2052	1094	1084
WUXGA	2292	1924	1214	1204
SXGA+	1764	1404	1064	1054

dummy data area (blanking)の仕様は下記のとおり。

Table 2: blanking spec

blanking spec		min	max
v_sync	[line]	1	total_v_size - disp_v_size - 2
v_bp	[line]	2	total_v_size - disp_v_size - 1
v_fp	[line]	0	total_v_size - disp_v_size - 3
h_sync	[dot]	0	total_h_size - disp_h_size
h_bp	[dot]	0	total_h_size - disp_h_size
h_fp	[dot]	0	total_h_size - disp_h_size

ただし、

$$(v_sync + v_bp + v_fp) = (total_v_size - disp_v_size) \geq \min 4 \text{ [line]}$$

$$(h_sync + h_bp + h_fp) = (total_h_size - disp_h_size) \geq \min 120 \text{ [dot]}$$

とする。

画像サイズは上記のもの以外にも bus performance の許容範囲内で対応可能とする(画像サイズは外部設定ピンで設定される)。

各 frame format の dot rate (min/max)をそれぞれ表に示す。

Table 3: dot rate

frame format name	dot rate [MHz]	
	min	max
QXGA	228.150	239.850
FHD	148.200	311.600
FHD(HFR)	147.713	310.575
WUXGA(HFR)	158.438	333.125
4K2.4K	169.650	356.700
2K1K	157.950	332.100
WUXGA	166.725	350.550
SXGA+	112.125	235.750

2.2.2. DRAM bit width

L_MIF の対応する DRAM ビット幅は 16bit (x1)である。

2.2.3. Additional frame format

下の表は L_MIF のサポートする追加の frame format である。

Table 4: Additional frame format

frame format name	total_h_size	disp_h_size	total_v_size	disp_v_size	dot rate [MHz]	
	[dot]	[dot]	[line]	[line]	min	max
MAXCASE	2500	2112	2058	2048	150.491	316.418
WXGA	2040	1368	820	768	48.929	205.754
BLKMIN	2040	1920	1084	1080	64.682	271.997
BLKMAX	4096	1368	4096	768	160.875	338.250

2.3. Test function

L_MIF はテスト用途として以下の機能を持つ。

Table 5: Test function

Name	Description
memory write test	Disable(ignore) write operation at system IF. User can write any data to given address in the memory through register IF.
memory read test	Disable(ignore) read operation at system IF. User can read any data from given address in the memory through register IF.
parity check mode	Add parity bit to memory write data. Check parity bit after memory read and notify error count.

memory write test と memory read test は、片方の機能のみ独立して動作、もしくは同時に両方とも動作させることもできる。片方のテスト機能のみ動作させる場合、もう一方は normal function を続ける。parity check mode は他のテスト機能とは独立して動作する。

2.4. Other function

L_MIF はその他の機能として以下のものを持つ。

- Variable DDR SDRAM timing parameter
- Variable PHY IF timing parameter
- Auto refresh control
- DDR initialize sequence control (auto/manual)
- DDR power-down/self-refresh control
- PHY setting control

2.5. Target device

- JEDEC DDR2-667D (5-5-5), 256Mb(16Mb x 16), 512Mb(32Mb x 16), 1Gb(64Mb x 16)
- JEDEC DDR2-800E (6-6-6), 256Mb(16Mb x 16), 512Mb(32Mb x 16), 1Gb(64Mb x 16)

3. Basic configuration

3.1. Block diagram

L_MIF のブロック図を下に示す。

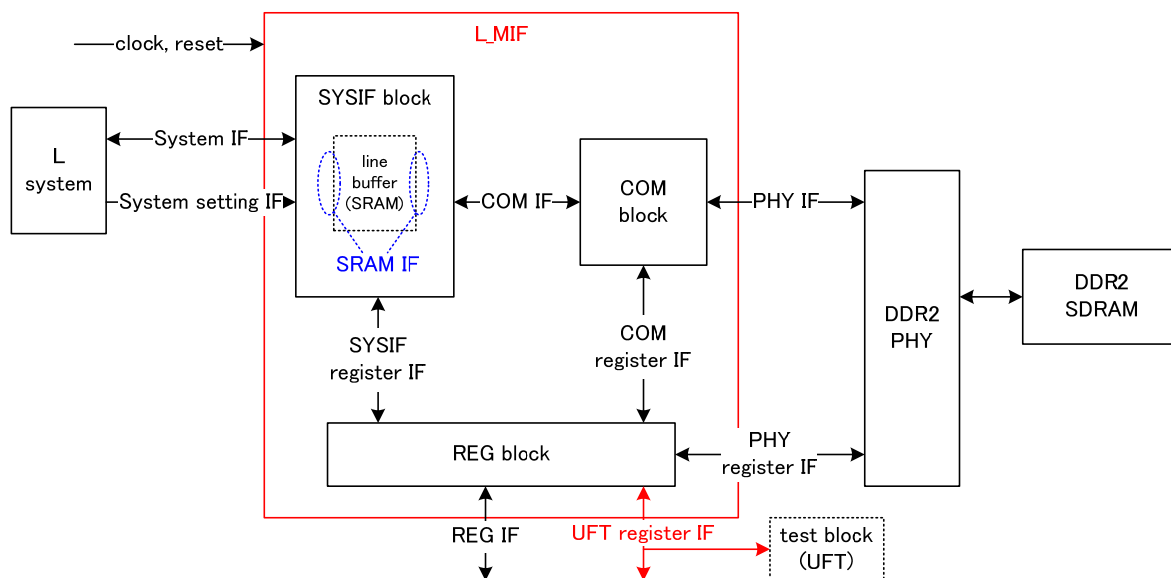


Fig. 2: Block diagram

3.1.1. SYSIF block

SYSIF block は System IF と COM IF 間のプロトコル変換を行う。System IF から frame-rw request と write frame data を受け取り、COM block が受け取り可能な format に変換して memory write request と一緒に COM IF へ出力する。また COM へ memory read を request して read frame data を受け取り、System IF の format に変換して出力する。

SYSIF block は write 側に 1line 分、read 側に 2line 分の data を保持する line buffer(SRAM)を持つ。

System IF からの frame-rw request に対して SYSIF ブロックは規則的に(決まった順番で)COM IF の arbitration を行う。

また System setting IF の設定により画像サイズの制御が行われる。

3.1.2. line buffer(SRAM)

line buffer は異なるクロックで動作する 2 つのポートを持つ dual port SRAM である。line buffer 1 個分のサイズは 64bit x 528word。SYSIF block の write 側に 1 個、read 側に 2 個、計 3 個の line buffer が搭載される。2 つのポートのうち片方は write のみ、もう一方は read のみに使用される。

line buffer は line data を保持して、line data のクロック乗り換えと、System IF と COM IF 間のプロトコル変換(ビット幅変換と arbitration)に必要なタイミング調整を行うために使用される。

3.1.3. COM block

COM block は COM IF からの memory write/read request または COM register IF からの要求により、DDR2

SDRAMコマンドシーケンスを生成してPHY IFへと出力を行う。またCOM IFとPHY IFの間でwrite/read dataの受け渡しを行い、PHY IFの制御に必要な信号の生成も行う。またCOM blockはregister settingによりDDR initialize、AREF、power-down、self-refreshの制御も行う。

3.1.4. REG block

REG blockはSYSIF、COM、PHYの各ブロックの設定を行うためのregister blockである。ユーザはREG IFを通して各registerにアクセスすることができる。register settingは、SYSIF register IF、COM register IF、PHY register IFを通して各ブロックへ渡される。

3.1.5. DDR2 PHY

DDR2 PHYはPHY IF、PHY register IFを通してL_MIFと接続されるハードマクロで、L_MIFからの制御に従ってDDR2 SDRAMのアクセスを行う。

3.2. Clock specification

L_MIFのクロックリストを下の表に挙げる。

Table 6: clock list

Clock name	Frequency	Description
pclk	max 180MHz	Clock for System write IF. equal to (write dot rate/2).
mclk	max 166MHz	Clock for COM block, REG block, COM IF, REG IF, SYSTEM register IF, COM register IF and PHY register IF.
mclkx2	max 333MHz	Clock for PHY IF.

クロックの関係は下の図のとおりで、mclkの立上りエッジとmclkx2の立上りエッジは同期しており、他の全てのクロック間の関係は非同期である。

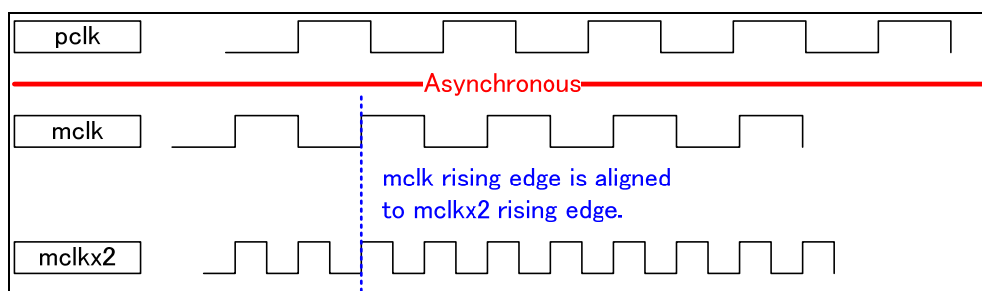


Fig. 3: clock relationship

クロックドメインを下の図に示す。

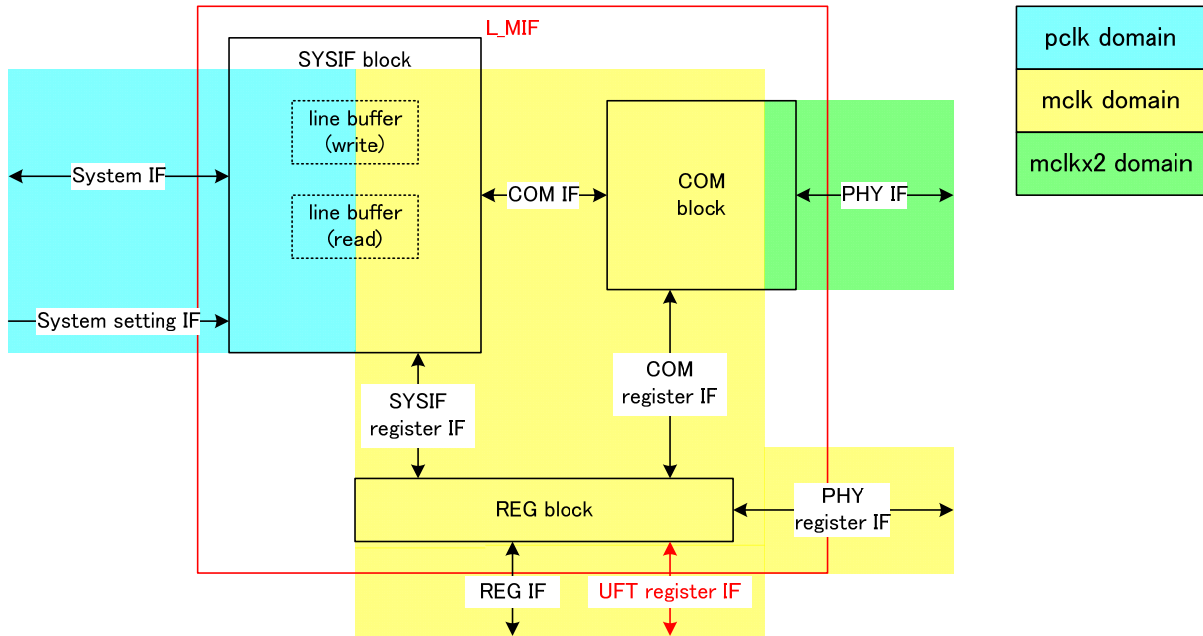


Fig. 4: clock domain

3.3. Reset specification

L_MIF の asynchronous reset を下の表に挙げる。

Table 7: reset list

Reset name	Active	Description
rst_x	Low	Asynchronous reset for all FFs in L_MIF.

4. IO pin specification

以下、L_MIF の IO ピンリストを示す。

4.1. Clock, reset

Table 8: clock, reset pin list

Name	Bit num	In Out	Initial value	From	To	Clock domain	Description
pclk	1	In	-	TOP	SYSIF	-	Clock for System IF.
mclk	1	In	-	TOP	SYSIF COM REG	-	Clock for COM block, REG block, COM IF, REG IF, SYSTEM register IF, COM register IF and PHY register IF.
mclkx2	1	In	-	TOP	COM	-	Clock for PHY IF.
rst_x	1	In	-	TOP	SYSIF COM REG	-	Asynchronous reset for all FFs in L_MIF.

4.2. System IF

Table 9: System IF pin list

Name	Bit num	In Out	Initial value	From	To	Clock domain	Description
mif_rst	1	In	-	system	SYSIF	pclk	Synchronous reset for System IF. Active high.
mif_enable	1	In	-	system	SYSIF	pclk	Write and Read Data Enable.
wdata	32	In	-	system	SYSIF	pclk	Write Data.
rdata	32	Out	0	SYSIF	system	pclk	Read Data.

4.3. System setting IF

Table 10: System setting IF pin list

Name	Bit num	In Out	Initial value	From	To	Clock domain	Description
v_addr_time_mif	12	In	-	system	SYSIF	pclk	Indicates vertical line number of a frame.
h_addr_time_mif	12	In	-	system	SYSIF	pclk	Indicates horizontal dot number of a frame.

4.4. REG IF

Table 11: REG IF pin list

Name	Bit num	In Out	Initial value	From	To	Clock domain	Description
madres	16	In	-	regbus	REG	mclk	register bus address
mdataw	8	In	-	regbus	REG	mclk	register bus write data
mdataw	8	Out	0	REG	regbus	mclk	register bus read data
mwen	1	In	-	regbus	REG	mclk	register bus write enable
mren	1	In	-	regbus	REG	mclk	register bus read enable

4.5. PHY IF

Table 12: PHY IF pin list

Name	Bit num	In Out	Initial value	From	To	Clock domain	Description
phy_cs_x	1	Out	1	COM	PHY	mclkx2	DDR CS
phy_ras_x	1	Out	1	COM	PHY	mclkx2	DDR RAS
phy_cas_x	1	Out	1	COM	PHY	mclkx2	DDR CAS
phy_we_x	1	Out	1	COM	PHY	mclkx2	DDR WE
phy_odt	1	Out	0	COM	PHY	mclkx2	DDR on-die termination control
phy_cke	1	Out	0	COM	PHY	mclkx2	DDR CKE
phy_ba	3	Out	0	COM	PHY	mclkx2	DDR bank address
phy_addr	15	Out	0	COM	PHY	mclkx2	DDR address
phy_dqsoe	1	Out	0	COM	PHY	mclkx2	DQS and DQ output buffer enable
phy_wdq0	16	Out	0	COM	PHY	mclkx2	Write data 0
phy_wdq1	16	Out	0	COM	PHY	mclkx2	Write data 1
phy_wdm0	2	Out	0	COM	PHY	mclkx2	Write data mask 0
phy_wdm1	2	Out	0	COM	PHY	mclkx2	Write data mask 1
phy_rdcmd	1	Out	0	COM	PHY	mclkx2	Read command enable to PHY
phy_rdq0	16	In	-	PHY	COM	mclkx2	Read data 0
phy_rdq1	16	In	-	PHY	COM	mclkx2	Read data 1
phy_rodten	1	Out	0	COM	PHY	mclkx2	PHY read on-die termination enable

4.6. PHY register IF

Table 13: PHY register IF pin list

Name	Bit num	In Out	Initial value	From	To	Clock domain	Description
phy_rst_x	1	Out	0	REG	PHY	mclk	PHY reset (Active low)
phy_ddr1	1	Out	0	REG	PHY	mclk	Set ddr1 pin to 0
phy_cal_en	1	Out	0	REG	PHY	mclk	Calibraton enable
phy_comp_sel	1	Out	1	REG	PHY	mclk	Auto calibration select
phy_dip	5	Out	0	REG	PHY	mclk	PHY DIP setting
phy_din	5	Out	0	REG	PHY	mclk	PHY DIN setting
phy_clkoen	1	Out	1	REG	PHY	mclk	DDR clock and ODT driver output enable
phy_cmdoen	1	Out	1	REG	PHY	mclk	DDR ADDR,BA,RAS,CAS,WE and CS driver output enable
phy_gds	2	Out	2'b01	REG	PHY	mclk	Read pointer counter control
phy_sio	1	Out	0	REG	PHY	mclk	DQS single-ended mode select
phy_odtmd	2	Out	0	REG	PHY	mclk	PHY read ODT setting
phy_dqie	1	Out	1	REG	PHY	mclk	Enable DQ and DQS receiver
phy_dllsel	3	Out	0	REG	PHY	mclk	Read DQS DLL delay setting
phy_flodo	1	In	-	PHY	REG	-	DLL lock flag
phy_dllpdn	1	Out	0	REG	PHY	mclk	DLL power-down control
phy_dllfrange	2	Out	0	REG	PHY	mclk	DLL reference frequency select
phy_msdy	2	Out	0	REG	PHY	mclk	Read window position logic setting
phy_dmyodt	1	Out	0	REG	PHY	mclk	Enable ODT for the DUMMY pin

4.7. UFT register IF

Table 14: UFT register IF pin list

Name	Bit num	In Out	Initial value	From	To	Clock domain	Description
p_l_uft_phy_on	1	Out	0	REG	UFT	mclk	for test block
p_l_phy_tpg_on	1	Out	0	REG	UFT	mclk	for test block
p_l_phy_tpg_start	13	Out	0	REG	UFT	mclk	for test block
p_l_phy_tpg_end	13	Out	0	REG	UFT	mclk	for test block
p_l_phy_al_delay	1	Out	0	REG	UFT	mclk	for test block
p_l_phy_gds_delay	1	Out	0	REG	UFT	mclk	for test block
p_l_phy_tpg_seed	8	Out	0	REG	UFT	mclk	for test block
p_l_phy_crc_on	1	Out	0	REG	UFT	mclk	for test block
p_l_phy_crc_trig	13	Out	0	REG	UFT	mclk	for test block
p_l_phy_crc_start	13	Out	0	REG	UFT	mclk	for test block
p_l_phy_crc_end	13	Out	0	REG	UFT	mclk	for test block
p_l_phy_crc_result	32	In	0	UFT	REG	mclk	for test block
p_uft_lvds_run	1	Our	0	REG	TOP	mclk	for test block
p_uft_lvds_ch_en	4	Our	0	REG	TOP	mclk	for test block
p_uft_lvds_vsz_all	13	Our	0x020D	REG	TOP	mclk	for test block
p_uft_lvds_vsz_act	13	Our	0x01E0	REG	TOP	mclk	for test block
p_uft_lvds_vpos	13	Our	0x0028	REG	TOP	mclk	for test block
p_uft_lvds_hsz_all	13	Our	0x035A	REG	TOP	mclk	for test block
p_uft_lvds_hsz_act	13	Our	0x02D0	REG	TOP	mclk	for test block
p_uft_lvds_hpos	13	Our	0x0078	REG	TOP	mclk	for test block
p_uft_lvds_seed	10	Our	0x155	REG	TOP	mclk	for test block

5. Register map

L_MIF のレジスタマップのベースアドレスは、
<Base Address>: 0x0F00

実際の Register のアドレスは、
<Register Address> = <Base Address> + <Offset Address>

以下、L_MIF のレジスタマップを示す。

5.1. SYSIF register

Table 15: SYSIF register map

Offset Address	Register name	Bit	Parameter assign	Read Write	Initial value	Function
0x0000	reserved					
0x0001	TST_MODE	[2:0]	tst_mode	RW	0	0: Normal function mode 1: Memory write test is enabled 2: Memory read test is enabled 3: Memory write test and memory read test are enabled 4: Parity check mode is enabled 5-7: reserved
		[4]	tst_mode_force	RW	0	When toggled 0 to 1, test mode is forced (enabled without wr_rst/rd_rst).
0x0002	TST_CTRL	[0]	tst_cmd_req	RW	0	When set to 1, test command request is issued. (clear function is deleted)
		[1]	tst_cmd_status	R	0	Indicates test command is being executed.
		[7:4]	tst_cmd_mode	RW	0	0x0: Test command is Read x 1 0x1: Test command is Write x 1 0x2: Test command is Write x 2 : : 0xF: Test command is Write x 15
0x0003	TST_ADR0	[7:0]	tst_cmd_addr[7:0]	RW	0	Test command address.
0x0004	TST_ADR1	[7:0]	tst_cmd_addr[15:8]	RW	0	Corresponds to CMD IF address.
0x0005	TST_ADR2	[5:0]	tst_cmd_addr[21:16]	RW	0	
0x0006	reserved					
0x0007 0x0008	TST_WD0	[7:0]	tst_wd0[7:0]	RW	0	Test write data 0.
		[7:0]	tst_wd0[15:8]			
0x0009 0x000A	TST_WD1	[7:0]	tst_wd1[7:0]	RW	0	Test write data 1.
		[7:0]	tst_wd1[15:8]			
0x000B 0x000C	TST_WD2	[7:0]	tst_wd2[7:0]	RW	0	Test write data 2.
		[7:0]	tst_wd2[15:8]			
0x000D 0x000E	TST_WD3	[7:0]	tst_wd3[7:0]	RW	0	Test write data 3.
		[7:0]	tst_wd3[15:8]			
0x000F	reserved					
0x0010 0x0011	TST_RD0	[7:0]	tst_rd0[7:0]	R	0	Test read data 0.
		[7:0]	tst_rd0[15:8]			
0x0012 0x0013	TST_RD1	[7:0]	tst_rd1[7:0]	R	0	Test read data 1.
		[7:0]	tst_rd1[15:8]			
0x0014 0x0015	TST_RD2	[7:0]	tst_rd2[7:0]	R	0	Test read data 2.
		[7:0]	tst_rd2[15:8]			
0x0016 0x0017	TST_RD3	[7:0]	tst_rd3[7:0]	R	0	Test read data 3.
		[7:0]	tst_rd3[15:8]			

0x0018	ERR_HOLD	[0]	err_hold	RW	0	0: Refresh error count at every mif_rst. 1: Hold error count.
0x0019 : 0x001B	ERR_EVEN_L ERR_EVEN_M ERR_EVEN_H	[7:0] [7:0] [7:0]	err_even[7:0] err_even[15:8] err_even[23:16]	R	0	Parity error count for even data.
0x001C : 0x001E	ERR_ODD_L ERR_ODD_M ERR_ODD_H	[7:0] [7:0] [7:0]	err_odd[7:0] err_odd[15:8] err_odd[23:16]	R	0	Parity error count for odd data.
0x001F : 0x0030	reserved					

