M3 Low-Power Flash Layer (Version 3 Large) Documentation (FLPv3L)

Revision 1.0

Yejoong Kim*[1], Qing Dong†[1], Inhee Lee‡[1], and Xun Sun§[1]

[1]Michigan Integrated Circuits Laboratory, University of Michigan, Ann Arbor

March 15, 2019

*yejoong@umich.edu
†qingdong@umich.edu
‡inhee@umich.edu
§xusun@umich.edu

Contents

1 Revision History

- FLPv1L/FLPv1S
  - First versions taped-out on September 16th 2015, by Yejoong Kim
  - Die size: 2500um $\times$ 3000um (FLPv1L), 1050um $\times$ 2230um (FLPv1S)
- FLPv2L/FLPv2S/FLPv2LL/FLPv2SL
  - Taped-out on May 18th 2016, by Yejoong Kim
  - FLPv2S, FLPv2L are built with Regular Vth (nch/pch) digital logic
  - FLPv2SL, FLPv2LL are built with Ultra-Low Leakage Vth (nch_ull/pch_ull) digital logic
  - Removed decap filler cells due to the high leakage
  - Updated the Clock Gen to fix the high current when CLKcomp is not running.
  - Auto-tuning for Self-Boot-Up: adjust the clamper strength in case of header mismatch
  - Self-Boot-Up flag: send out an MBus message containing information in case of Self-Boot-Up failure
  - Clock-gating by Design Compiler
  - Clock pad for external streaming is now built with Schmitt Trigger
  - Flash macro has various bug fixes and improvements
  - Die size: 2500um $\times$ 3000um (FLPv2L, FLPv2LL), 1050um $\times$ 2230um (FLPv2S, FLPv2SL)
- FLPv3L/FLPv3S
  - Taped-out on September 11, 2017, by Yejoong Kim
  - Designed and organized in the new m3_hdk directory
  - Removed VREF_EXT pad
  - Removed unnecessary bits in MBus Register File
  - Introduced LC-type Register File (non-retentive) for power and area saving
  - Default values of some MBus Register File have been changed.
  - Die size: 2500um $\times$ 3000um (FLPv3L), 1050m $\times$ 2230um (FLPv3S)

2 Layer Description

The Low-Power Flash Layer (Version 3 Large) (FLPv3L) contains an 8Mb custom Flash newly designed by Qing Dong. It also includes a write buffer (32kB SRAM).

- MBus Full Prefix is 0x12303.
- Designed in TSMC 90nm (CMN90G rf3p7m5x1n0u2ff).
- Taped-out on September 11, 2017.
- Top-Level layout is located at:
  `m3_hdk/virtuoso/TSMC90/FLPv3L_TOP/FLPv3L/layout`
- Top-Level LVS netlist is located at:
  `m3_hdk/layer/FLP/FLPv3L/cdl/FLPv3L.cdl`
- Top-Level Spice netlist is located at:
  `m3_hdk/layer/FLP/FLPv3L/ckt/FLPv3L.ckt`
- C header file is located at:
  `m3_hdk/layer/FLP/FLPv3L/verilog/genRF/FLPv3L_RF.h`

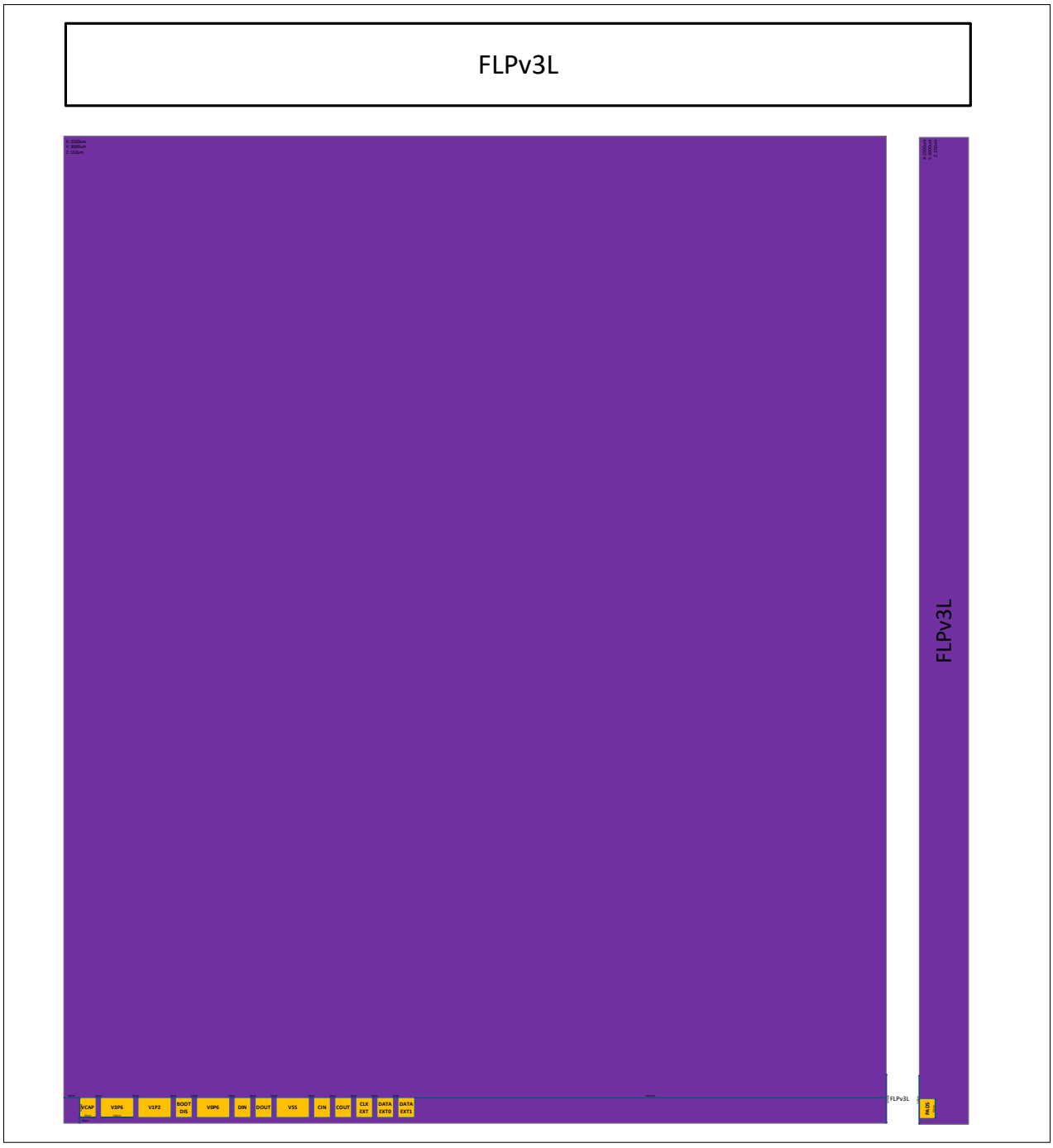

Figure 1: Low-Power Flash Layer (Version 3 Large) (FLPv3L) (2500um $\times$ 3000um)