5.2. COM register

Table 16: COM register map

Offset Address	Register name	Bit	Parameter assign	Read Write	Initial value	Function
0x0031	COM_CTRL	[0]	com_enable	RW	0	When set to 1, COM block is enabled. COM mode setting and DDR timing parameter will be valid when this bit is toggled 0 to 1.
		[4]	com_status	R	0	0: COM block is not active. 1: COM block is active.
0x0032	COM_DEVSEL	[0]	ddr_cabit	RW	0	0:CA=9bit 1:CA=10bit
		[1]	ddr_8bank	RW	0	0:BA=2bit (4bank device) 1:BA=3bit (8bank device)
0x0033	COM_TPARAM0	[2:0]	ddr_t_rcd	RW	0x5	DDR timing parameter
		[6:4]	ddr_t_rp	RW	0x5	DDR timing parameter
0x0034	COM_TPARAM1	[4:0]	ddr_t_rc	RW	0x14	DDR timing parameter
0x0035	COM_TPARAM2	[4:0]	ddr_t_ras_min	RW	0x0F	DDR timing parameter
0x0036	COM_TPARAM3	[2:0]	ddr_t_rrd	RW	0x4	DDR timing parameter
		[5:4]	ddr_t_rtp	RW	0x3	DDR timing parameter
0x0037	COM_TPARAM4	[2:0]	ddr_t_mrd	RW	0x2	DDR timing parameter
0x0038	COM_TPARAM5	[7:0]	ddr_t_rfc	RW	0x23	DDR timing parameter
0x0039	COM_TPARAM6	[7:0]	ddr_t_refi[7:0]	RW	0x28	DDR timing parameter
0x003A	COM_TPARAM7	[3:0]	ddr_t_refi[11:8]	RW	0xA	DDR timing parameter
0x003B	COM_TPARAM8	[1:0]	ddr_t_wtr	RW	0x3	DDR timing parameter
		[5:4]	ddr_t_rtw_add	RW	0	DDR timing parameter
0x003C	COM_INITMOD	[0]	ddr_auto_init_en	RW	0x1	0: Disable auto DDR initialization at pos-edge of com_enable. 1: Enable auto DDR initialization at pos-edge of com_enable.
0x003D	COM_AREFMOD	[0]	ddr_aref_en	RW	0x1	0: Disable auto refresh. 1: Enable auto-refresh.
		[7:4]	ddr_aref_num	RW	0x1	Number of AREF commands consecutively performed after one AREF request. AREF request is issued every period of (ddr_aref_num * ddr_t_refi / 2) cycle @mclk.
0x003E	COM_SET_REFI	[0]	ddr_set_trefi	RW	0	After changing ddr_t_refi, toggle this bit 0 to 1 (if tREFI needs to be changed), then new ddr_t_refi value is set to AREF controller.
0x003F	DRCMD_CTRL	[0]	drcmd_req	RW	0	When set to 1, direct command request is issued. (clear function is deleted)

0x0040	DRCMD_CMD	[3:0]	drcmd_cmd	RW	0	Direct command type 0x0: NOP 0x1: Precharge all 0x2: Auto refresh 0x3: MRS 0x4: EMRS1 0x5: EMRS2 0x6: EMRS3 0x7: Device deselect 0x8: Self refresh entry 0x9: Power down entry 0xA to 0xF: Reserved
0x0041	DDR_MR_L	[7:0]	ddr_mr[7:0]	RW	0x52	DDR MR setting
0x0042	DDR_MR_H	[7:0]	ddr_mr[15:8]	RW	0x08	Used for auto-initialization and direct command
0x0043	DDR_EMR1_L	[7:0]	ddr_emr1[7:0]	RW	0	DDR EMR1 setting
0x0044	DDR_EMR1_H	[7:0]	ddr_emr1[15:8]	RW	0	Used for auto-initialization and direct command
0x0045	DDR_EMR2_L	[7:0]	ddr_emr2[7:0]	RW	0	DDR EMR2 setting
0x0046	DDR_EMR2_H	[7:0]	ddr_emr2[15:8]	RW	0	Used for auto-initialization and direct command
0x0047	DDR_EMR3_L	[7:0]	ddr_emr3[7:0]	RW	0	DDR EMR3 setting
0x0048	DDR_EMR3_H	[7:0]	ddr_emr3[15:8]	RW	0	Used for auto-initialization and direct command
0x0049	PHYIF_PARAM0	[2:0]	phyif_wen_lat	RW	0x3	PHY IF setting
		[6:4]	phyif_wodt_lat	RW	0x1	PHY IF setting
0x004A	PHYIF_PARAM1	[2:0]	phyif_wodt_ext	RW	0x1	PHY IF setting
		[6:4]	phyif_wdata_lat	RW	0x4	PHY IF setting
0x004B	PHYIF_PARAM2	[2:0]	phyif_ren_lat	RW	0x5	PHY IF setting
		[6:4]	phyif_rodt_lat	RW	0x3	PHY IF setting
0x004C	PHYIF_PARAM3	[2:0]	phyif_rodt_ext	RW	0x5	PHY IF setting
		[4]	phyif_wodt_en	RW	0	Write ODT enable
0x004D	PHYIF_RDLAT	[2:0]	phyif_rdlat	RW	0x3	PHY IF setting

5.3. PHY register

Table 17: PHY register map

Offset Address	Register name	Bit	Parameter assign	Read Write	Initial value	Function
0x004E	PHY_RESET	[0]	phy_rst_x	RW	0	PHY reset (Active low)
0x004F	PHYC_DDR1	[0]	phy_ddr1	RW	0	Set phy_ddr1 pin to 0
0x0050	PHYC_CALCTRL	[0]	phy_cal_en	RW	0	Calibraton enable
		[4]	phy_comp_sel	RW	0x1	Auto calibration select 0: Use dip/din setting for impedance configuration. 1: Auto calibration select.
0x0051	PHYC_DIP	[4:0]	phy_dip	RW	0	PHY DIP setting
0x0052	PHYC_DIN	[4:0]	phy_din	RW	0	PHY DIN setting
0x0053	PHYA_OE	[0]	phy_clkoen	RW	0x1	DDR clock and ODT driver output enable 0: Output disable 1: Output enable
		[4]	phy_cmdoen	RW	0x1	DDR ADDR,BA,RAS,CAS,WE and CS driver output enable 0: Output disable 1: Output enable
0x0054	PHYD_GDS	[1:0]	phy_gds	RW	0x1	Read pointer counter control 1: read data is valid RL+4 after read command 2: read data is valid RL+5 after read command 0, 3: invalid setting

0x0055	PHYD_IOCTL	[0]	phy_sio	RW	0	DQS single-ended mode select 0: DQS differential mode 1: DQS single-ended mode
		[4]	phy_dqie	RW	0x1	Enable DQ and DQS receiver 0: Input disable 1: Input enable
0x0056	PHYD_ODTMD	[1:0]	phy_odtmd	RW	0	PHY read ODT setting 0: Off 1: 75 ohm 2: 150 ohm 3: 50 ohm
0x0057	PHYD_DLLSEL	[2:0]	phy_dllsel	RW	0	Read DQS DLL delay setting
0x0058	reserved					
0x0059	PHYD_DLL_LOCK	[0]	phy_flco	R	-	DLL lock flag
0x005A	PHYD_DLLMD	[0]	phy_dllpdn	RW	0	DLL power-down control 0: DLL power-down 1: DLL power-on
		[5:4]	phy_dllfrange	RW	0	DLL reference frequency select 0: DDR2 600 to 810 2: DDR2 350 to 600 1, 3: Reserved
0x005B	PHYD_MSDLY	[1:0]	phy_msdlly	RW	0	Read window position logic setting
0x005C	PHYD_DMYODT	[0]	phy_dmyodten	RW	0	Enable ODT for the DUMMY pin

5.4. UFT register

Table 18: UFT register map

Offset Address	Register name	Bit	Parameter assign	Read Write	Initial value	Function
0x0060	UFT_PHY_L_00	[0]	p_l_uft_phy_on	RW	0	for test block
		[1]	p_l_phy_tpg_on	RW	0	for test block
		[7:2]	p_l_phy_tpg_start[5:0]	RW	0	for test block
0x0061	UFT_PHY_L_01	[6:0]	p_l_phy_tpg_start[12:6]	RW	0	for test block
		[7]	p_l_phy_tpg_end[0]	RW	0	for test block
0x0062	UFT_PHY_L_02	[7:0]	p_l_phy_tpg_end[8:1]	RW	0	for test block
0x0063	UFT_PHY_L_03	[3:0]	p_l_phy_tpg_end[12:9]	RW	0	for test block
		[4]	p_l_phy_al_delay	RW	0	for test block
		[5]	p_l_phy_gds_delay	RW	0	for test block
0x0064	UFT_PHY_L_04	[7:0]	p_l_phy_tpg_seed[7:0]	RW	0	for test block
0x0065	UFT_PHY_L_05	[0]	p_l_phy_crc_on	RW	0	for test block
		[7:1]	p_l_phy_crc_trig[6:0]	RW	0	for test block
0x0066	UFT_PHY_L_06	[5:0]	p_l_phy_crc_trig[12:7]	RW	0	for test block
		[7:6]	p_l_phy_crc_start[1:0]	RW	0	for test block
0x0067	UFT_PHY_L_07	[7:0]	p_l_phy_crc_start[9:2]	RW	0	for test block
0x0068	UFT_PHY_L_08	[2:0]	p_l_phy_crc_start[12:10]	RW	0	for test block
		[7:3]	p_l_phy_crc_end[4:0]	RW	0	for test block
0x0069	UFT_PHY_L_09	[7:0]	p_l_phy_crc_end[12:5]	RW	0	for test block
0x006A	UFT_PHY_L_0A	[7:0]	p_l_phy_crc_result[7:0]	R	0	for test block
0x006B	UFT_PHY_L_0B	[7:0]	p_l_phy_crc_result[15:8]	R	0	for test block
0x006C	UFT_PHY_L_0C	[7:0]	p_l_phy_crc_result[23:16]	R	0	for test block
0x006D	UFT_PHY_L_0D	[7:0]	p_l_phy_crc_result[31:24]	R	0	for test block
0x006E	UFT_LVDS_TPG_00	[0]	p_uft_lvds_run	RW	0	for test block
		[4:1]	p_uft_lvds_ch_en[3:0]	RW	0	for test block

0x006F	UFT_LVDS_TPG_01	[7:0]	p_uft_lvds_vsz_all[7:0]	RW	0x0D	for test block
0x0070	UFT_LVDS_TPG_02	[4:0]	p_uft_lvds_vsz_all[12:8]	RW	0x02	for test block
0x0071	UFT_LVDS_TPG_03	[7:0]	p_uft_lvds_vsz_act[7:0]	RW	0xE0	for test block
0x0072	UFT_LVDS_TPG_04	[4:0]	p_uft_lvds_vsz_act[12:8]	RW	0x01	for test block
0x0073	UFT_LVDS_TPG_05	[7:0]	p_uft_lvds_vpos[7:0]	RW	0x28	for test block
0x0074	UFT_LVDS_TPG_06	[4:0]	p_uft_lvds_vpos[12:8]	RW	0	for test block
0x0075	UFT_LVDS_TPG_07	[7:0]	p_uft_lvds_hsz_all[7:0]	RW	0x5A	for test block
0x0076	UFT_LVDS_TPG_08	[4:0]	p_uft_lvds_hsz_all[12:8]	RW	0x03	for test block
0x0077	UFT_LVDS_TPG_09	[7:0]	p_uft_lvds_hsz_act[7:0]	RW	0xD0	for test block
0x0078	UFT_LVDS_TPG_0A	[4:0]	p_uft_lvds_hsz_act[12:8]	RW	0x02	for test block
0x0079	UFT_LVDS_TPG_0B	[7:0]	p_uft_lvds_hpos[7:0]	RW	0x78	for test block
0x007A	UFT_LVDS_TPG_0C	[4:0]	p_uft_lvds_hpos[12:8]	RW	0	for test block
0x007B	UFT_LVDS_TPG_0D	[7:0]	p_uft_lvds_seed[7:0]	RW	0x55	for test block
0x007C	UFT_LVDS_TPG_0E	[1:0]	p_uft_lvds_seed[9:8]	RW	0x1	for test block

6. IF specification

6.1. System IF

System IF は、L_MIF が system より frame-rw request と write frame data を受け取り、read frame data を出力する IF である。この IF の信号は全て pclk に同期して動作する。

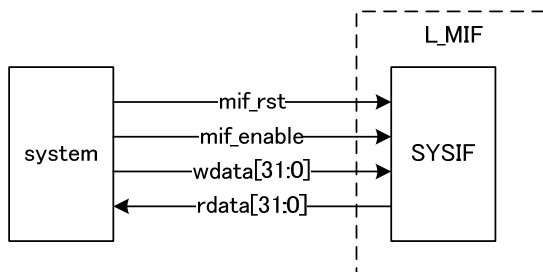


Fig. 5: System IF

wdata[31:0]と rdata[31:0]はそれぞれ 2dot 分(16bit x 2)のビット幅を持ち、dot data は下位側から順に詰められる。

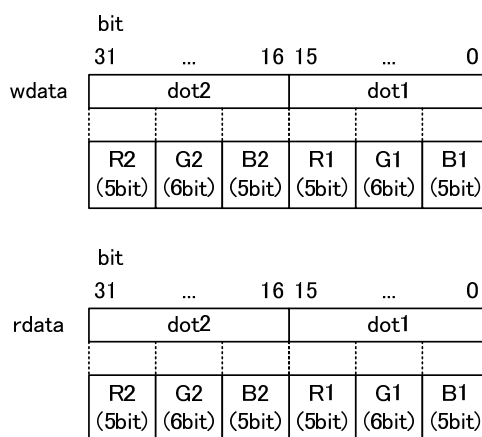


Fig. 6: wdata and rdata

1 frame は v_addr_time_mif line から構成され、1 line は h_addr_time_mif/2 cycle@pclk の wdata または rdata から構成される。

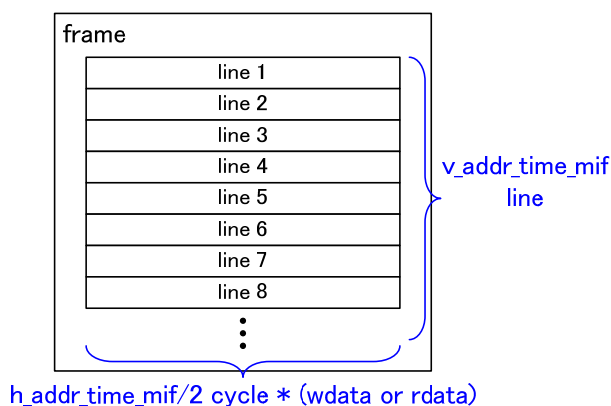


Fig. 7: frame image

下の図は System IF の基本動作である。

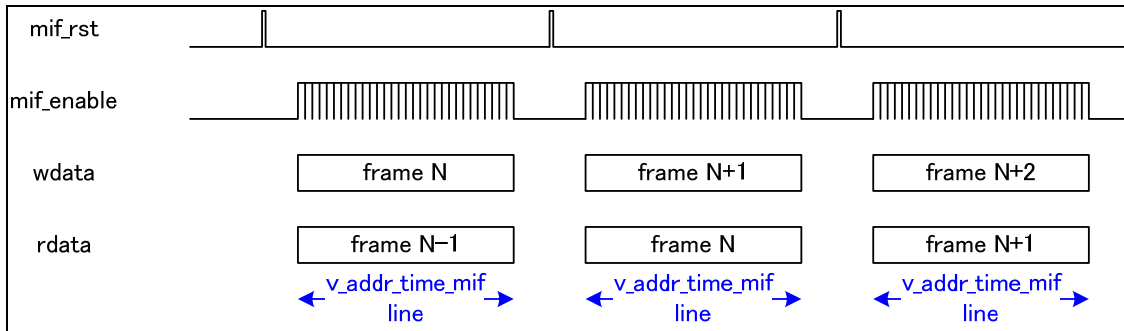


Fig. 8: basic operation

mif_rst=1 が frame-rw request の始まりを示し、その後 frame の write と read が同時に行われる。read 側はその 1 frame 前に write したフレームを読み出して出力する。

mif_enable=1 のときの wdata が有効な write frame data となり、rdata はそれから 3cycle 遅れて有効な read frame data の出力を行う。write 側が frame N の line M を write しているとき、read 側は frame N-1 の line M を出力する。

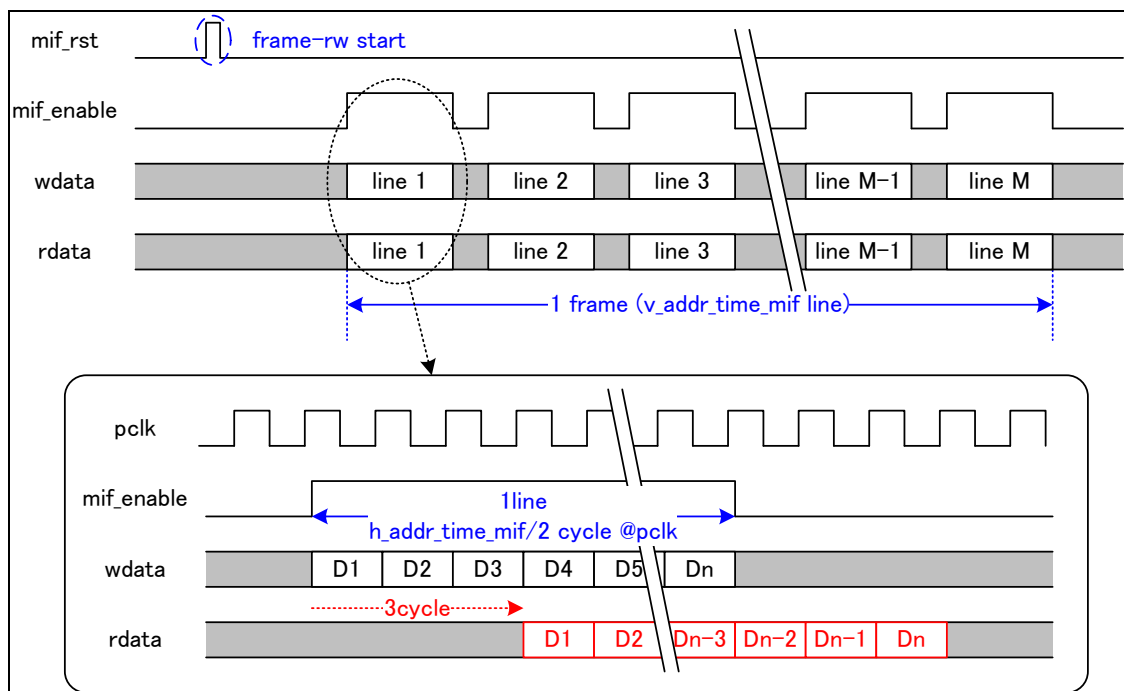


Fig. 9: frame and line data

mif_rst と mif_enable の詳細な timing spec は下の図のとおり。

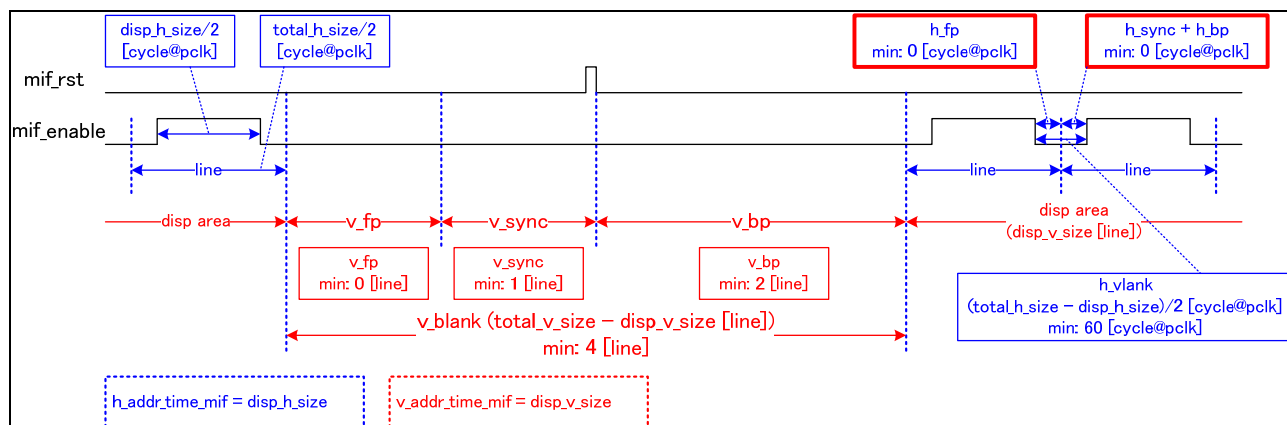


Fig. 10: detailed timing spec of mif_rst/mif_enable

mif_enable=1 の 1 区間が 1 line となる (mif_enable の立上りから line が始まり、mif_enable の立下りで終わる)。mif_enable=0 の区間を挟んで複数の line が read/write されることにより frame が構成される。

mif_enable=1 の期間が h_addr_time_mif/2 サイクルを超えると、その line の read/write はそれ以上行わない。また mif_enable の示す line 数が v_addr_time_mif を超えると、その frame の read/write はそれ以上行わず、次の mif_rst 入力を待つ。

6.2. Internal Bus (COM IF)

COM IF は、SYSIF block から COM block へ memory write/read を要求し、write data や read data の受け渡しを行う IF である。この IF の信号は全て mclk に同期して動作する。