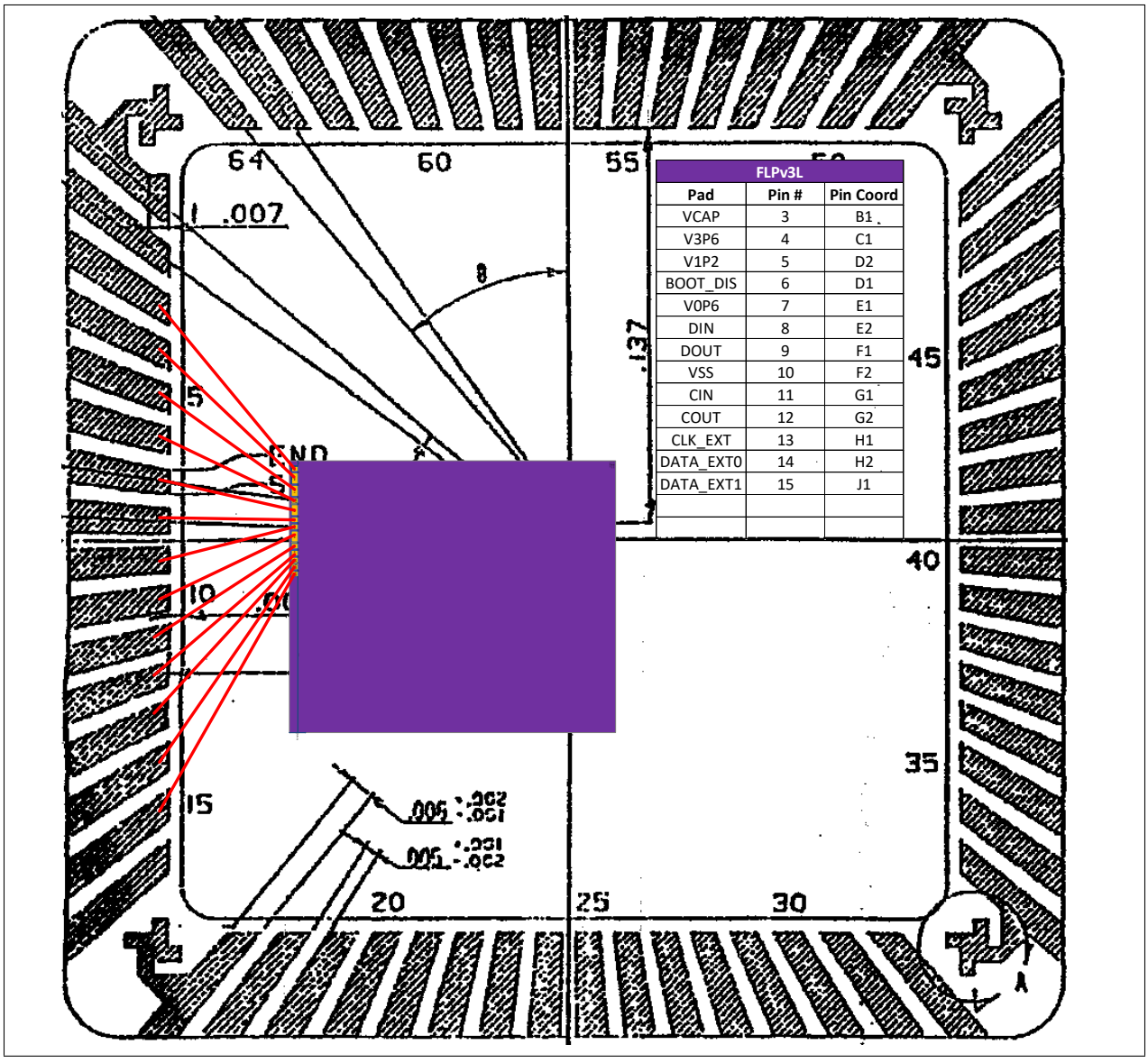

Figure 2: FLPv3L Wirebonding Diagram

3 Block Diagram

Figure 3 shows a simple block diagram of FLPv3L. For simplicity, MBus blocks/connections and Flash tuning bits are not shown.