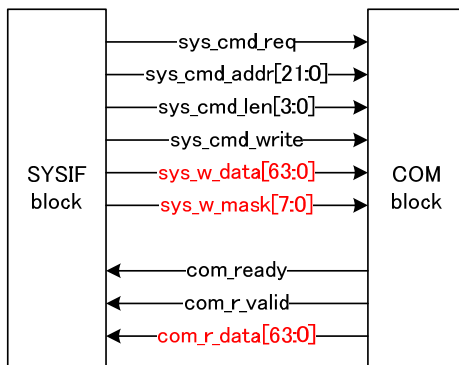


Fig. 11: COM IF

6.2.1. COM IF write

COM IF write 時の基本動作を示す。

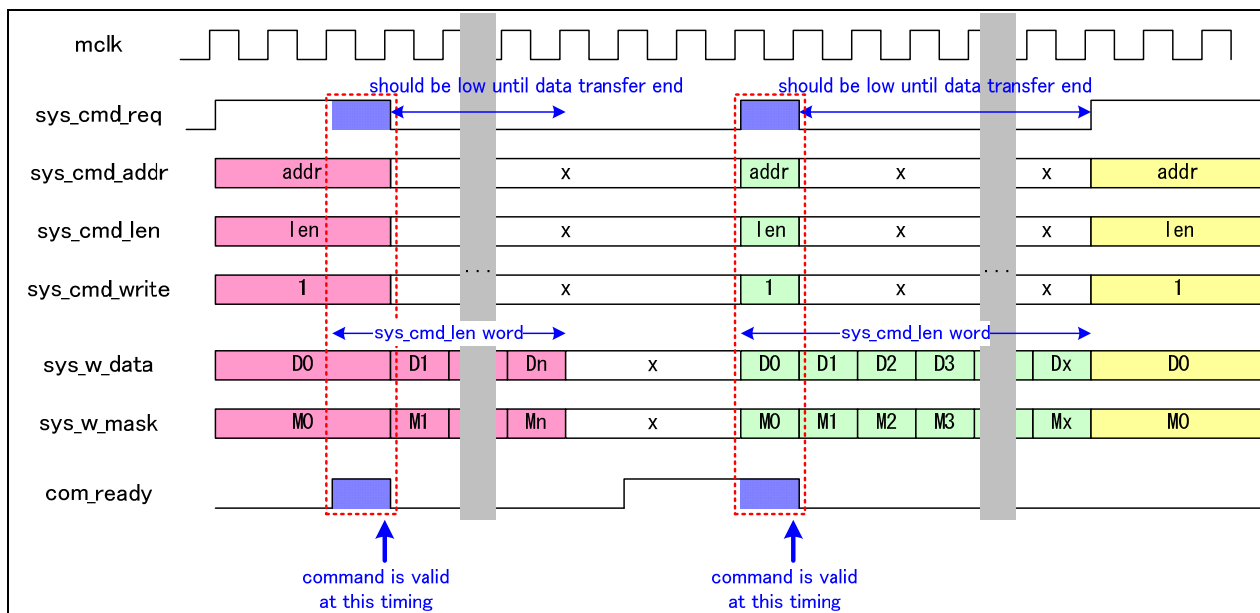


Fig. 12: COM IF write request

sys_cmd_req=1 のとき SYSIF から COM への memory access 要求(command)を示し、そのとき sys_cmd_write=1 であればそれが write command であることを示す。SYSIF は sys_cmd_req=1 と同時に sys_cmd_addr と sys_cmd_len を出力し、また sys_w_data と sys_w_mask は write する最初の word の data と mask を出力する。COM block は com_ready 信号で command を受け取り可能なことを示し、sys_cmd_req=1 かつ com_ready=1 のとき、command は有効となる。

sys_cmd_len は data の長さを示し、1 から 8 までの数値が設定される(それ以外の数値は設定禁止)。command が有効になったサイクルを含めて sys_cmd_len サイクルの間、SYSIF は連続して data と mask を出力する。その data 転送の期間中、sys_cmd_req=0 でなくてはならない。

sys_w_mask は sys_w_data の 1Byte ごとの data mask である (sys_w_mask[0] が sys_w_data[7:0] の mask、sys_w_mask[1] が sys_w_data[15:8] の mask...となる)。mask=0 のとき対応する data の byte は write され、mask=1 のときは write されない。

6.2.2. COM IF read

COM IF read 時の基本動作を示す。

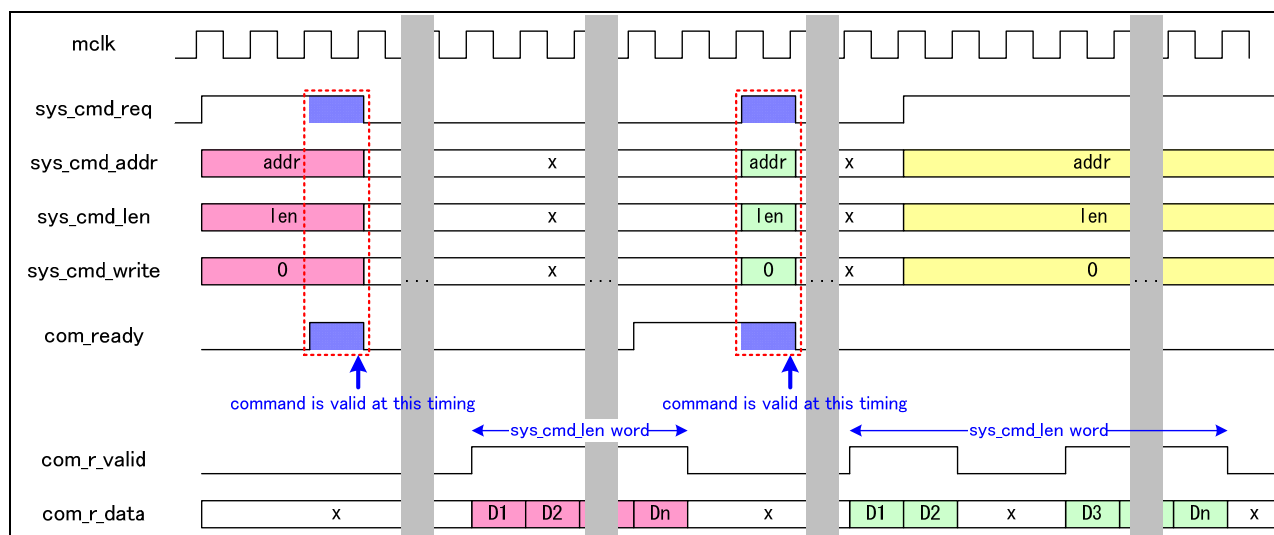


Fig. 13: COM IF read request

sys_cmd_req=1 のとき、sys_cmd_write=0 であればそれが read command であることを示す。Write 時と同様、SYSIF は sys_cmd_req=1 と同時に sys_cmd_addr と sys_cmd_len を出力し、sys_cmd_req=1 かつ com_ready=1 のとき、command は有効となる。

COM は受け取った command に従って memory read を行った後、com_r_valid=1 と一緒に read data (com_r_data) を出力する。read data 出力は interrupt されることもあり(間に com_r_valid=0 の期間が入る)、com_r_valid=1 の期間の合計サイクル数が sys_cmd_len cycle になったときにその read command は完了となる。また SYSIF は read command 出力後、その read data の受け取りを待たずに次の command を出力することができる。

6.2.3. COM IF data and DDR IF data

COM IF 入力の sys_w_data と sys_w_mask 信号が DDR IF へ出力されるとき、下位の 2byte 分が最初に出 force され、上位側へと順に出力されていく。DDR IF からの read data 入力は、最初に入力された 2byte 分が com_r_data の下位側へ置かれ、そこから順に上位側へと詰められていく。

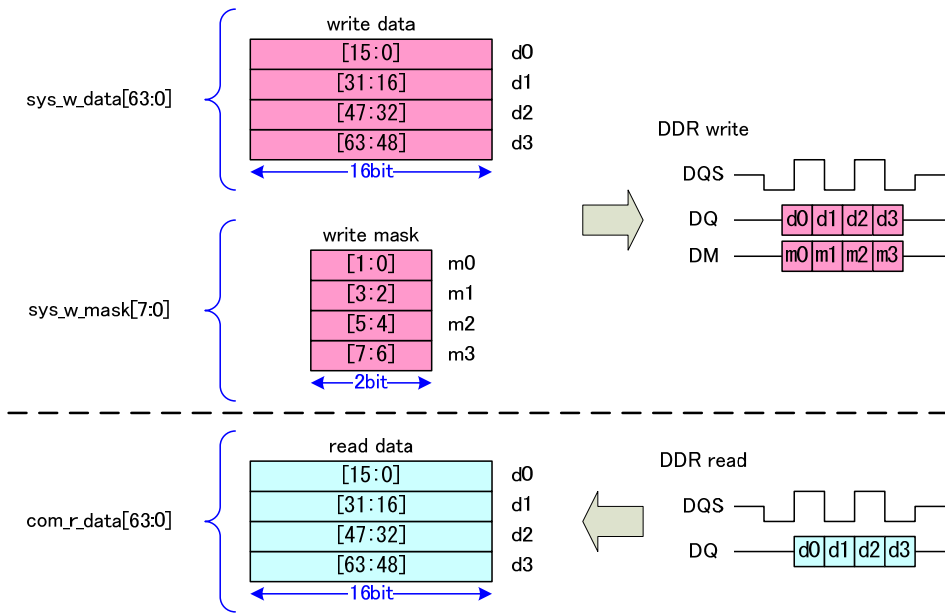


Fig. 14: COM IF data and DDR IF data

6.3. PHY IF

PHY IF は、L_MIF と DDR2 PHY との間で DDR2 SDRAM コマンド、write data、read data、write/read に必要な制御信号を受け渡す IF である。この IF の信号は全て mclkx2 に同期して動作する。

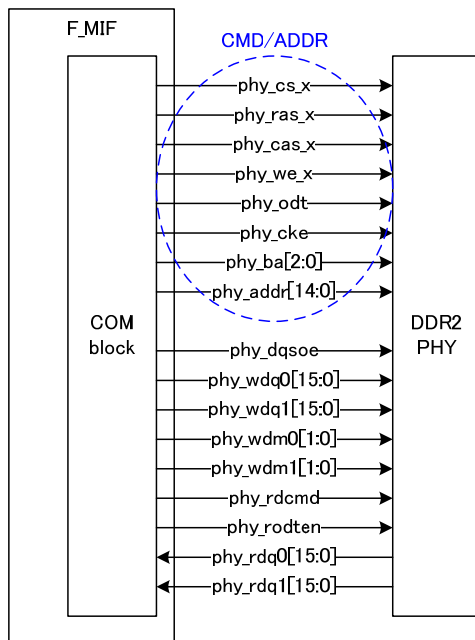


Fig. 15: PHY IF

CMD/ADDR ポートの信号は、JEDEC STANDARD(JESD79-2E)に従って適切な command/address の制御を行う。以下、write/read 時の CMD/ADDR ポート以外の信号について説明する。

6.3.1. PHY IF write

PHY IF write 時の動作を示す。

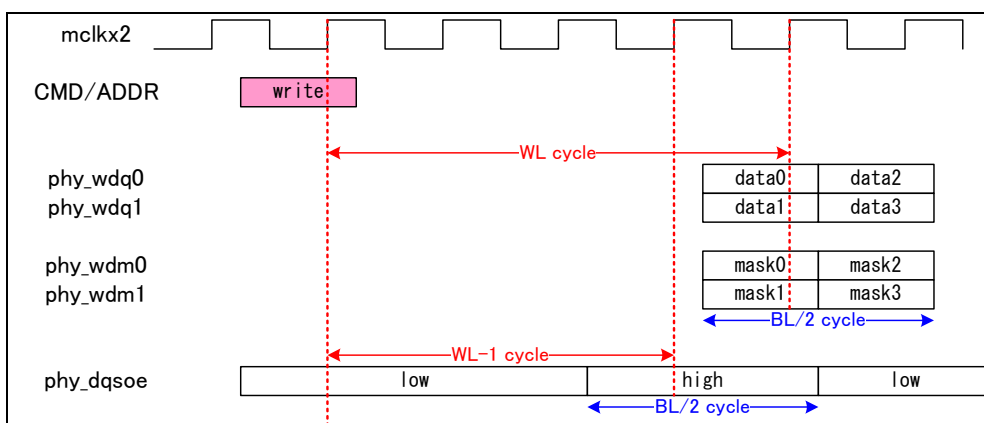


Fig. 16: PHY IF write (WL=4, BL=4)

phy_wdq0/1 と phy_wdm0/1 は write command 出力の WL cycle 後から BL/2 cycle の期間、有効な write data/mask を出力する。phy_dqsoe 信号は write command 出力の(WL-1) cycle 後から BL/2 cycle の期間は high 出力とし、その他の期間は low 出力とする。

6.3.2. PHY IF read

PHY IF read 時の動作を示す。

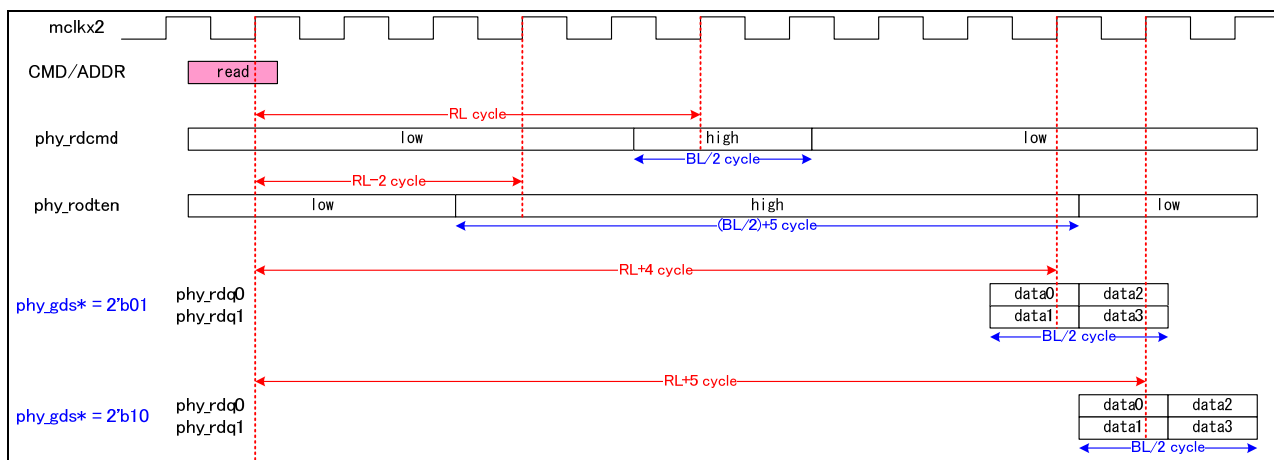


Fig. 17: PHY IF read (RL=5, BL=4)

`phy_rdcmd` 信号は read command 出力の `RL` cycle 後から `BL/2` cycle の期間は high 出力とし、その他の期間は low 出力とする。`phy_rodten` 信号は read command 出力の `(RL-2)` cycle 後から `(BL/2)+5` cycle の期間は high 出力とし、その他の期間は low 出力とする。

read data (`phy_rdq0/1`)は read command 出力後、`phy_gds0/1/2/3=2'b01` 設定のときは `(RL+4)` cycle 後、`phy_gds0/1/2/3=2'b10` 設定のときは `(RL+5)` cycle 後のタイミングで `L_MIF` 側でラッチすることができる。

6.4. REG IF

REG IF は、register bus (`regbus`)から REG block 内の register へ write/read を行うための IF である。この IF の信号は全て `mclk` に同期して動作する。

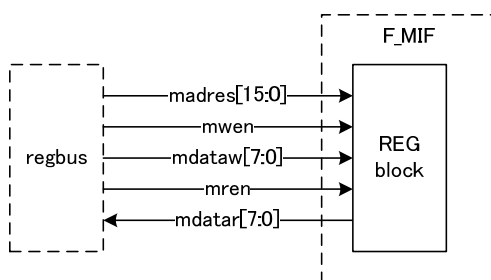


Fig. 18: REG IF

`mwen=1` のとき write が行われ、そのときの `mdataw` の値 (write data) が `mdr` の指す register へ書き込まれる。`mren=1` のとき read が行われ、そのときの `mdr` の指す register から read data が読み出され、read data は `mren=1` の次のサイクルで `mdatar` へ出力される。`mdatar` の値は次の read が行われるまでその値を保持する。また不正なアドレスからの読み出しが行われた場合、`mdatar` は 0 を出力する。

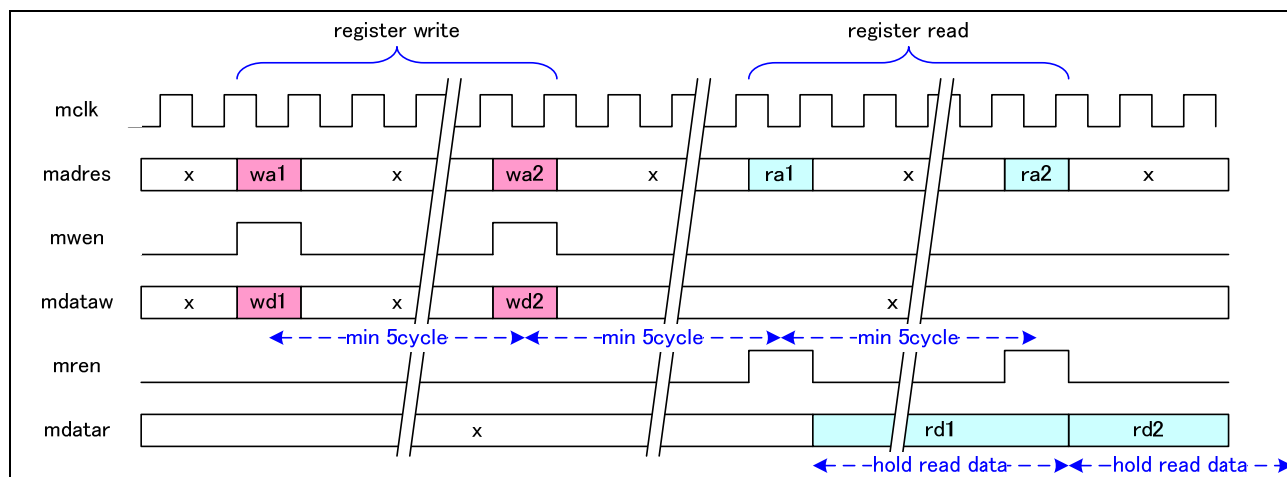


Fig. 19: REG IF write and read

register read/write の interval は minimum 5 cycle@mclk で、burst access は行われたい。

6.5. SRAM IF

SRAM IF は SRAM のデータシートを参照。

6.6. SYSIF register IF

SYSIF register IF の信号リストを示す。

Table 19: SYSIF register IF

Signal name	Bit	From	To	Note
tst_mode	3	REG	SYSIF	directly connected from parameter register in REG block.
tst_mode_force	1	REG	SYSIF	directly connected from parameter register in REG block.
tst_cmd_req	1	REG	SYSIF	directly connected from parameter register in REG block.
tst_cmd_status	1	SYSIF	REG	directly connected to parameter register in REG block.
tst_cmd_mode	4	REG	SYSIF	directly connected from parameter register in REG block.
tst_cmd_addr	22	REG	SYSIF	directly connected from parameter register in REG block.
tst_wd0	16	REG	SYSIF	directly connected from parameter register in REG block.
tst_wd1	16	REG	SYSIF	directly connected from parameter register in REG block.
tst_wd2	16	REG	SYSIF	directly connected from parameter register in REG block.
tst_wd3	16	REG	SYSIF	directly connected from parameter register in REG block.
tst_rd0	16	SYSIF	REG	directly connected to parameter register in REG block.
tst_rd1	16	SYSIF	REG	directly connected to parameter register in REG block.
tst_rd2	16	SYSIF	REG	directly connected to parameter register in REG block.
tst_rd3	16	SYSIF	REG	directly connected to parameter register in REG block.
err_hold	1	REG	SYSIF	directly connected from parameter register in REG block.
err_even	24	SYSIF	REG	directly connected to parameter register in REG block.
err_odd	24	SYSIF	REG	directly connected to parameter register in REG block.

tst_cmd_status は tst_cmd_req の立上りで set され、test command の終了後に 0 に clear される。(tst_cmd_req_clear による tst_cmd_req 信号の clear function は削除。)

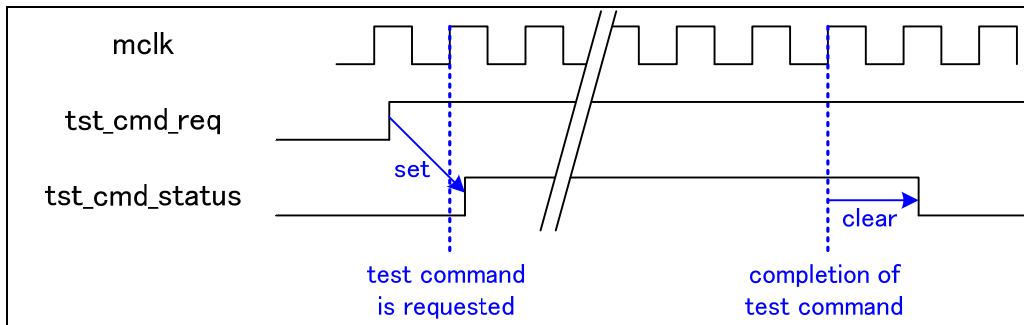


Fig. 20: tst_cmd_status

6.7. COM register IF

COM register IF の信号リストを示す。

Table 20: COM register IF

Signal name	Bit	From	To	Note
com_enable	1	REG	COM	directly connected from parameter register in REG block.
com_status	1	COM	REG	directly connected to parameter register in REG block.
ddr_cabit	1	REG	COM	directly connected from parameter register in REG block.
ddr_8bank	1	REG	COM	directly connected from parameter register in REG block.
ddr_t_rcd	3	REG	COM	directly connected from parameter register in REG block.
ddr_t_rp	3	REG	COM	directly connected from parameter register in REG block.
ddr_t_rc	5	REG	COM	directly connected from parameter register in REG block.
ddr_t_ras_min	5	REG	COM	directly connected from parameter register in REG block.
ddr_t_rrd	3	REG	COM	directly connected from parameter register in REG block.
ddr_t_rtp	2	REG	COM	directly connected from parameter register in REG block.
ddr_t_mrd	3	REG	COM	directly connected from parameter register in REG block.
ddr_t_rfc	8	REG	COM	directly connected from parameter register in REG block.
ddr_t_refi	12	REG	COM	directly connected from parameter register in REG block.
ddr_t_wtr	2	REG	COM	directly connected from parameter register in REG block.
ddr_t_rtw_add	2	REG	COM	directly connected from parameter register in REG block.
ddr_auto_init_en	1	REG	COM	directly connected from parameter register in REG block.
ddr_aref_en	1	REG	COM	directly connected from parameter register in REG block.
ddr_aref_num	4	REG	COM	directly connected from parameter register in REG block.
ddr_set_trefi	1	REG	COM	directly connected from parameter register in REG block.
drcmd_req	1	REG	COM	directly connected from parameter register in REG block.
drcmd_cmd	4	REG	COM	directly connected from parameter register in REG block.
ddr_mr	16	REG	COM	directly connected from parameter register in REG block.
ddr_emr1	16	REG	COM	directly connected from parameter register in REG block.
ddr_emr2	16	REG	COM	directly connected from parameter register in REG block.
ddr_emr3	16	REG	COM	directly connected from parameter register in REG block.
phyif_wen_lat	3	REG	COM	directly connected from parameter register in REG block.
phyif_wodt_lat	3	REG	COM	directly connected from parameter register in REG block.
phyif_wodt_ext	3	REG	COM	directly connected from parameter register in REG block.
phyif_wdata_lat	3	REG	COM	directly connected from parameter register in REG block.
phyif_ren_lat	3	REG	COM	directly connected from parameter register in REG block.
phyif_rodt_lat	3	REG	COM	directly connected from parameter register in REG block.
phyif_rodt_ext	3	REG	COM	directly connected from parameter register in REG block.
phyif_wodt_en	1	REG	COM	directly connected from parameter register in REG block.
phyif_rdlat	3	REG	COM	directly connected from parameter register in REG block.

(drcmd_req_clear による drcmd_req 信号の clear function は削除。)

6.8. PHY register IF

PHY register IF の信号リストを示す。

Table 21: PHY register IF

Signal name	Bit	From	To	Note
phy_rst_x	1	REG	PHY	directly connected from parameter register in REG block.
phy_ddr1	1	REG	PHY	directly connected from parameter register in REG block.
phy_cal_en	1	REG	PHY	directly connected from parameter register in REG block.
phy_comp_sel	1	REG	PHY	directly connected from parameter register in REG block.
phy_dip	5	REG	PHY	directly connected from parameter register in REG block.
phy_din	5	REG	PHY	directly connected from parameter register in REG block.
phy_clkoen	1	REG	PHY	directly connected from parameter register in REG block.
phy_cmdoen	1	REG	PHY	directly connected from parameter register in REG block.
phy_gds	2	REG	PHY COM	directly connected from parameter register in REG block.
phy_sio	1	REG	PHY	directly connected from parameter register in REG block.
phy_dqie	1	REG	PHY	directly connected from parameter register in REG block.
phy_odtmd	2	REG	PHY	directly connected from parameter register in REG block.
phy_dllsel	3	REG	PHY	directly connected from parameter register in REG block.
phy_fldo	1	PHY	REG	directly connected to parameter register in REG block.
phy_dllpdn	1	REG	PHY	directly connected from parameter register in REG block.
phy_dllfrange	2	REG	PHY	directly connected from parameter register in REG block.
phy_msdlly	2	REG	PHY	directly connected from parameter register in REG block.
phy_dmyodten	1	REG	PHY	directly connected from parameter register in REG block.

PHY register IF の信号は全て REG block 内の parameter register から(または parameter register へ)直接接続されている。

phy_gds 信号は、PHY block だけでなく COM block へも入力される。

6.9. UFT register IF

Table 22: UFT register IF

Signal name	Bit	From	To	Note
p_l_uft_phy_on	1	REG	UFT	directly connected from parameter register in REG block.
p_l_phy_tpg_on	1	REG	UFT	directly connected from parameter register in REG block.
p_l_phy_tpg_start	13	REG	UFT	directly connected from parameter register in REG block.
p_l_phy_tpg_end	13	REG	UFT	directly connected from parameter register in REG block.
p_l_phy_al_delay	1	REG	UFT	directly connected from parameter register in REG block.
p_l_phy_gds_delay	1	REG	UFT	directly connected from parameter register in REG block.
p_l_phy_tpg_seed	8	REG	UFT	directly connected from parameter register in REG block.
p_l_phy_crc_on	1	REG	UFT	directly connected from parameter register in REG block.
p_l_phy_crc_trig	13	REG	UFT	directly connected from parameter register in REG block.
p_l_phy_crc_start	13	REG	UFT	directly connected from parameter register in REG block.
p_l_phy_crc_end	13	REG	UFT	directly connected from parameter register in REG block.
p_l_phy_crc_result	32	UFT	REG	directly connected to parameter register in REG block.
p_uft_lvds_run	1	REG	TOP	directly connected from parameter register in REG block.
p_uft_lvds_ch_en	4	REG	TOP	directly connected from parameter register in REG block.

p_uft_lvds_vsz_all	13	REG	TOP	directly connected from parameter register in REG block.
p_uft_lvds_vsz_act	13	REG	TOP	directly connected from parameter register in REG block.
p_uft_lvds_vpos	13	REG	TOP	directly connected from parameter register in REG block.
p_uft_lvds_hsz_all	13	REG	TOP	directly connected from parameter register in REG block.
p_uft_lvds_hsz_act	13	REG	TOP	directly connected from parameter register in REG block.
p_uft_lvds_hpos	13	REG	TOP	directly connected from parameter register in REG block.
p_uft_lvds_seed	10	REG	TOP	directly connected from parameter register in REG block.

7. Normal operation mode specification

7.1. Start-up sequence

ここでは、電源/クロックが安定した状態で L_MIF リセットを解除後、REG IF を通して設定する必要がある start-up sequence について説明する。

L_MIF start-up sequence は PHY start-up 部分と MIF setting/DDR initialization 部分の 2 つに分けられる。

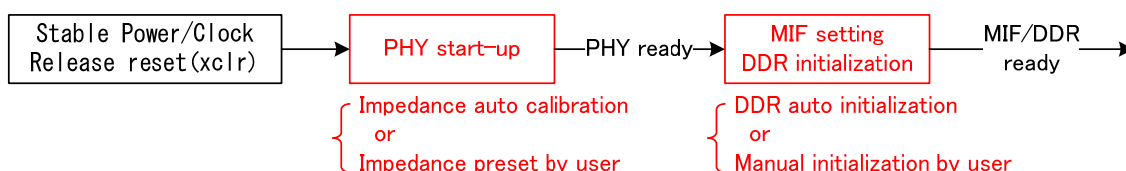


Fig. 21: L_MIF start-up sequence

まず最初に PHY start-up を行い PHY が ready 状態になった後で MIF setting/DDR initialization を行う。PHY start-up には Impedance auto calibration と Impedance preset by user の 2 通りの方法がある。DDR initialization には Auto initialization と Manual initialization の 2 通りの方法がある。以下、これらの詳細について説明する。

7.1.1. PHY start-up

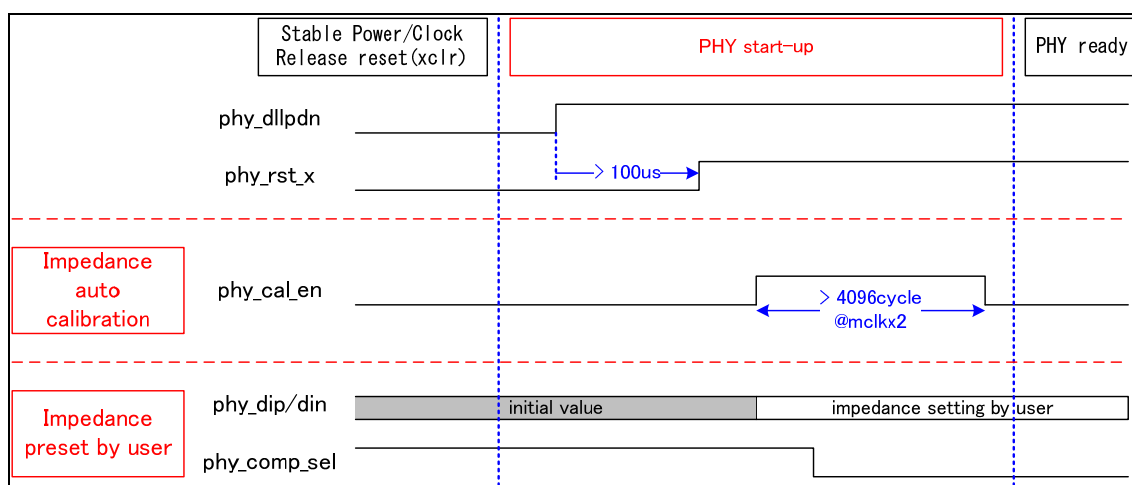


Fig. 22: PHY start-up

PHY start-up では、まず phy_dllpdn 信号を low から high へと変化させる。その後 100us 以上待ってから phy_rst_x 信号を 0 から 1 へと変化させる。

Impedance auto calibration の場合、phy_rst_x を high とした後で phy_cal_en を 4098cycle@mclkx2 以上の期間 high にした後、low へ戻す。これで PHY は ready 状態となる。

Impedance preset by user の場合、phy_rst_x を high とした後で phy_dip/phy_din に所望の値を設定する。その後、phy_comp_sel=low とすることで PHY は ready 状態となる。

7.1.2. MIF setting and DDR initialization

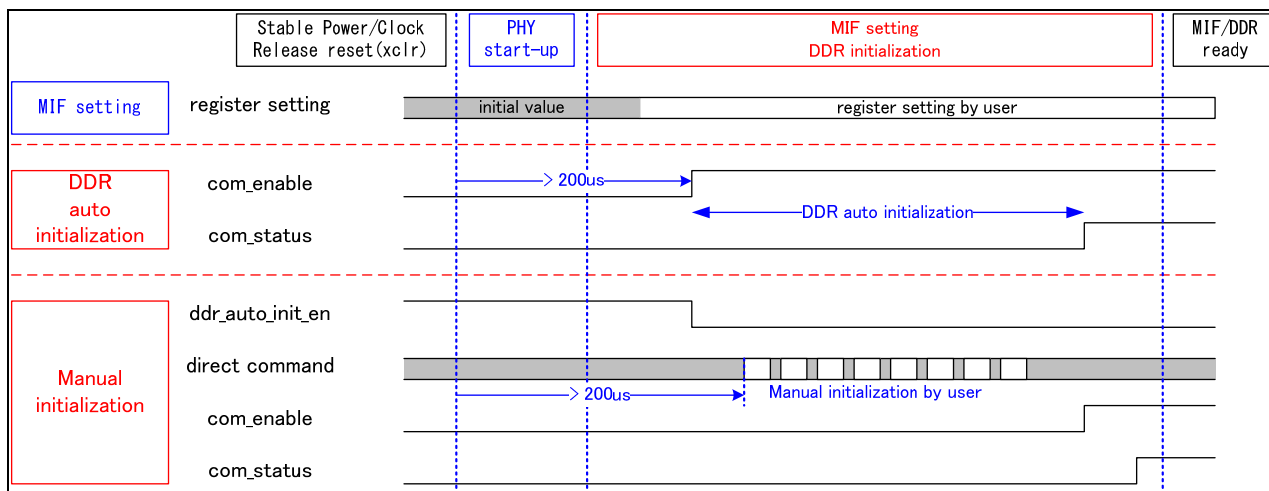


Fig. 23: MIF setting/DDR initialization

PHY start-up が完了した後、SYSIF/COM/PHY の各 register に必要な設定を行う(初期値から変更する必要がある場合)。その後、DDR initialization を下記の通り行う。

- Auto initialization

DDR auto initialization の場合、com_enable を low から high へと変化させることで PHY IF へ DDR initialization command が自動的に発行される。この自動発行される command sequence の詳細は下図を参照。

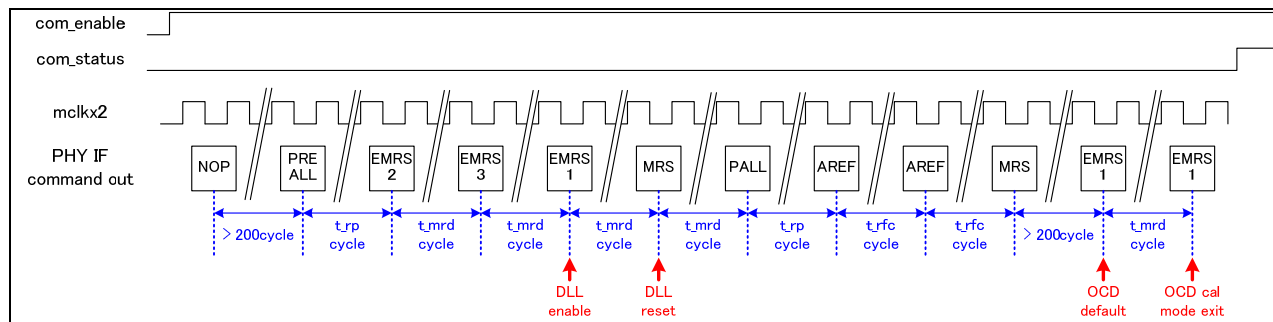


Fig. 24: DDR auto initialization sequence

Auto initialization が完了すると com_status=high となり、MIF/DDR が ready 状態であることを示す。Auto initialization 時の MR/EMR 設定値にはレジスタ設定値(DDR_MR, DDR_EMRS1, DDR_EMRS2, DDR_EMRS3)が使われるが、DDR_MR[8](DLL reset)と DDR_EMRS1[9:7](OCD cal program)のビットはレジスタ設定によらず自動的に制御される。

- Manual initialization

Manual initialization の場合、まず ddr_auto_init_en=low とする。その後、direct command 機能を使って user が initialization command sequence の発行を行う。このときの MR/EMR 設定値にはレジスタ設定値(DDR_MR, DDR_EMRS1, DDR_EMRS2, DDR_EMRS3)がそのまま使用されるので、DDR_MR[8](DLL reset)や DDR_EMRS1[9:7](OCD cal program)の値もユーザにより適宜設定される必要がある。Manual initialization 完了後、user は com_enable を low から high へと変化させ、com_status=high となったところで MIF/DDR は ready 状態となる。

7.2. Write and read operation

SYSIF の frame write/read 時の動作について説明する。

7.2.1. basic operatoin

System IF から入力された data は write line buffer に格納され、1line 分の data が buffer にたまとその buffer data は memory write request と一緒に COM IF へ出力される。また memory read request により COM IF から read された data は read line buffer に格納され、frame-rw request により System IF へ出力される。SYSIF は 1line 分の write line buffer と 2line 分の read line buffer を持ち、2 つの read line buffer は交互に使用される。(mif_rst 入力後、最初の read line は read line buffer0 へ格納される)

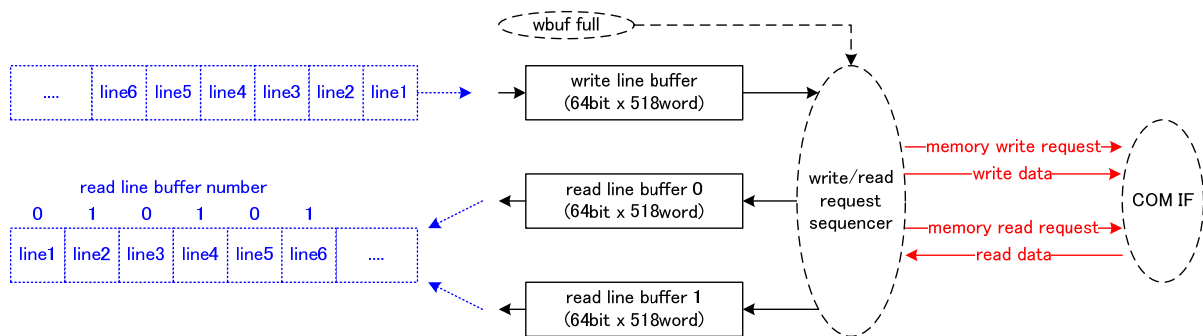


Fig. 25: basic operation image

mif_rst 入力後、memory から 2line 分のデータを読み出し、read line buffer0/1 へそれぞれ格納する。frame-rw request (mif_enable)が入力されたとき、wdata を write line buffer へ書き込みながら同時に read line buffer から rdata への出力を行う。1line 分の wdata が write line buffer へ格納された後、すぐにその write line buffer の内容の memory write を行う。その完了後すぐに 1line 分の memory read を行い、空いている方の read line buffer へ格納する。

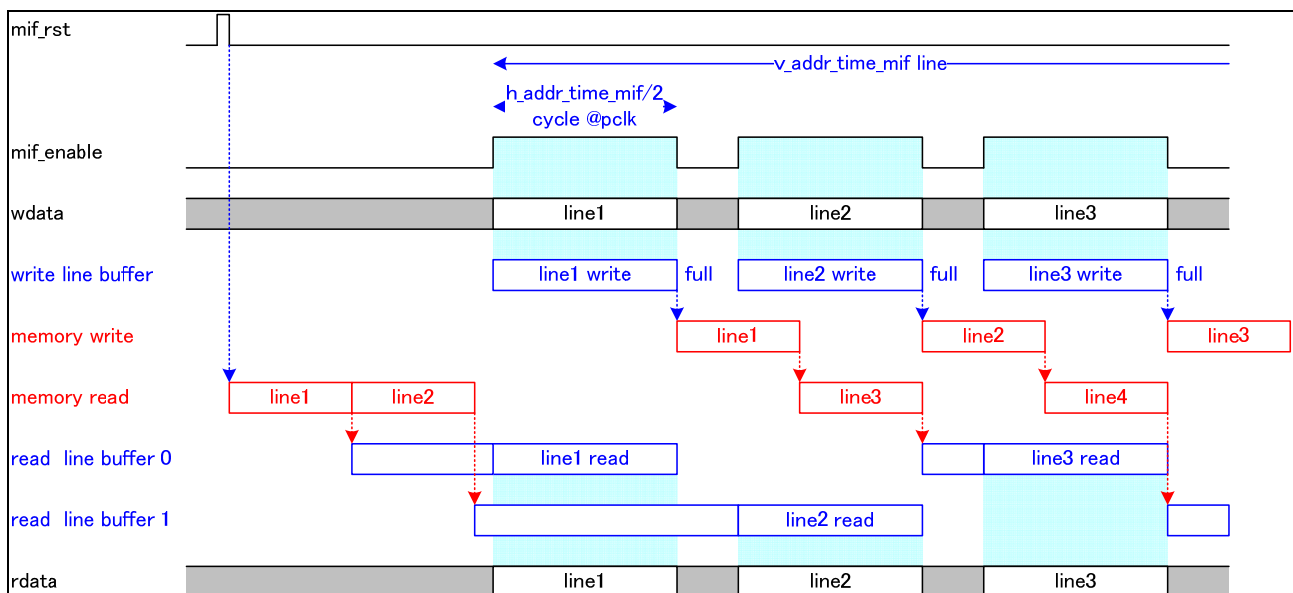


Fig. 26: basic operation

最初の 2line 分の read 以外は常に同じ動作を繰り返し、write line buffer の full 状態により発生する memory

write と memory read の順番は入れ替わらない。

mif_enable=1 の期間が h_addr_time_mif/2 cycle@pclk より短いときは、mif_enable の立下りまでを 1line とする。mif_enable=1 の期間が h_addr_time_mif/2 cycle@pclk を超えるときは、h_addr_time_mif/2 cycle までを 1line としてそれを超える部分は無視する。また mif_enable の回数(line 数)が v_addr_time_mif を超える場合も、超過した部分は無視して memory access は行わない。

7.2.2. bit width conversion

下の図のとおり、System IF の wdata/rdata のビット幅は 32bit であるが、write/read line buffer のビット幅および COM IF の sys_w_data/com_r_data のビット幅は 64 ビットである。

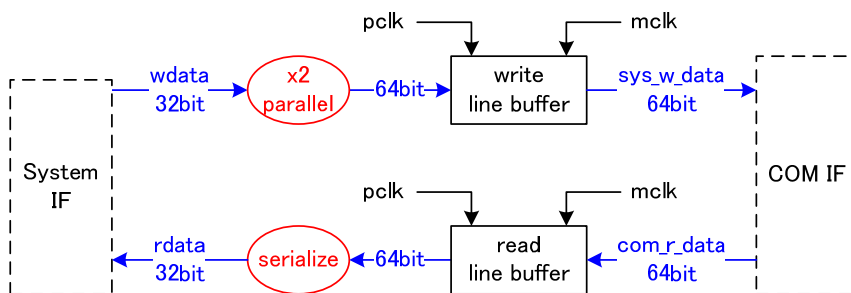


Fig. 27: bit width conversion

wdata を write line buffer へ書き込むときは 2 並列化してビット幅を 64bit に揃えてから write を行い、read line buffer から読み出した 64bit のデータは直列化して 32bit に揃えてから rdata へ出力する。

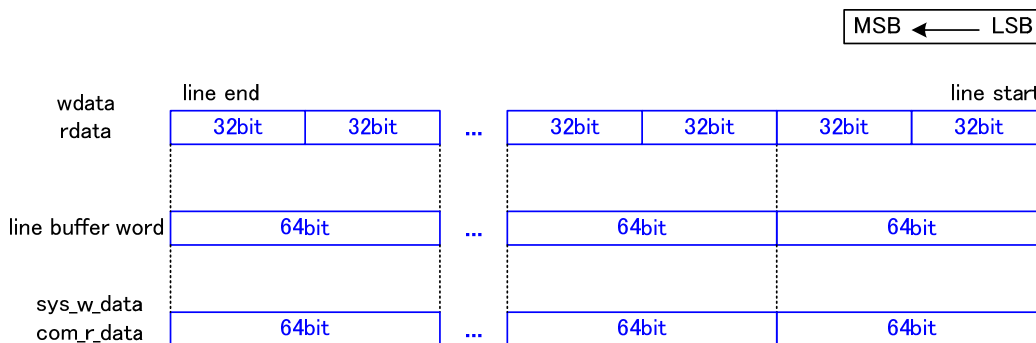


Fig. 28: word alignment

7.2.3. memory write/read request

SYSIF は、System IF からの frame-rw request に応じて COM IF へその line の memory write request および memory read request を発行する。sys_cmd_addr[21:0]はその request の start address を示し、sys_cmd_len[3:0]は start address から連続して write/read を行う data (sys_w_data または com_r_data) のワード長を示す。sys_cmd_len[3:0]の最小値は 1、最大値は 8 である。

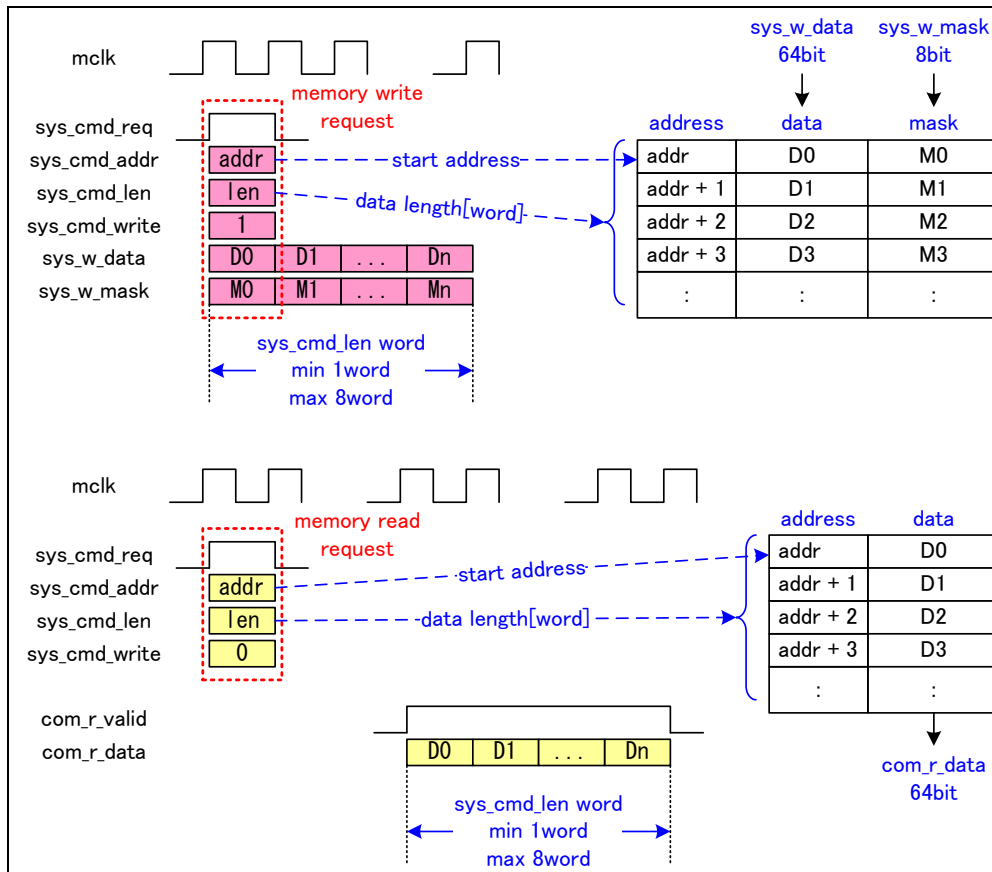


Fig. 29: memory write/read request

以下、memory write/read request のルールについて説明する。

- 1line 分の write/read は、h_addr_time_mif x 32bit のデータを write/read するのに最低限の数の write/read request で実行する
- sys_cmd_addr[21]は frame ID とする
- frame ID は mif_rst の入力毎に 0 と 1 を交互に繰り返す
- write と read の frame ID は常に異なる frame を指し、同時に同じ frame の write と read は行わない
- sys_cmd_addr[20:10]はそのフレーム中の line number とする
- line number は mif_enable が入力されるごとに以下のようにインクリメントされる
 - line 1 sys_cmd_addr[20:10] = 11'h000
 - line 2 sys_cmd_addr[20:10] = 11'h001
 - line 3 sys_cmd_addr[20:10] = 11'h002
 - ⋮
 - line2048 sys_cmd_addr[20:10] = 11'h7FF
- sys_cmd_addr[9:0]はその line 中の 64bit write data の word count とする
- ある line の先頭の word は sys_cmd_addr[9:0]=10'h000 に割り当てられ、それ以降の word は続く address に順に割り当てられる