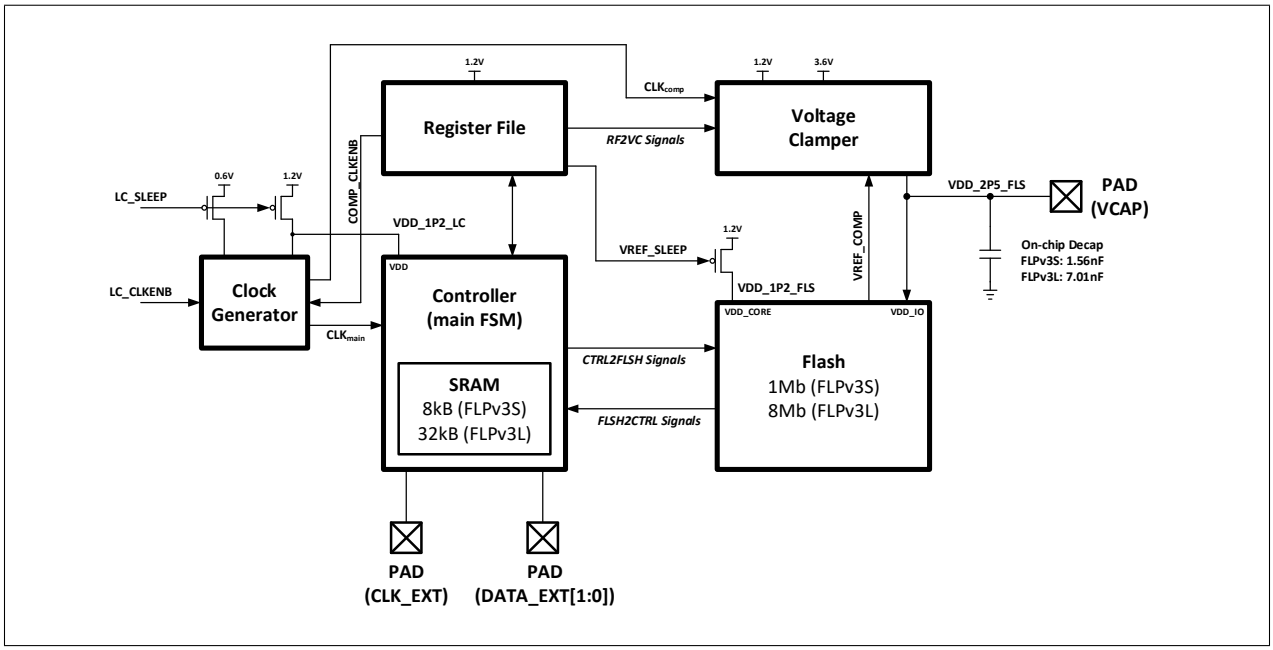

Figure 3: FLPv3L Block Diagram

RF2VC Signals include the following:

- COMP_SLEEP
- COMP_ISOL
- DISABLE_BYPASS_MIRROR
- COMP_CTRL_VOUT
- COMP_CTRL_I_1STG
- COMP_CTRL_I_2STG_BAR

CTRL2FLSH Signals include the following:

- SE (Read Clock), XE (Row Address Enable), YE (Column Address Enable)
- XADR (Row Address), YADR (Column Address), DIN (Data Input)
- PROG (Program Enable), NVSTR (High Voltage Enable), ERASE (Erase Enable)
- IREF_ERASE_EN (Reference Cell Erase Enable)
- ABUF_EN (Vref Analog Buffer Enable), RESETB (Flash Reset)
- HVCP_EN (HV-Pump Enable), MVCP_EN (MV-Pump Enable)
- SC_EN (Switch-Cap Enable), BEN (Bank Select)

FLSH2CTRL Signals include the following:

- DOUT (Data Output)

4 Memory Space

Figure 4 shows the memory space in FLPv3L. SRAM consists of two sub-SRAMs, so actually it is an SRAM Bank. However, user shall treat it as a big one SRAM. Thus, whenever this document says 'SRAM', it actually means this SRAM Bank. The reason of having the two sub-SRAMs is to enable ping-pong streaming.

There are three kinds of memory address type as shown in the figures:

- **MBus Memory ADDR**: This is a 32-bit memory address and should be used in all MBus transactions. It is word-aligned, so its lowest 2-bits are always 0.
- **SRAM ADDR**: This is a 13-bit SRAM address. Each LSB indicates a word (32-bit). This address should be used to specify SRAM addresses in MBus Register File, such as SRAM_START_ADDR.
- **Flash ADDR**: This is an 18-bit Flash address. Each LSB indicates a word (32-bit). This address should be used to specify Flash addresses in MBus Register File, such as FLSH_START_ADDR.

Note that, internally, the Flash has 8-bit I/O. The controller (main FSM) automatically handles the 8-bit ↔ 32-bit conversions, so user only sees the 32-bit I/O.