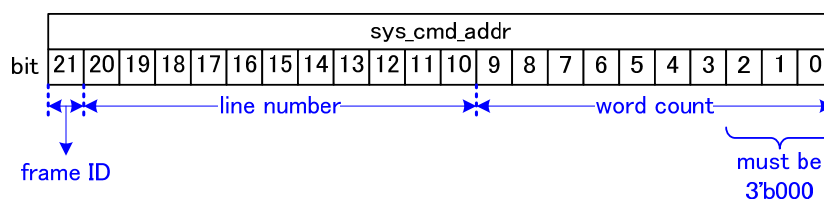


Fig. 30: addressing

- `sys_cmd_addr[2:0]`の値は常に `3'b0` でなければならない
- 1回の request で write/read する word 数は、基本的に `sys_cmd_len=8` とする
- その line の最後で半端な 8word 以下のデータが残ったときのみ、`sys_cmd_len` は 1 から 8 の値とすることができる

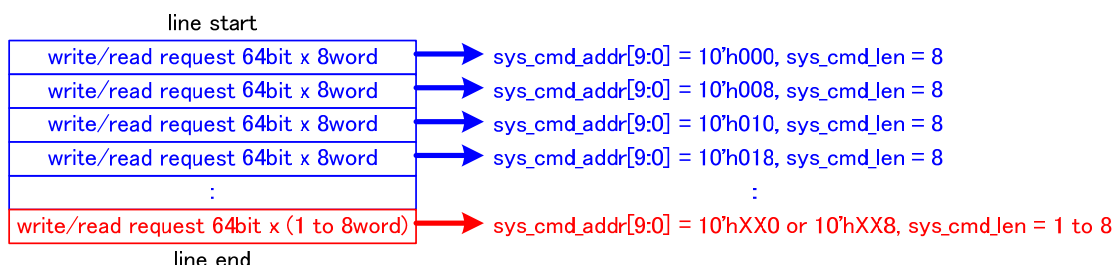


Fig. 31: write request sequence at 1 line write

- `sys_w_mask` は使用しないので全て 0 とする

7.2.4. mif_rst function

`mif_rst` 信号の入力が新しい frame の始まりを示す。`mif_rst` 信号の立下り後は、それまでの状態に関わらず、必ず新しい frame の最初の line (line 1)からの処理が開始される(`sys_cmd_addr[20:0]`は 0 にクリアされ、`sys_cmd_addr[21]`は反転する)。

`mif_rst` 信号の立下りの時点である line の memory access が行われている場合は、その line の途中で memory access を中断するが、既に成立している memory write/read request は完了させる (COM IF プロトコルに違反しない)。

7.3. Arbitration

section 7.2.1 に記載のとおり、SYSIFは規則的な順番でmemory writeとmemory readのarbitrationを行う。このarbitrationはline単位で行われ、あるlineについてmemory access (write or read)が行われているとき、そのlineのmemory accessは他のlineのmemory accessによって中断されない。

7.4. recovery from illegal case

System IF からの入力が illegal な状態となっても、次の `mif_rst` 入力 (の立下り) の後には正しく次の frame の memory write/read を開始することができる状態に復帰する。

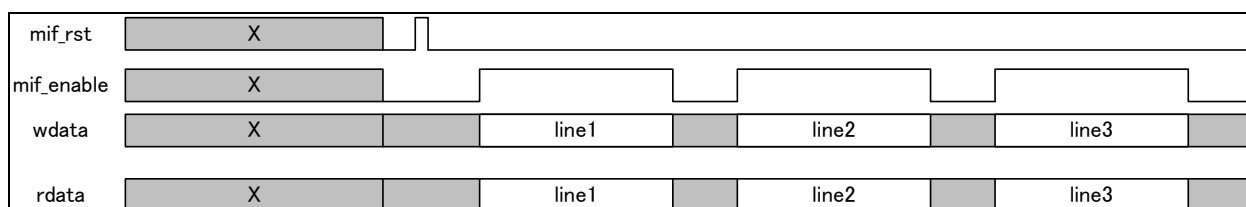


Fig. 32: recovery from illegal case

7.5. frame format change

下の図は frame 3 の write 中(frame 3 の write が開始される mif_rst の立下りから次の mif_rst の立下りまでの間)に frame format(clock frequency を含む)が変更された場合を示す。

このとき、frame 2 と frame 3 については don't care として画像の入出力動作を保証しないが、frame 1 と frame4 についてはその画像の入出力動作を保証する。

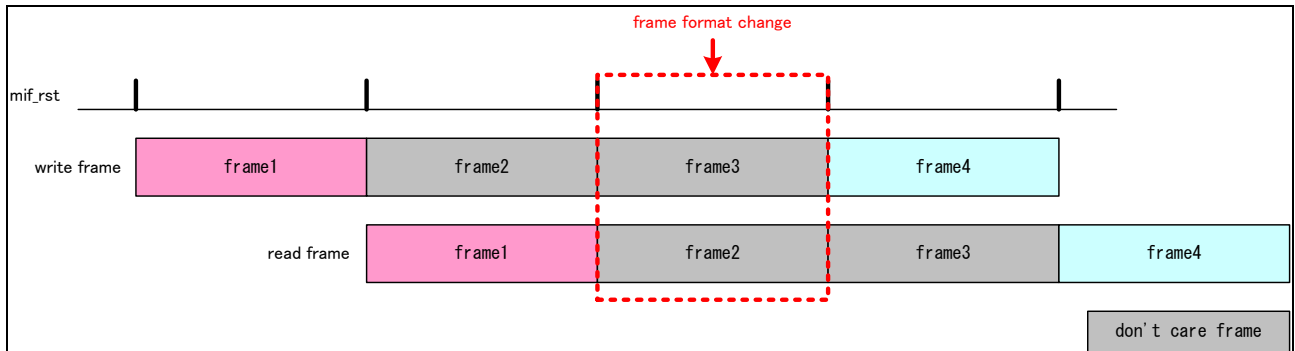


Fig. 33: frame format change

8. Test mode specification

8.1. Memory write test

tst_mode = 1 or 3 のとき、tst_write_en=1 となり memory write test 機能が有効になる。このとき System IF からの frame-rw request による memory write は無効になり、レジスタ IF から memory の任意の場所への write が可能となる。

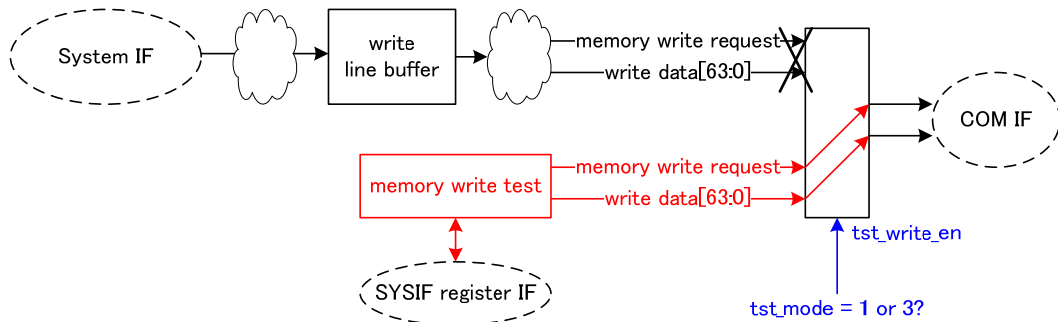


Fig. 34: memory write test

tst_write_en の値は mif_rst の立下りの次のサイクルでラッチされ、そのときの値が 1 であれば、memory write test が有効となる。つまり frame の途中で mode は切り替わらず、次の frame の開始時に mode が切り替えられる。(normal mode に戻るときも同様)

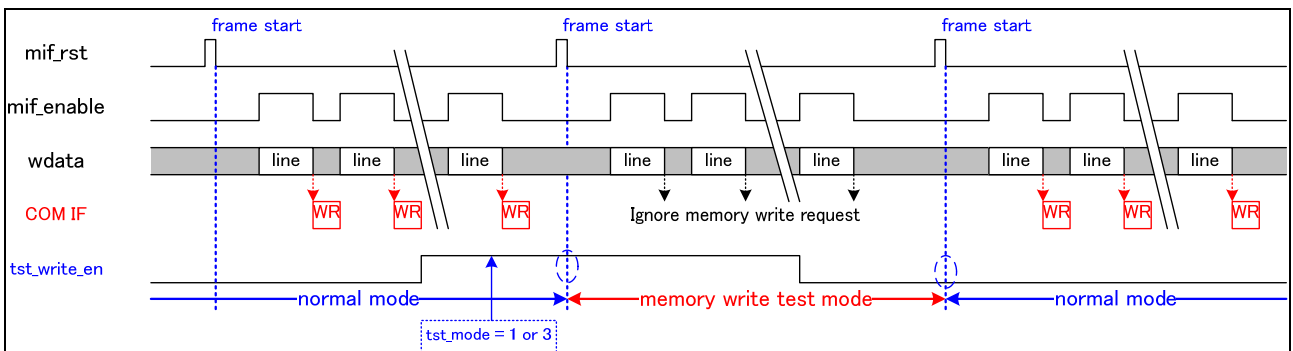


Fig. 35: mode change timing

下の図は write request 生成部分の構成である。

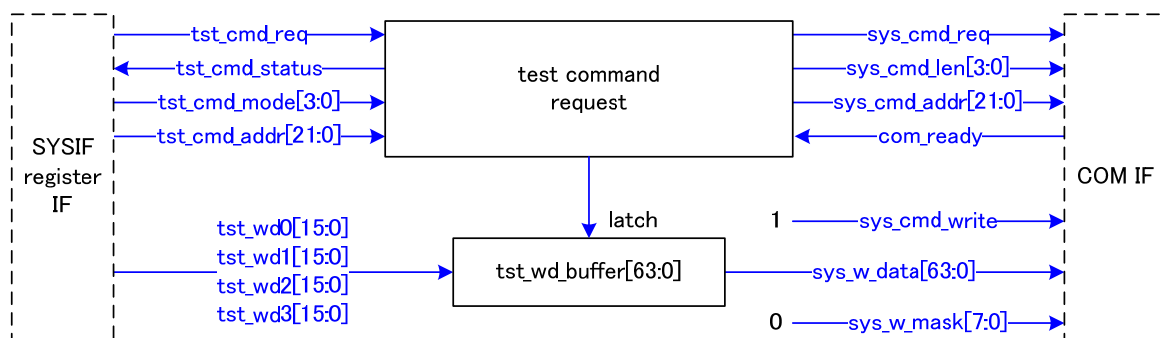


Fig. 36: write request diagram

レジスタ IF から memory write request が発行されたとき、tst_wd0/1/2/3 の値が tst_wd_buffer へラッチされ、その値がそのまま sys_w_data へ出力される。tst_wd_buffer と tst_wd0/1/2/3 のビットの対応は以下の通り。

➤ $tst_wd_buffer = \{ tst_wd3, tst_wd2, tst_wd1, tst_wd0 \}$

tst_cmd_mode=0 でないとき、tst_cmd_req を 0 から 1 へ変化させると memory write 動作が開始される。この tst_cmd_req の立上りをトリガに tst_cmd_mode、tst_cmd_addr、tst_wd0/1/2/3 の設定がラッチされ、COM IF へ出力する memory write request の制御が行われる。

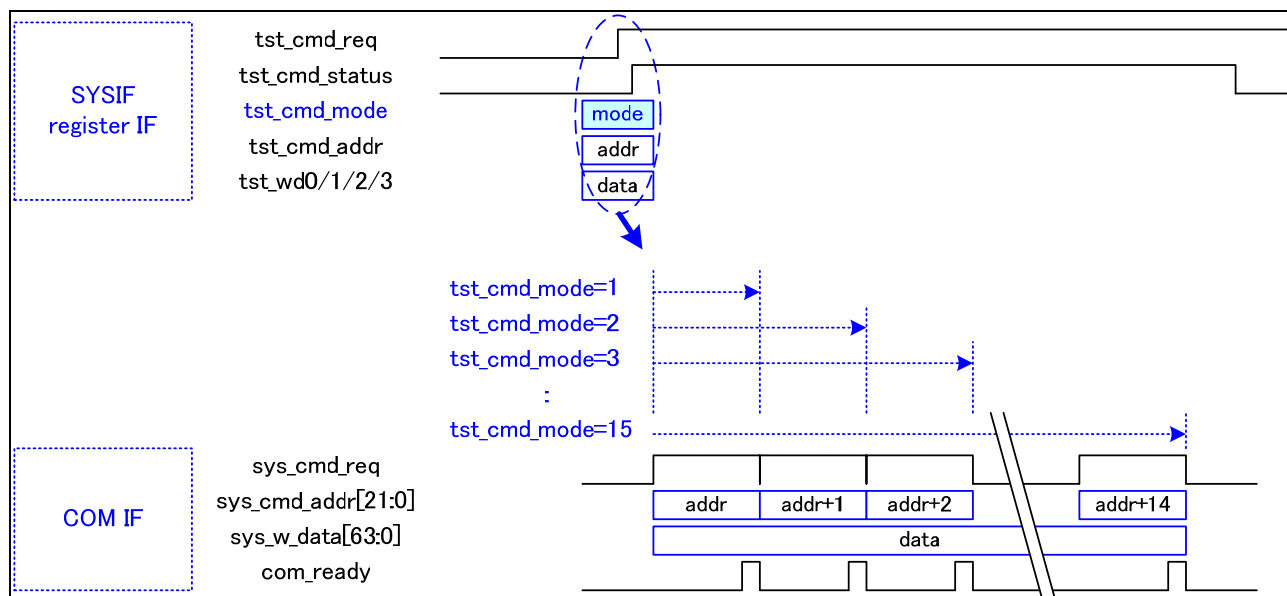


Fig. 37: memory write request

tst_cmd_addr が write request の開始アドレスとなり、tst_wd0/1/2/3 の値が write data となる。tst_cmd_mode の値により開始アドレスから何ワードの write を行うかが指定され、tst_cmd_mode の値(1 から 15)がそのまま memory write を行う word 数となる。複数 word の write の場合でも、write data は同じ値 (tst_wd0/1/2/3 の値)が続けて write される。

tst_cmd_req の立上りで tst_cmd_status=1 となり、memory write の完了後、tst_cmd_status=0 となる。(tst_cmd_req_clear による tst_cmd_req 信号の clear function は削除。)

8.2. Memory read test

tst_mode = 2 or 3 のとき、tst_read_en=1 となり memory read test 機能が有効になる。このとき System IF からの frame-rw request による memory read は無効になり、レジスタ IF から memory の任意の場所からの read が可能となる。

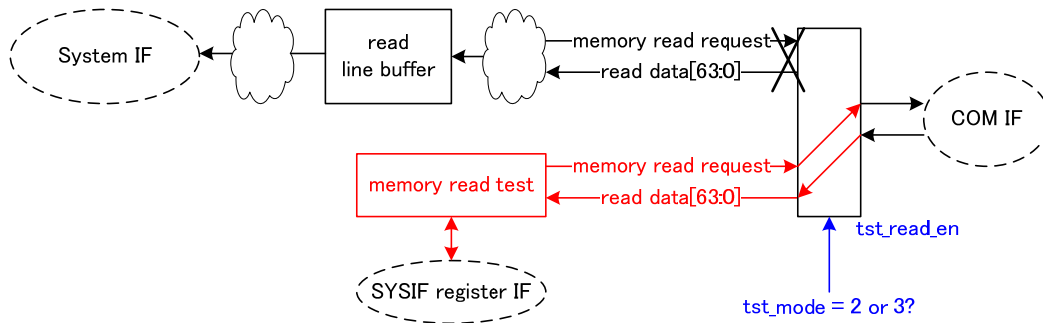


Fig. 38: memory read test

tst_read_en の値は rd_rst の立下りの次のサイクルでラッチされ、そのときの値が 1 であれば、memory read test が有効となる。つまり frame の途中で mode は切り替わらず、次の frame の開始時に mode が切り替えられる。(normal mode に戻るときも同様)

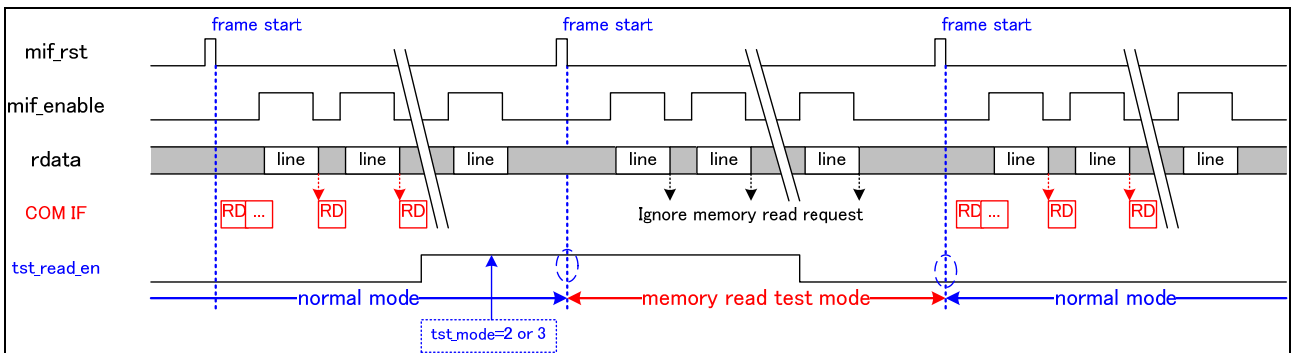


Fig. 39: mode change timing

下の図は read request 生成部分の構成である。

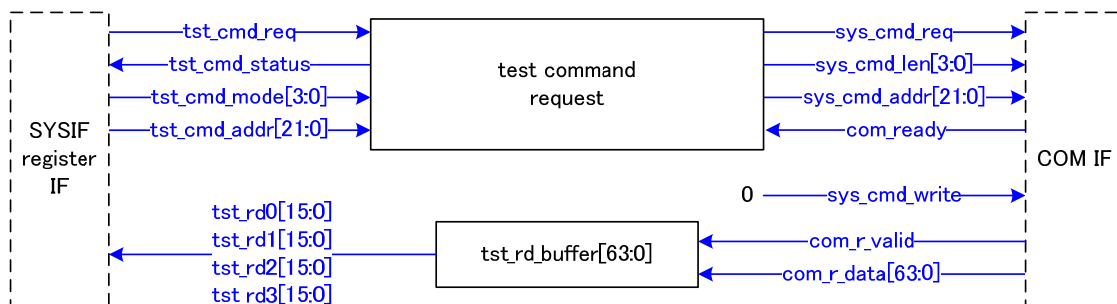


Fig. 40: read request diagram

tst_cmd_mode=0 のとき、tst_cmd_req を 0 から 1 へ変化させると memory read 動作が開始される。この tst_cmd_req の立上りをトリガに tst_cmd_mode、tst_cmd_addr の設定がラッチされる。

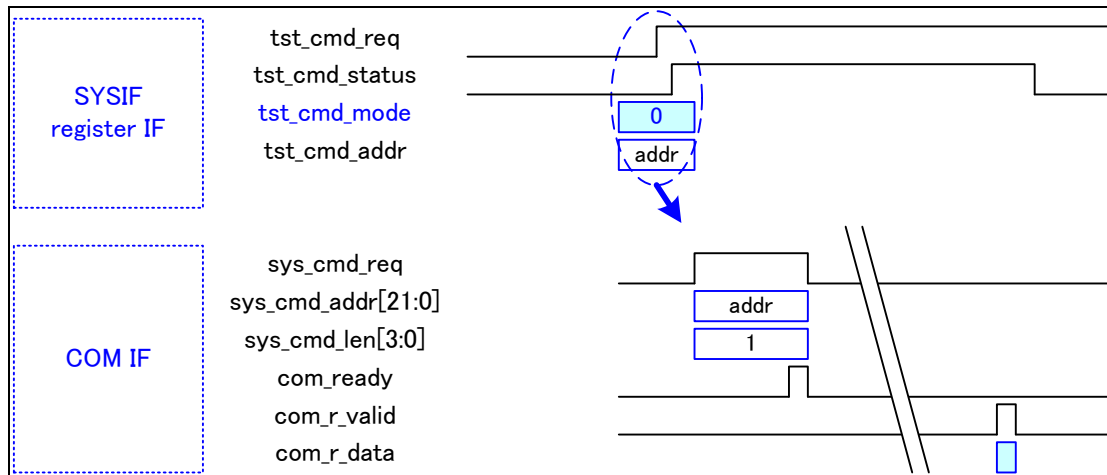


Fig. 41: memory read request

memory read test では、1 回の read request で読み出す word 数は必ず 1word に固定(sys_cmd_len=1)となり、tst_cmd_addr がその読み出しアドレスとなる。read data は tst_rd_buffer[63:0]へ格納され、その値がそのまま tst_rd0/1/2/3 へ出力される。tst_rd_buffer と tst_rd0/1/2/3 のビットの対応は以下の通り。

- tst_rd0 = tst_rd_buffer[15:0]
- tst_rd1 = tst_rd_buffer[31:16]
- tst_rd2 = tst_rd_buffer[47:32]
- tst_rd3 = tst_rd_buffer[63:48]

tst_cmd_req の立上りで tst_cmd_status=1 となり、memory read の完了後、tst_cmd_status=0 となる。(tst_cmd_req_clear による tst_cmd_req 信号の clear function は削除。)

8.3. Parity check mode

tst_mode=4 のとき、parity check mode が有効になる。

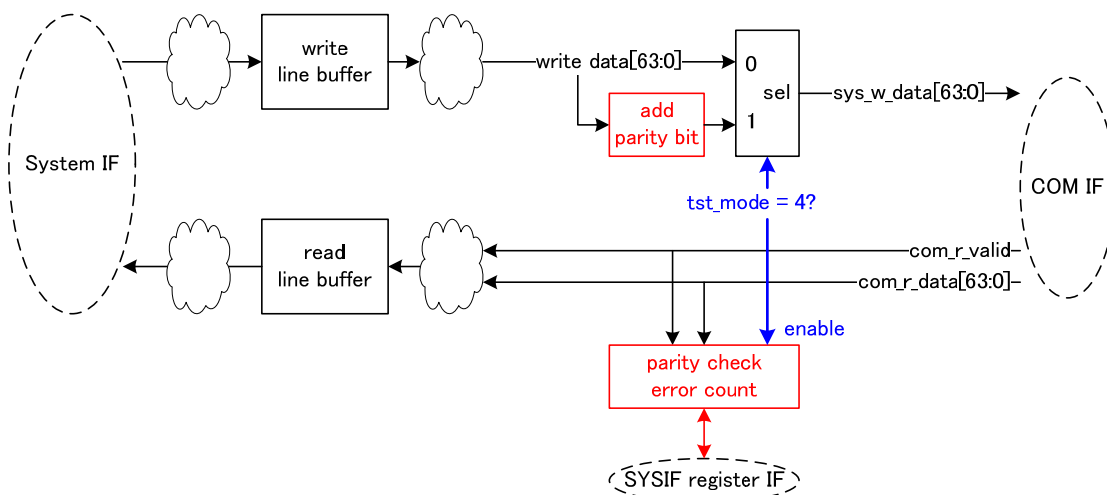


Fig. 42: parity check mode

parity check mode では、System IF は normal operation を続け、memory access も normal mode と同様に

行われる。ただし、memory write 時に write data に parity bit が付加され、memory read 後は read data の parity check が行われ、そのエラー数がカウントされる。

parity bit を付加するときは、write data を 16bit ごとのブロックに分割してそれぞれの上位 15bit の XOR をとり、その結果を parity bit として各ブロックの最下位ビットに加える。そのとき、もとの最下位ビットは捨てられる。

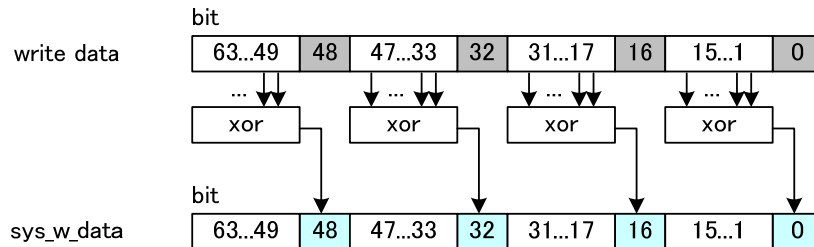


Fig. 43: add parity bit

read data に対しても 16bit のブロックごとに parity check が行われ、エラーがあった場合、エラーカウンタにその数がカウントされる。エラーカウンタは DQ の 16bit ごとに even と odd のカウンタを持ち、com_r_data の各 16 ビットブロックに対して下の図のとおり parity check およびエラーカウントを行う。

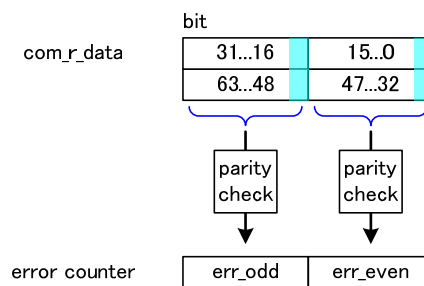


Fig. 44: parity check and error counter

err_hold=0 のとき error counter は mif_rst の立下りごとに更新され、その直前の 1frame 分のエラー数を表示する。err_hold=1 のとき、エラーカウンタは hold され、mif_rst による更新は行われない。

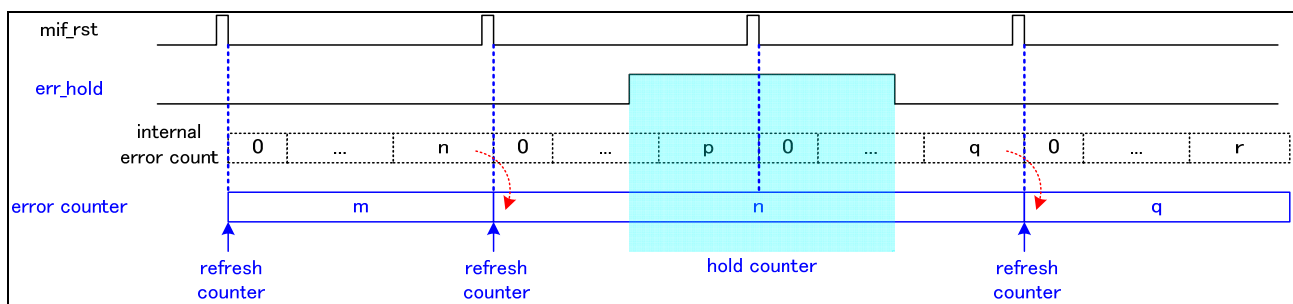


Fig. 45: error hold function

8.4. tst_mode_force function

memory write/read test の切り替えは mif_rst をトリガーに行われるが、それに加えて tst_mode_force 信号の立上りもトリガーとして test mode の切り替えを可能とする。(画像信号の入出力が行われないときにも test mode による memory read/write を可能にするため)

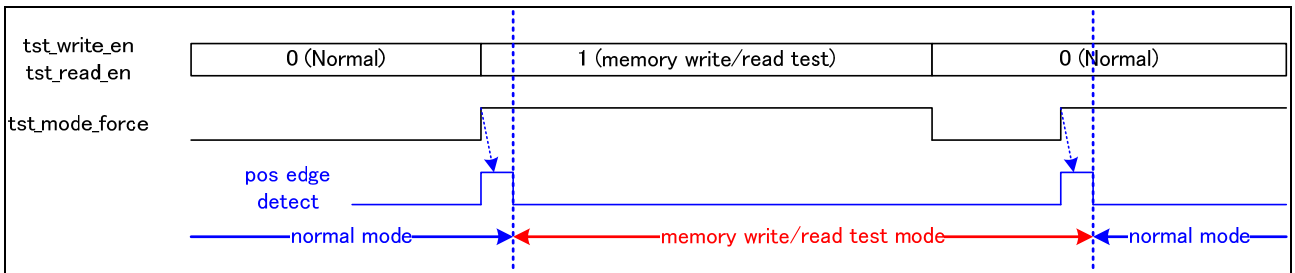


Fig. 46: `tst_mode_force` function

9. COM block specification

9.1. Overview

9.1.1. Function overview

COM block は、COM IF からの memory access request に従い DDR2 SDRAM アクセスに必要なコマンドシーケンスを生成し、PHY IF を通して memory access を行うブロックである。各 register IF からの制御により、DDR initialize、AREF、power-down、self-refresh の制御などを行う。

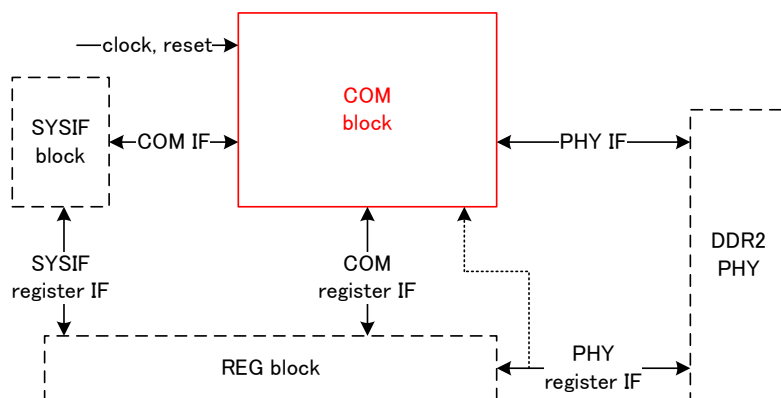


Fig. 47: basic configuration

主な IF は clock/reset, COM IF, PHY IF, COM register IF だが、PHY register IF の信号の一部も入力される。

COM block は COM IF からのリクエストに対して、本仕様および JEDEC JESD79-2E の制約を守り、かつ最大限の効率で memory access を行う (bank interleave 時や write-to-read、read-to-write 切り替え時に無駄なサイクルを挟まない)。

9.1.2. Parameters to be supported

COM block は parameter input signal により、下の表のとおり DDR2 SDRAM の各種パラメータをサポートする。

Table 23: supported parameter 1

Parameter input signal name	Required setting to be supported (decimal)	Unit	Description
ddr_cabit	0	-	Column address bit width 0:CA=9bit 1:CA=10bit
ddr_8bank	0	-	Bank address bit width 0:BA=2bit (4bank device) 1:BA=3bit (8bank device)
ddr_t_rcd	5, 6	cycle@mclkx2	DDR timing parameter: tRCD
ddr_t_rp	5, 6	cycle@mclkx2	DDR timing parameter: tRP
ddr_t_rc	20 to 24	cycle@mclkx2	DDR timing parameter: tRC
ddr_t_ras_min	15 to 18	cycle@mclkx2	DDR timing parameter: tRAS(min)
ddr_t_rrd	4	cycle@mclkx2	DDR timing parameter: tRRD
ddr_t_rtp	3	cycle@mclkx2	DDR timing parameter: tRTP
ddr_t_mrd	2	cycle@mclkx2	DDR timing parameter: tMRD
ddr_t_rfc	25 to 131	cycle@mclkx2	DDR timing parameter: tRFC

ddr_t_refi	1300 to 3120	cycle@mclkx2	DDR timing parameter: tREFI
ddr_t_wtr	3	cycle@mclkx2	DDR timing parameter: tWTR
ddr_t_rtw_add	0, 1	cycle@mclkx2	Extra clock cycle from read command to write command.
ddr_aref_num	1	-	Number of AREF commands consecutively performed after one AREF request. AREF request is issued every period of (ddr_aref_num * ddr_t_refi / 2) cycle @mclk.

また Mode register 設定に関連して以下のパラメータをサポートする。

Table 24: supported parameter 2

Parameter	Required setting to be supported (decimal)	Unit	Description
Burst length	4	-	Indicated by MR setting (ddr_mr)
CAS latency	5, 6	cycle@mclkx2	Indicated by MR setting (ddr_mr)
tWR	5, 6	cycle@mclkx2	Indicated by MR setting (ddr_mr)
Additive latency	0, 1	cycle@mclkx2	Indicated by EMR1 setting (ddr_emr1)

PHY IF signal timing parameter として以下の設定値をサポートする。

Table 25: supported parameter 3

Parameter input signal name	Required setting to be supported (decimal)	Unit	Description
phyif_wen_lat	3 to 5	cycle@mclkx2	PHY IF signal timing setting. Refer to 9.6.9 for detail.
phyif_wodt_lat	1 to 4	cycle@mclkx2	
phyif_wodt_ext	0 to 2	cycle@mclkx2	
phyif_wdata_lat	4 to 6	cycle@mclkx2	
phyif_ren_lat	5 to 7	cycle@mclkx2	
phyif_rodt_lat	2 to 6	cycle@mclkx2	
phyif_rodt_ext	4 to 6	cycle@mclkx2	
phyif_rdlat	2 to 6	cycle@mclkx2	

ここに挙げた設定値は必要最小限のサポート範囲であり、この範囲外の設定値での動作は実装による。

9.2. Basic configuration

9.2.1. Clock specification

COM blockに入力されるクロックは、mclkとmclkx2 の 2 つである。これら 2 つのクロックの仕様は 3.2 節を参照。

9.2.2. Reset specification

COM blockに入力されるリセットは、rst_xの 1 つのみ。リセットの仕様は 3.3 節を参照。

9.2.3. Clock usage

COM block の動作クロックは基本的に mclk であり、DDR command sequence 生成やその他の制御も全て mclk ドメインで行われる。mclkx2 は PHY IF への出力信号を mclk から mclkx2 ドメインへ渡すため(serialization)とPHY IF からの入力信号を mclkx2 から mclk ドメインへ渡すため(parallelization)だけに使用される。

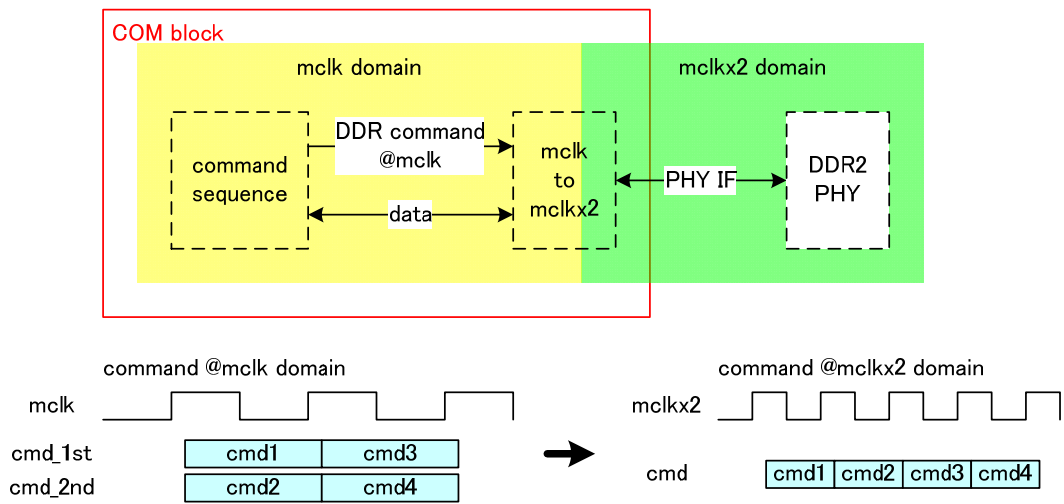


Fig. 48: mclk and mclkx2 usage

9.3. IO pin specification

以下、COM block の IO ピンリストを示す。

9.3.1. Clock, reset

Table 26: clock, reset pin list

Name	Bit num	In Out	Initial value	From	To	Clock domain	Description
mclk	1	In	–	TOP	COM	–	Clock for COM IF, COM register IF and COM block internal logic except for PHY IF.
mclkx2	1	In	–	TOP	COM	–	Clock for PHY IF.
rst_x	1	In	–	TOP	COM	–	Asynchronous reset for all FFs in COM block.

9.3.2. COM IF

Table 27: COM IF pin list

Name	Bit num	In Out	Initial value	From	To	Clock domain	Description
sys_cmd_req	1	In	–	SYSIF	COM	mclk	Memory access command request.
sys_cmd_addr	22	In	–	SYSIF	COM	mclk	Command address.
sys_cmd_len	4	In	–	SYSIF	COM	mclk	Data word length.
sys_cmd_write	1	In	–	SYSIF	COM	mclk	0: Command is read. 1: Command is write
sys_w_data	64	In	–	SYSIF	COM	mclk	Write data.
sys_w_mask	8	In	–	SYSIF	COM	mclk	Write data mask.
com_ready	1	Out	0	COM	SYSIF	mclk	COM block ready signal.
com_r_valid	1	Out	0	COM	SYSIF	mclk	Read data valid signal.
com_r_data	64	Out	0	COM	SYSIF	mclk	Read data.

9.3.3. PHY IF

Table 28: PHY IF pin list

Name	Bit num	In Out	Initial value	From	To	Clock domain	Description
phy_cs_x	1	Out	1	COM	PHY	mclkx2	DDR CS
phy_ras_x	1	Out	1	COM	PHY	mclkx2	DDR RAS
phy_cas_x	1	Out	1	COM	PHY	mclkx2	DDR CAS
phy_we_x	1	Out	1	COM	PHY	mclkx2	DDR WE
phy_odt	1	Out	0	COM	PHY	mclkx2	DDR on-die termination control
phy_cke	1	Out	0	COM	PHY	mclkx2	DDR CKE
phy_ba	3	Out	0	COM	PHY	mclkx2	DDR bank address
phy_addr	15	Out	0	COM	PHY	mclkx2	DDR address
phy_dqsoe	1	Out	0	COM	PHY	mclkx2	DQS and DQ output buffer enable
phy_wdq0	16	Out	0	COM	PHY	mclkx2	Write data 0
phy_wdq1	16	Out	0	COM	PHY	mclkx2	Write data 1
phy_wdm0	2	Out	0	COM	PHY	mclkx2	Write data mask 0
phy_wdm1	2	Out	0	COM	PHY	mclkx2	Write data mask 1
phy_rdcmd	1	Out	0	COM	PHY	mclkx2	Read command enable to PHY
phy_rdq0	16	In	–	PHY	COM	mclkx2	Read data 0

phy_rdq1	16	In	-	PHY	COM	mclkx2	Read data 1
phy_rodten	1	Out	0	COM	PHY	mclkx2	PHY read on-die termination enable

9.3.4. COM register IF

Table 29: COM register IF pin list

Name	Bit num	In Out	Initial value	From	To	Clock domain	Description
com_enable	1	In	-	REG	COM	mclk	When set to 1, COM block is enabled. COM mode setting and DDR timing parameter will be valid when this bit is toggled 0 to 1.
com_status	1	Out	0	COM	REG	mclk	0: COM block is not active. 1: COM block is active.
ddr_cabit	1	In	-	REG	COM	mclk	0:CA=9bit 1:CA=10bit
ddr_8bank	1	In	-	REG	COM	mclk	0:BA=2bit (4bank device) 1:BA=3bit (8bank device)
ddr_t_rcd	3	In	-	REG	COM	mclk	DDR timing parameter
ddr_t_rp	3	In	-	REG	COM	mclk	DDR timing parameter
ddr_t_rc	5	In	-	REG	COM	mclk	DDR timing parameter
ddr_t_ras_min	5	In	-	REG	COM	mclk	DDR timing parameter
ddr_t_rrd	3	In	-	REG	COM	mclk	DDR timing parameter
ddr_t_rtp	2	In	-	REG	COM	mclk	DDR timing parameter
ddr_t_mrd	3	In	-	REG	COM	mclk	DDR timing parameter
ddr_t_rfc	8	In	-	REG	COM	mclk	DDR timing parameter
ddr_t_refi	12	In	-	REG	COM	mclk	DDR timing parameter
ddr_t_wtr	2	In	-	REG	COM	mclk	DDR timing parameter
ddr_t_rtw_add	2	In	-	REG	COM	mclk	DDR timing parameter
ddr_auto_init_en	1	In	-	REG	COM	mclk	0: Disable auto DDR initialization at pos-edge of com_enable. 1: Enable auto DDR initialization at pos-edge of com_enable.
ddr_aref_en	1	In	-	REG	COM	mclk	0: Disable auto refresh. 1: Enable auto-refresh.
ddr_aref_num	4	In	-	REG	COM	mclk	Number of AREF commands consecutively performed after one AREF request. AREF request is issued every period of (ddr_aref_num * ddr_t_refi / 2) cycle @mclk.
ddr_set_trefi	1	In	-	REG	COM	mclk	After changing ddr_t_refi, toggle this bit 0 to 1 (if tREFI needs to be changed), then new ddr_t_refi value is set to AREF controller.
drcmd_req	1	In	-	REG	COM	mclk	When set to 1, direct command request is issued. Automatically cleared after completion of the command.
drcmd_cmd	4	In	-	REG	COM	mclk	Direct command type 0x0: NOP 0x1: Precharge all 0x2: Auto refresh 0x3: MRS 0x4: EMRS1 0x5: EMRS2 0x6: EMRS3 0x7: Device deselect 0x8: Self refresh entry 0x9: Power down entry

							0xA to 0xF: Reserved
ddr_mr	16	In	–	REG	COM	mclk	DDR MR setting Used for auto-initialization and direct command
ddr_emr1	16	In	–	REG	COM	mclk	DDR EMR1 setting Used for auto-initialization and direct command
ddr_emr2	16	In	–	REG	COM	mclk	DDR EMR2 setting Used for auto-initialization and direct command
ddr_emr3	16	In	–	REG	COM	mclk	DDR EMR3 setting Used for auto-initialization and direct command
phyif_wen_lat	3	In	–	REG	COM	mclk	PHY IF setting
phyif_wodt_lat	3	In	–	REG	COM	mclk	PHY IF setting
phyif_wodt_ext	3	In	–	REG	COM	mclk	PHY IF setting
phyif_wdata_lat	3	In	–	REG	COM	mclk	PHY IF setting
phyif_ren_lat	3	In	–	REG	COM	mclk	PHY IF setting
phyif_rodt_lat	3	In	–	REG	COM	mclk	PHY IF setting
phyif_rodt_ext	3	In	–	REG	COM	mclk	PHY IF setting
phyif_wodt_en	1	In	–	REG	COM	mclk	Write ODT enable
phyif_rdlat	3	In	–	REG	COM	mclk	PHY IF setting

9.3.5. Other register IF

Table 30: PHY register IF input

Name	Bit num	In Out	Initial value	From	To	Clock domain	Description
phy_gds	2	In	–	REG	COM	mclk	Read pointer counter control 1: read data is valid RL+4 after read command 2: read data is valid RL+5 after read command 0, 3: invalid setting

9.4. Register map

No register map in COM block.

9.5. IF specification

9.5.1. COM IF

Refer to section 6.2.

9.5.2. PHY IF

Refer to section 6.3.

9.5.3. COM/PHY register IF

Refer to section 6.6, 6.7 and 6.8.

9.6. Function specification

9.6.1. DDR initialize

DDR initialize を行う方法として、auto と manual の 2 通りの方法がある。

com_enable の立上り時に ddr_auto_init_en=1 であれば auto initialize sequence が発行される。auto initialize sequence は図の通り発行され、MRS, EMRS1/2/3 コマンドの設定値には ddr_mr、ddr_emr1/2/3 の設定値が使用される。

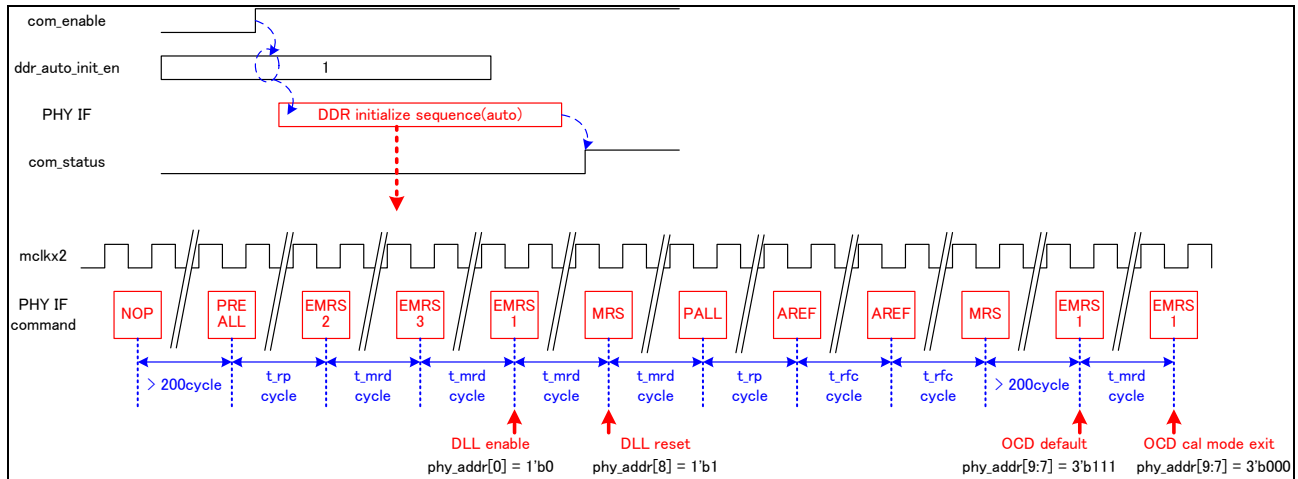


Fig. 49: DDR initialize (auto)

ただし DLL および OCD の制御を行うコマンドでは、phy_addr ビットの一部分は設定値によらず下記の通り制御される。

- DLL enable phy_addr[0] = 1'b0
- DLL reset phy_addr[8] = 1'b1
- OCD default phy_addr[9:7] = 3'b111
- OCD cal mode exit phy_addr[9:7] = 3'b000

また、auto initialize sequence の終了後、com_status=1 として COM block が ready となったことを示す。

com_enable の立上り時に ddr_auto_init_en=0 であれば auto initialize sequence は発行されない。この場合、com_enable=0 の期間に direct command を使って manual で DDR initialize を行い、その完了後に com_enable=1 とする。

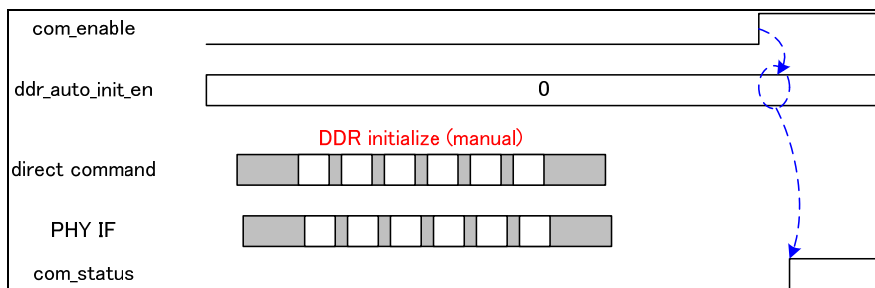


Fig. 50: DDR initialize (manual)

ddr_auto_init_en=0 のときは com_enable の立上りの後すぐに com_status=1 として、COM block は ready 状態となる。

9.6.2. Addressing

COM IF の memory access request から DDR command を生成するときのアドレス変換イメージを示す。Activate と Write/Read (auto precharge 付きを含む) command を出力するときの Row address, Column address, Bank address には、ddr_cabit と ddr_8bank 設定により下記のとおり sys_cmd_addr の各ビットが設定される。

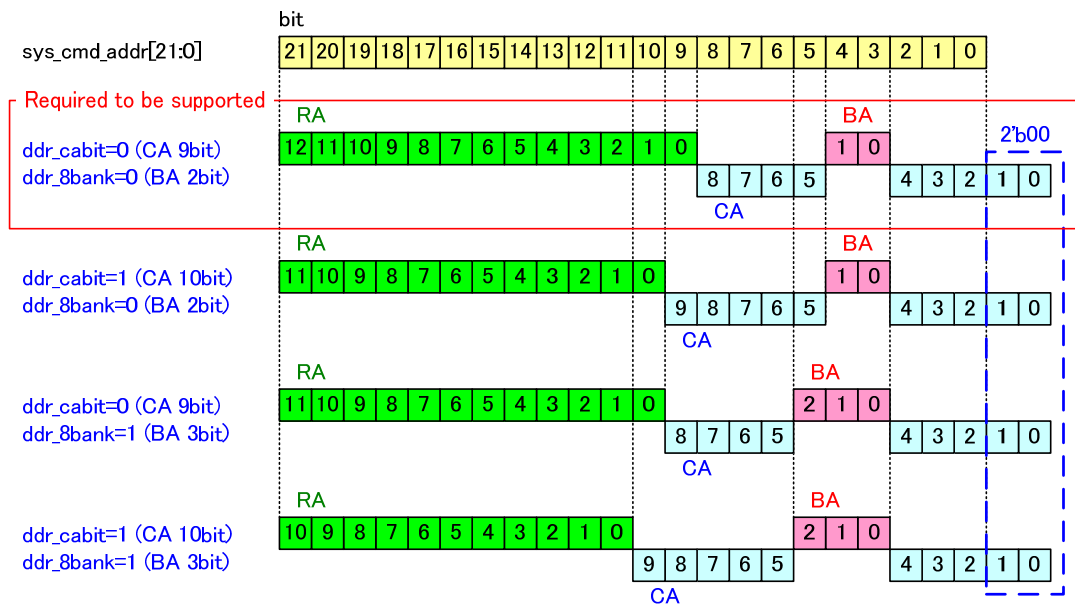


Fig. 51: addressing image

Column address の bit[1:0]は常に0とする。またそのコマンドで使われない address bit がある場合も、その未使用ビットは0 固定とする (CA=10bit のデバイスを CA=9bit の設定で使うとき、8bank デバイスを 4bank 設定で使うときなど、不正な領域にアクセスするのを防ぐため)。

Bank address の最下位ビットは sys_cmd_addr[3]の位置にあたるため、sys_cmd_len が 1 から 8 のコマンドは、sys_cmd_addr[2:0]=3'd0 であれば同一バンクへのアクセスとなる。

9.6.3. DDR command generation

COM block は、COM IF からの memory access request を受け取り、それに応じた DDR command sequence を生成して PHY IF へ出力する。

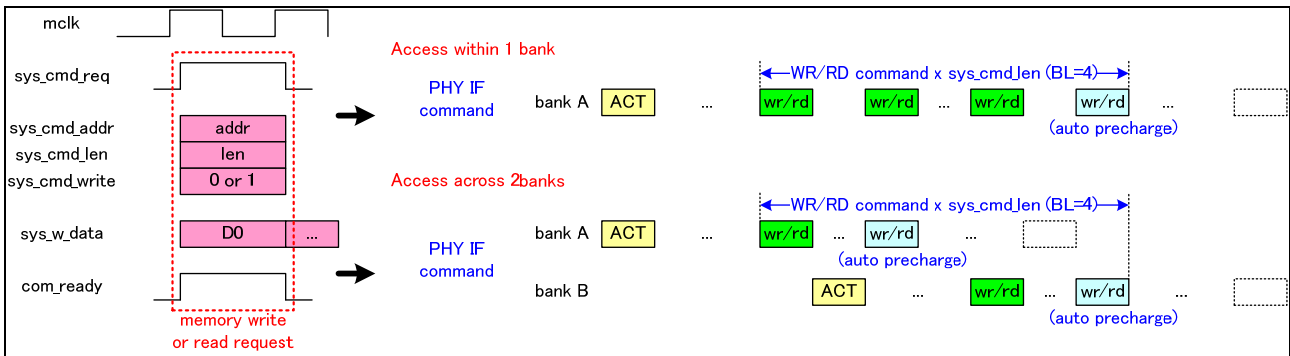


Fig. 52: DDR command sequence image

COM IF からの request を 1 つ受け取るごとに、DDR command sequence は Activate コマンドから始まり、その後 sys_cmd_len の個数の Write または Read コマンドを発行して(最後は Auto precharge 付きの Write または Read コマンド)、そのバンクを precharge した状態で終わる。その request による memory access が 2bank にまたがる場合は、それぞれの bank について同様のシーケンスを生成し、bank interleave させて発行する。この 2 つのシーケンスの Write または Read コマンドの合計が sys_cmd_len となる。

sys_cmd_len=8 のコマンド (read または write どちらか片方のみ) を連続して入力したとき、sys_cmd_addr[2:0]=0 かつ sys_cmd_addr[21:3] が incremental な場合(または異なるバンクへのアクセスが続く場合)は、それぞれの DDR command sequence は bank interleave されて効率よく並べられる。(tRCD - AL の値が奇数の場合は Write command または Read command の間隔は BL/2 に揃えられる)

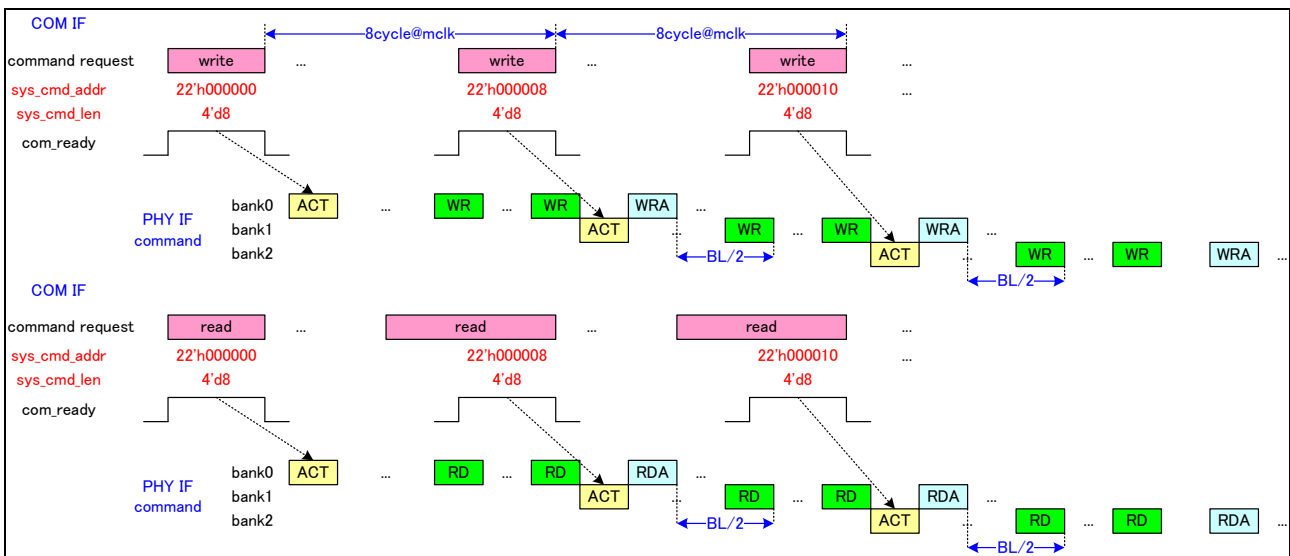


Fig. 53: bank interleave

9.6.4. Auto refresh

(ddr_aref_en & com_enable)=1 のとき、AREF コマンドの自動発行機能が enable となる。この機能が enable のときは、(ddr_aref_num x ddr_t_refi / 2) サイクル @ mclk ごとに AREF リクエストが発行される。AREF リクエストが発行されると、実行中の command sequence があればその完了を待ってから ddr_aref_num の回数ほど AREF コマンドが発行される。

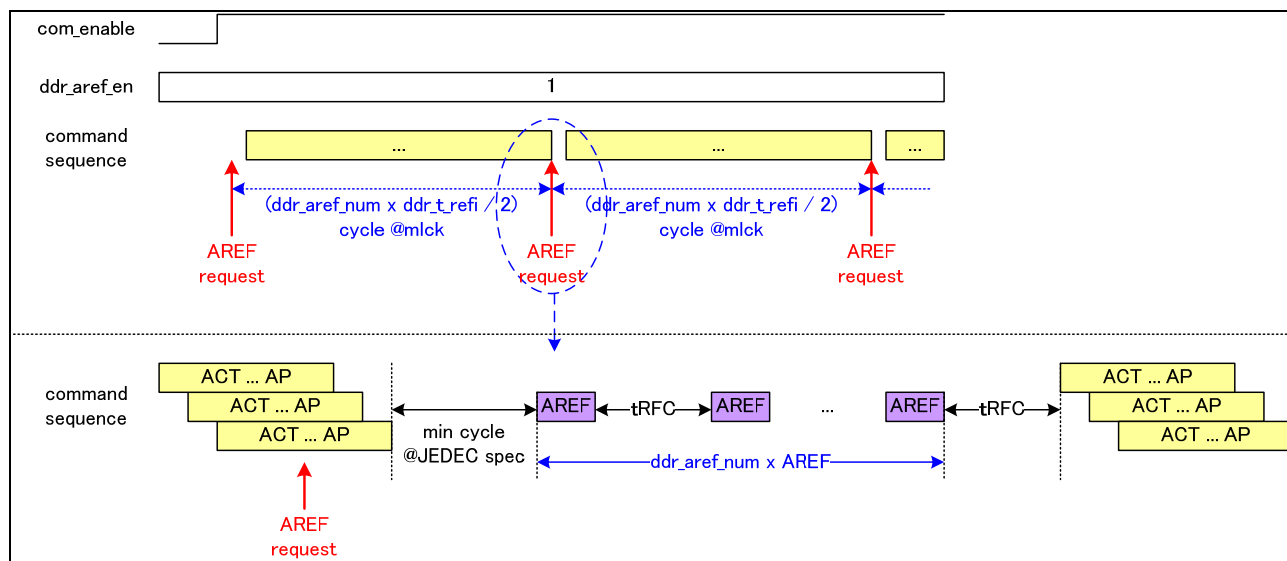


Fig. 54: auto refresh

AREF request 発行後、`ddr_oref_num` の回数ほど AREF コマンドが出力されるまで新しくコマンドシーケンスは開始できない。

9.6.5. tREFI setting

tREFI 設定は他の DDR timing パラメータ設定と違い、COM block 動作中に設定変更が行われる可能性がある。`ddr_t_refi` の設定は bit[7:0] と bit[11:8] で分かれて設定されるため、設定途中の値が COM 内部で使用されないよう注意する必要がある。このため、`ddr_t_refi` 設定は下に示すとおり `com_enable` の立上りまたは `ddr_set_trefi` 信号の立上り時でラッチされ、その後 COM 内部で使用される。

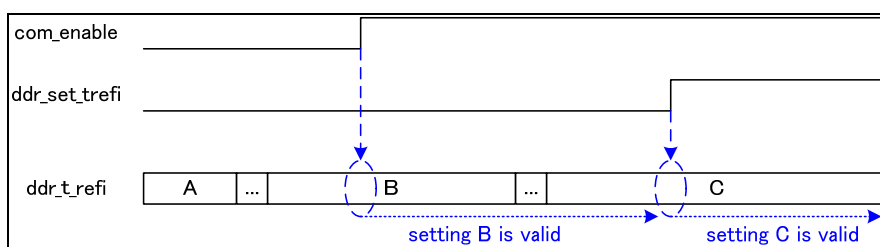


Fig. 55: tREFI setting

9.6.6. com_enable and com_status

`com_enable=0` かつ `com_status=0` のとき COM block は動作せず、この状態のとき register IF から memory へ direct command を発行できる。`com_enable` を 0 から 1 へトグルさせると auto initialize sequence の発行が行われ (enable であれば)、COM IF からのリクエストを受け付け可能となったところで `com_status=1` となる。この `com_status` 信号が、COM block が active な状態であることを示す。`com_enable` を 1 から 0 へ変えると、COM IF は disable 状態となりそれ以降のリクエストは受け付けない。ただしそのとき実行中のシーケンスがあればそれが完了するまで `com_status=1` の状態を保ち、シーケンス完了後に `com_status=0` として COM block が active 状態でなくなったことを示す。

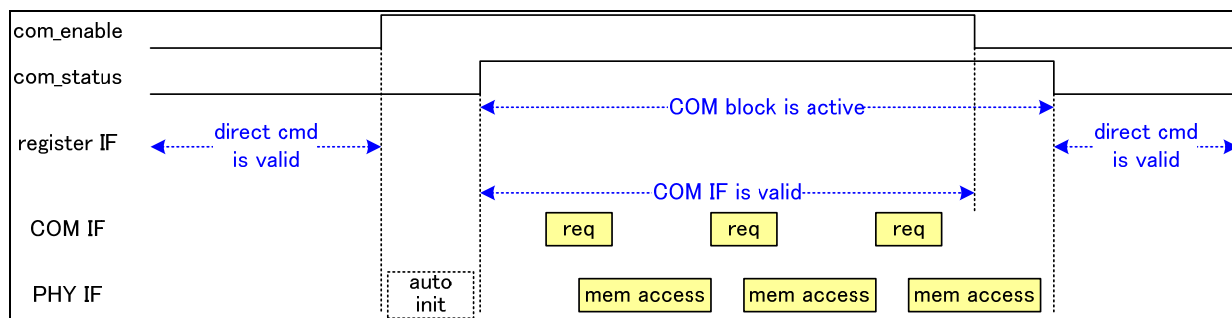


Fig. 56: com_enable and com_status

9.6.7. Direct command

Direct command はユーザが register IF を通して DRAM へコマンドを発行するための機能である。発行するコマンドの種類は `drcmd_cmd[3:0]` で指定され、以下のコマンドをサポートする。

- `drcmd_cmd[3:0]=4'd0` NOP
- `drcmd_cmd[3:0]=4'd1` Precharge All
- `drcmd_cmd[3:0]=4'd2` Auto refresh
- `drcmd_cmd[3:0]=4'd3` MRS
- `drcmd_cmd[3:0]=4'd4` EMRS1
- `drcmd_cmd[3:0]=4'd5` EMRS2
- `drcmd_cmd[3:0]=4'd6` EMRS3
- `drcmd_cmd[3:0]=4'd7` Device deselect
- `drcmd_cmd[3:0]=4'd8` Self refresh entry
- `drcmd_cmd[3:0]=4'd9` Power down entry

`drcmd_req` の立上りで `drcmd_cmd` の値をラッチして、その設定に従い PHY IF へコマンドを発行する。(`drcmd_req_clear` による `drcmd_req` 信号の clear function は削除。)

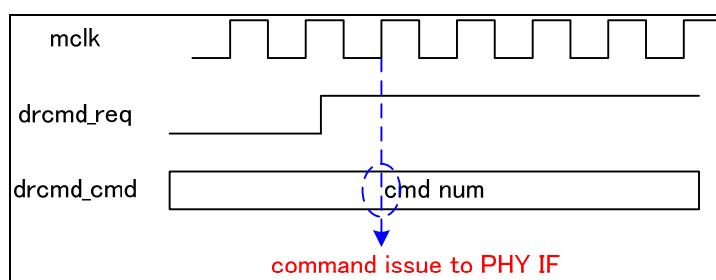


Fig. 57: direct command sequence

direct command の種類が Precharge All/Auto refresh/MRS/EMRS1/EMRS2/EMRS3 の場合、`drcmd_req` の立上り後、そのコマンドは `mclkx2` クロックで 1cycle の期間 PHY IF へ出力され、その後 NOP の状態へ戻る。direct command の種類が Device deselect/Self refresh entry/Power down entry の場合、`drcmd_req` の立上り後、PHY IF のコマンド出力は Device deselect/Self refresh/Power down の状態を保持する。direct command の NOP は、Device deselect/Self refresh/Power down またはリセット後の初期状態(CKE=low の状態)から NOP 状態へ戻るときに使用される。

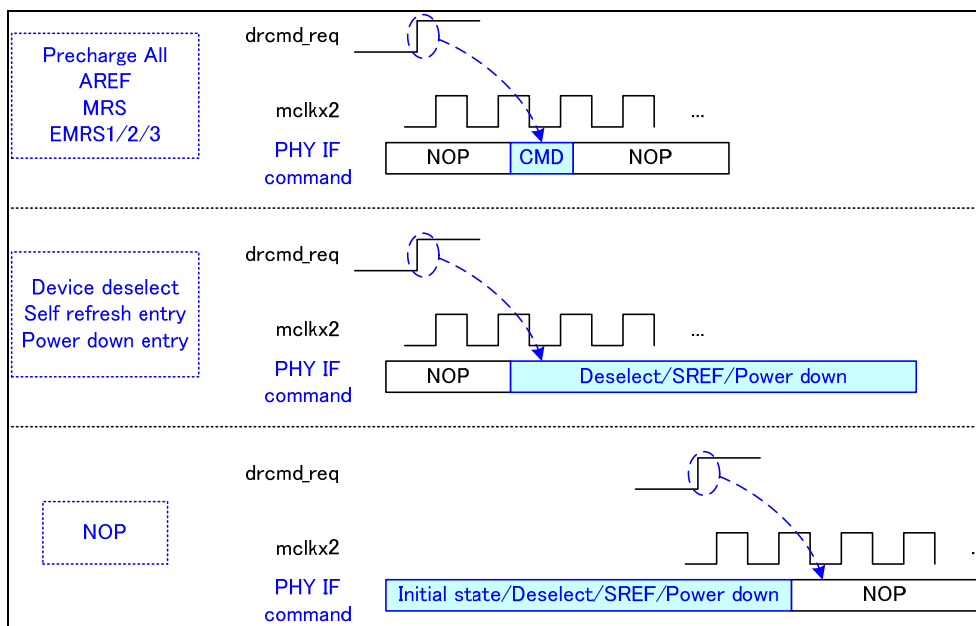


Fig. 58: direct command

direct command により MRS/EMRS1/EMRS2/EMRS3 が出力される時、`phy_addr` にはそのときに設定されている `ddr_mr/ddr_emr1/ddr_emr2/ddr_emr3` の `bit[14:0]` の値がそれぞれそのまま出力される。

9.6.8. ddr_t_rtw_add function

JEDEC specification では、read command 発行後に write command を発行できるまでの minimum time は $t_{RTW} = ((BL/2) + 2)$ cycle である。COM block の仕様として、 t_{RTW} は `ddr_t_rtw_add` の設定により次とおりとする。

$$\text{t}_{RTW} \text{ (COM block spec)} = (BL/2) + 2 + \text{ddr_t_rtw_add} \text{ [cycle @mclkx2]}$$

9.6.9. PHY IF signal timing setting

write data/mask 信号(`phy_wdq*/wdm*`)は write コマンド出力の `phyif_wdata_lat` サイクル後から $(BL/2)$ サイクルの期間で出力される。`phy_dqsoe` 信号は、write コマンド出力の `phyif_wen_lat` サイクル後から $(BL/2)$ サイクルの期間 High となり、それ以外は Low となる。`phyif_wodt_en=1` のとき、`phy_odt` 信号は write コマンド出力の `phyif_wodt_lat` サイクル後から $((BL/2)+\text{phyif_wodt_ext})$ サイクルの期間 High となり、それ以外は Low となる。`phyif_wodt_en=0` のとき、`phy_odt` は Low 固定となる。

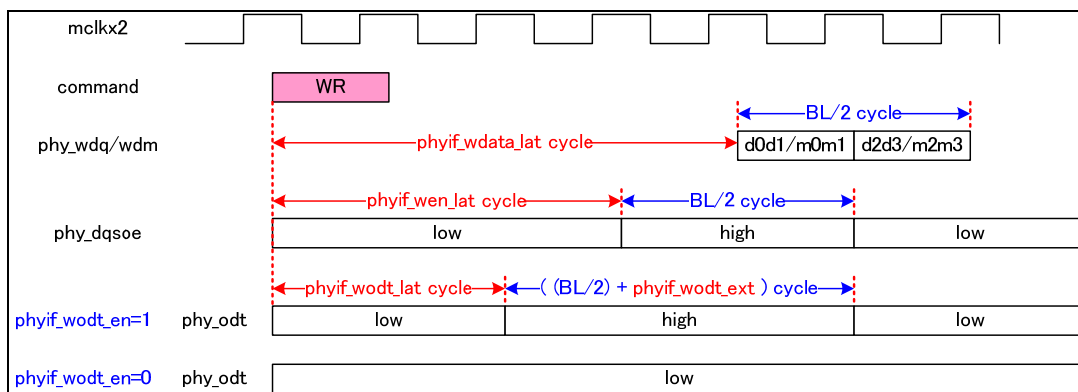


Fig. 59: PHY IF signal timing (write)

phy_rdcmd 信号は read コマンド出力の phyif_ren_lat サイクル後から(BL/2)サイクルの期間 High となり、それ以外は Low となる。phy_rdotden 信号は read コマンド出力の phyif_rodt_lat サイクル後から ((BL/2)+phyif_rodt_ext)サイクルの期間 High となり、それ以外は Low となる。

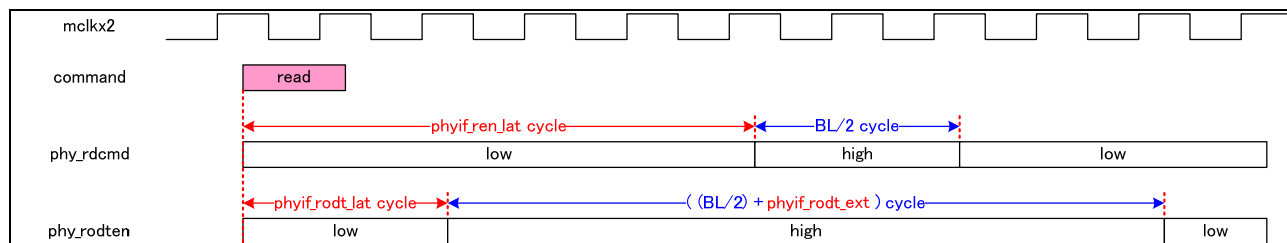


Fig. 60: PHY IF signal timing (read)

Read data(phy_rdq0, phy_rdq1)を COM block がラッチするタイミングは、phyif_ren_lat, phyif_rdlat と phy_gds により制御される。phy_rdq0/1 はこれらの設定により下の図のとおりラッチされる。

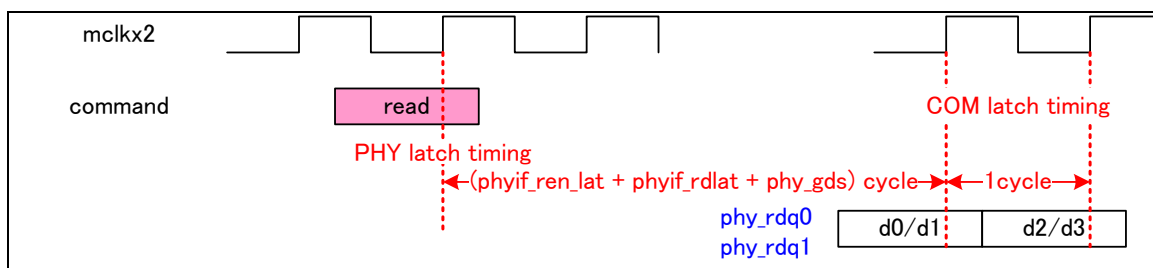


Fig. 61: read data latch timing

9.6.10. t_FAW restriction

COM block は各種レジスタ設定に関わらず、全てのモードで下記の t_FAW 制約を守るものとする (ddr_8bank=0 のときでも 8bank device が使われることがある)。

- t_FAW = 18 [cycle @mclkx2]

10. Appendix

10.1. Clock domain change

非同期な関係にあるクロック間でデータを受け渡しするときの方法を示す。

- Single bit signal (stable)
- Register setting

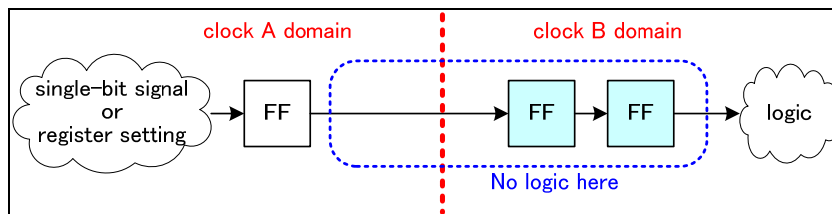


Fig. 62: single bit or register setting

- Single bit (pulse)

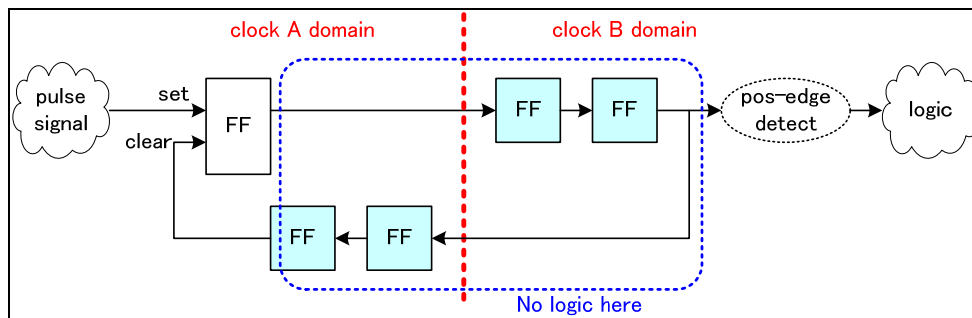


Fig. 63: pulse signal

- Multi-bit signal

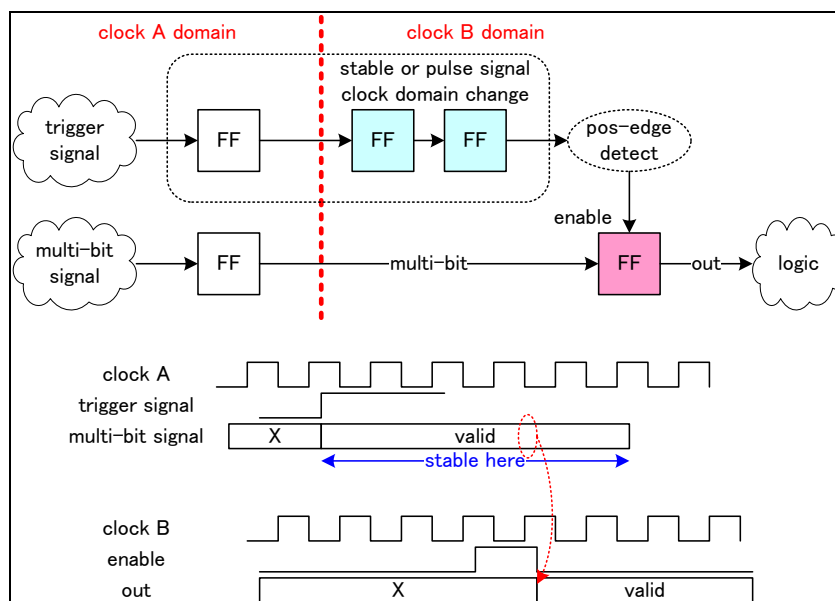


Fig. 64: multi-bit

- wdata, rdata (using SRAM)

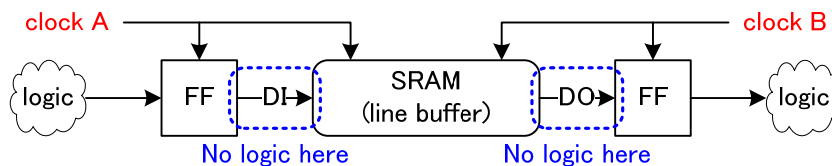


Fig. 65: wdata and rdata

SRAM の DI/DO ポートは FF と直接つなぎ、間に logic を挟まない。

10.2. Sysmte IF port restriction

L_MIF から System IF へ出力される rdata 信号は、FF から直接出力してロジックを挟まない。

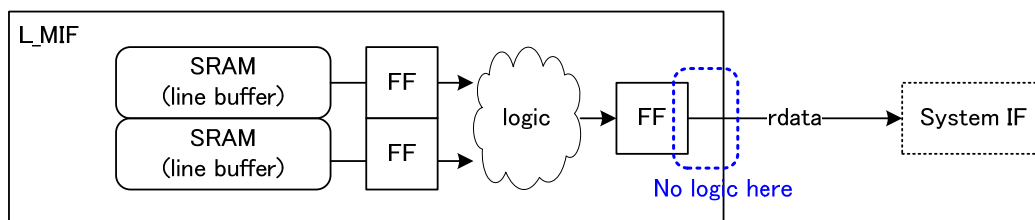


Fig. 66: System IF port restriction

10.3. PHY IF port restriction

COM block から PHY IF へ出力するポートは、FF から信号を直接出力してロジックを挟まない。また PHY IF からの入力も FF で直接信号を受け取り、間にロジックを挟まない。

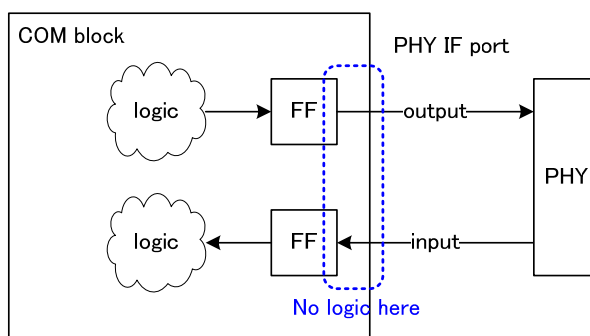


Fig. 67: PHY IF port restriction

10.4. COM IF address and data mapping image

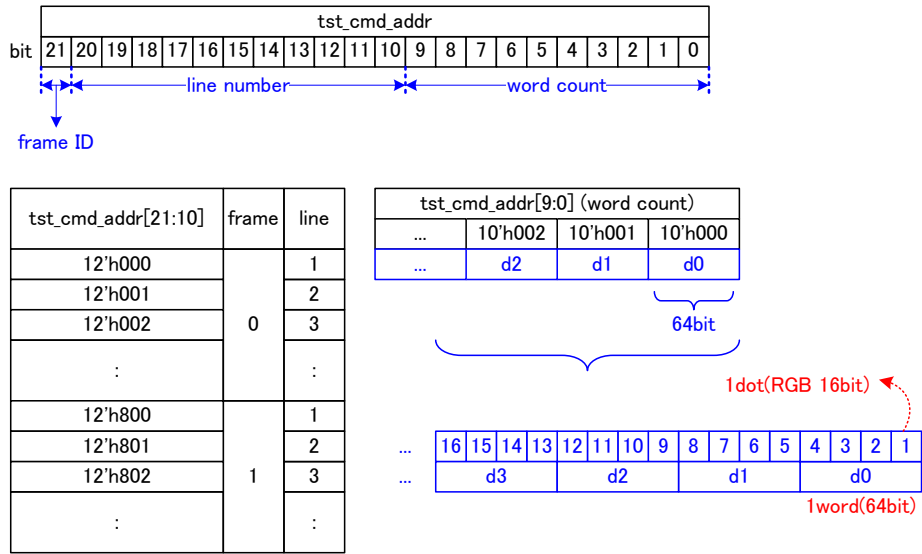


Fig. 68: COM IF address and data image

10.5. MIF, PHY, DRAM configuration

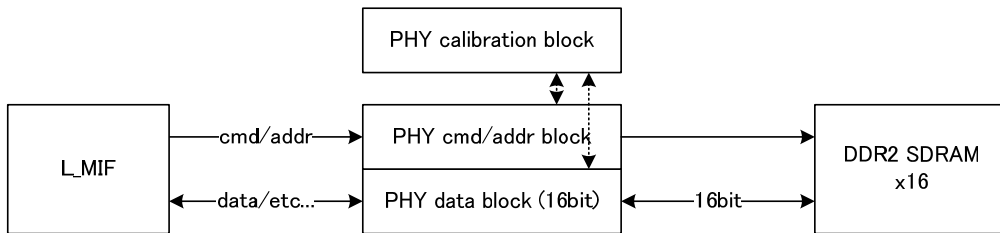


Fig. 69: MIF/PHY/DRAM configuration

10.6. COM block difference between F_MIF and L_MIF

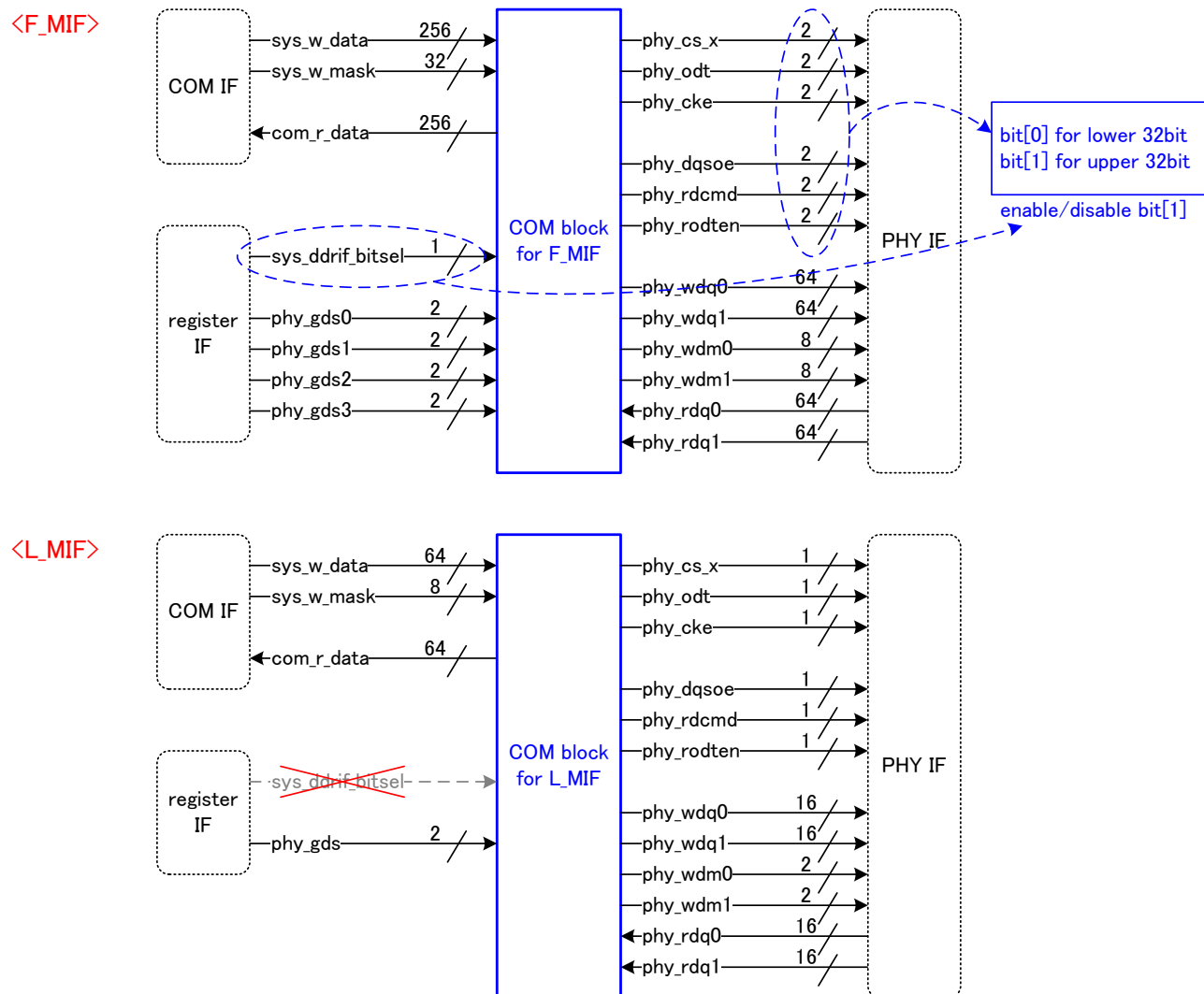


Fig. 70: COM block difference

10.7. SRAM wrapper

line buffer 用の SRAM は下記のものを使用する。

Name:	SZAA90_528X64X1CM4
Memory type:	FSD0A_A_SZ
Words:	528
Bits:	64
Column:	4
Output Loading:	0.5pf
Power ring width:	10um
Power ring type:	ppr

RTL 中に line buffer を置くときは、SLSI から提供する SRAM wrapper module(RAM2P528W64B_WRAP.v) を使用する。

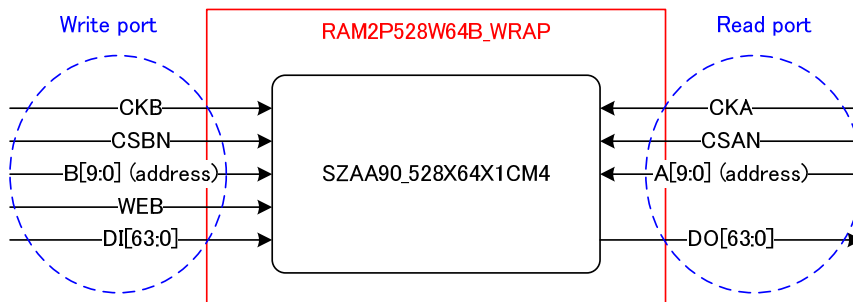


Fig. 71: SRAM wrapper

10.8. PHY wrapper

L_MIF と DDR PHY の接続には、SLSI から提供する wrapper module (L_MIF_TOP.v)を使用する。

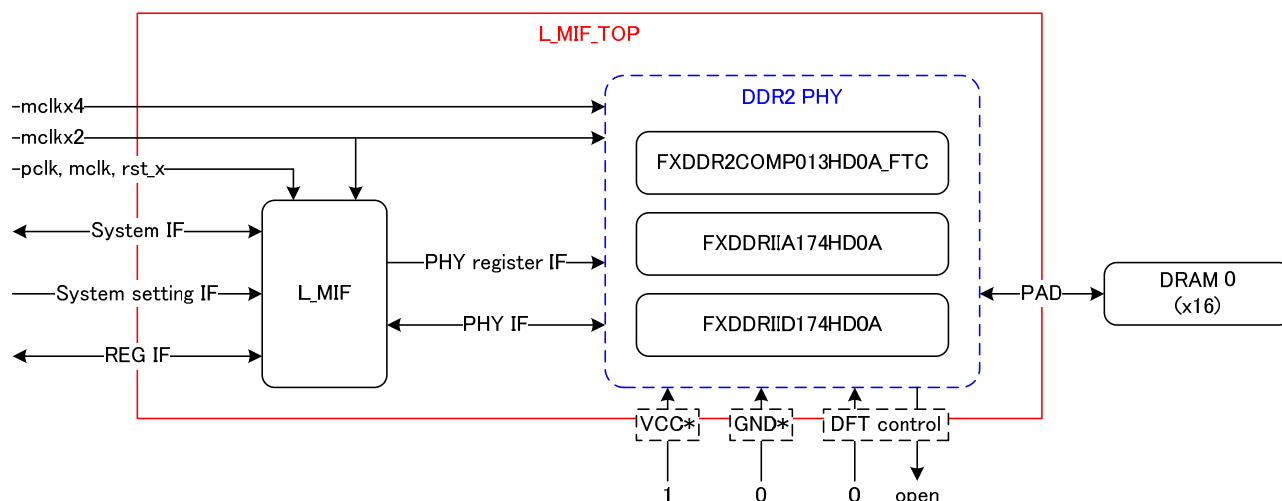


Fig. 72: PHY wrapper

Power pin(信号名が "VCC" で始まるもの)は全て 1 固定、ground pin(信号名が "GND" で始まるもの)は全て 0 固定で使用する。DFT control signal(信号名が "dft_" で始まるもの)の入力信号は全て 0 固定、出力信号は全て open で使用する。また PAD signal(信号名が "pad_" で始まるもの)は下の表のとおり DRAM 0 と接続する。

Table 31: PAD signal connection

PAD name	bit	direction	connected to
pad_rdrvdn	1	Out	open
pad_rdrvup	1	Out	open
pad_cas	1	Out	DRAM 0
pad_ras	1	Out	DRAM 0
pad_we	1	Out	DRAM 0
pad_addr	15	Out	DRAM 0
pad_ba	3	Out	DRAM 0
pad_ck	3	Out	[0]: DRAM 0 [2:1]: open
pad_ckb	3	Out	[0]: DRAM 0 [2:1]: open

pad_cke	2	Out	[0]: DRAM 0 [1]: open
pad_cs	2	Out	[0]: DRAM 0 [1]: open
pad_odt	2	Out	[0]: DRAM 0 [1]: open
pad_dq	16	Inout	DRAM 0
pad_ldqs	1	Inout	DRAM 0
pad_hdqs	1	Inout	DRAM 0
pad_ldqsb	1	Inout	DRAM 0
pad_hdqsb	1	Inout	DRAM 0
pad_ldm	1	Out	DRAM 0
pad_hdm	1	Out	DRAM 0
pad_vref	2	In	open?
pad_dummy	1	Inout	open

DRAM の Address/Bank address ビット数に対して pad_addr/pad_ba のビット数が余る場合は、上位側の余ったビットを open とする。

11. Revision history

0.00	2009/04/22	initial version
0.01	2009/04/22	- add 10.5, 1.1
0.02	2009/04/22	Add parameter phyif_rdlat to COM register update Table 16, Table 17, Table 20, Table 25, Table 29 update section 9.6.9, Fig. 61
0.03	2009/04/27	- update section 5, regmap base address - update section 6.1, Fig. 9 (rdvalid to rdata timing) - update section 10.1 (wdata, rdata (using SRAM)) - add section 10.2 Sysmte IF port restriction
0.04	2009/04/30	- add section 10.7 - add section 10.8
0.05	2009/05/14	- update clock frequency specification update Table 6 - add REGIF minimum access interval spec update section 6.4 - add tst_mode_force signal to force(enable) test mode update Table 15 update Table 19 add section 8.4 - delete tst_cmd_req_clear signal and add tst_cmd_status signal update section 5.1, 6.6, 8.1, 8.2 - delete drcmd_req_clear signal update section 5.2, 6.7, 9.3.4, 9.6.7 - update 2.2.1 Frame format
0.06	2009/05/14	- add detailed timing spec of mif_rst/mif_enable update section 6.1 System IF - add section 7.5 frame format change
0.07	2009/05/21	- typo correction - add section 2.2.3 Additional frame format
0.08	2009/05/25	- update blanking spec (total blank min spec) update section 2.2.1 update Fig. 10 - add frame format BLKMIN and BLKMAX to section 2.2.3 Additional frame format - update initial value and bit width which were TBD update Table 10, Table 13, Table 17 - delete fclk (change System setting IF clock to pclk) update section 3.2 Clock specification update Table 8, Table 10
0.09	2009/05/29	- update Table 4, disp size of WXGA and BLKMAX - update 10.8 PHY wrapper
0.10	2009/06/16	- add register map for test block and related IO port update Fig. 2, Fig. 4 add section 4.7 UFT register IF add section 5.4 UFT register add section 6.9 UFT register IF - correct mistake of <h_fp> and <h_sync+h_bp> period update Fig. 10 - update COM register initial value (phyif_wodt_lat, phyif_rodt_lat) update Table 16: COM register map