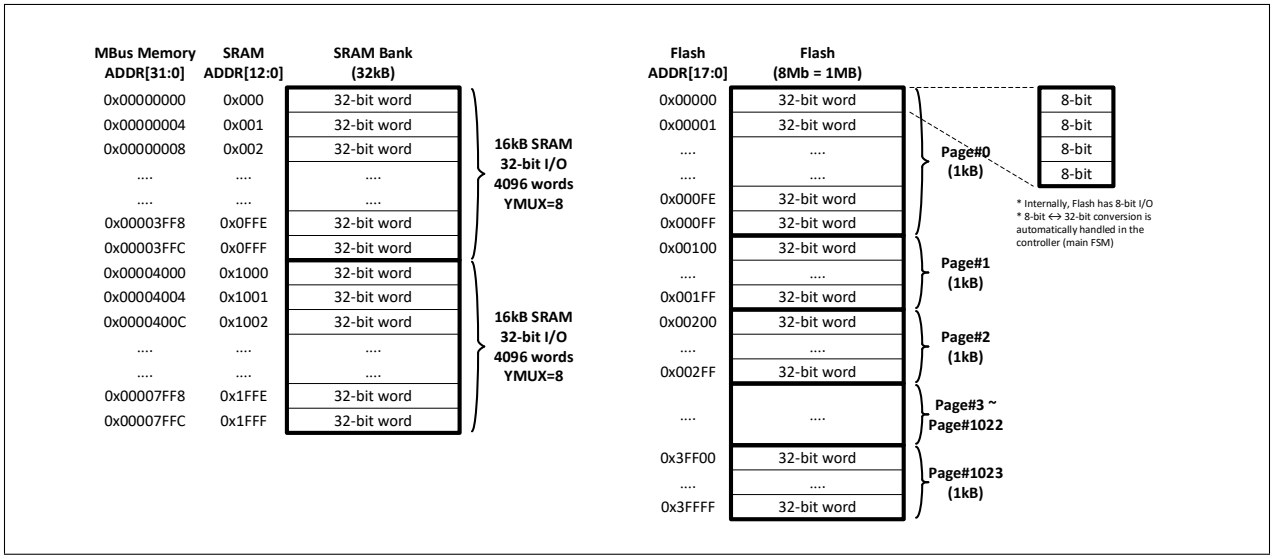

Figure 4: FLPv3L Memory Space

5 MBus Register File

5.1 MBus Register File Mapping

Table 1 shows MBus Register File mapping information. 'NR' indicates a non-retentive register.

Reg Addr	Bit Field	Reg Name	Property	Size & Reset	Remark
Register 0 (0x00) Default: 24'hF84209					
0x00	[23:19]	Tcyc_read	W/R	5'h1F	
	[18:13]	T3us	W/R	6'h02	
	[12:7]	T5us	W/R	6'h04	
	[6:0]	T10us	W/R	7'h09	
Register 1 (0x01) Default: 24'h007F09					
0x01	[23:8]	Tcyc_prog	W/R	16'h007F	
	[7:0]	Tprog	W/R	8'h09	
Register 2 (0x02) Default: 24'h000100					
0x02	[15:0]	Terase	W/R	16'h0100	
Register 3 (0x03) Default: 24'h0FA031					
0x03	[23:10]	Thvcp_en	W/R	14'h03E8	
	[9:0]	Tben	W/R	10'h031	
Register 4 (0x04) Default: 24'h3E83E8					
0x04	[23:12]	Tmvcp_en	W/R	12'h3E8	
	[11:0]	Tsc_en	W/R	12'h3E8	
Register 5 (0x05) Default: 24'h0007CF					
0x05	[19:0]	Tcap	W/R	20'h007CF	
Register 6 (0x06) Default: 24'h001F3F					
0x06	[16:0]	Tvref	W/R	17'h01F3F	
Register 7 (0x07) Default: 24'h000000					
0x07	[12:0]	SRAM_START_ADDR	W/R (NR)	13'h0000	
Register 8 (0x08) Default: 24'h000000					
0x08	[17:0]	FLSH_START_ADDR	W/R (NR)	18'h00000	
Register 9 (0x09) Default: 24'h000000					
0x09	[18:6]	LENGTH	W/R (NR)	13'h0000	
	[5]	IRQ_EN	W/R (NR)	1'h0	
	[4:1]	CMD	W/R (NR)	4'h0	
	[0]	GO	W/R (NR)	1'h0	
Register 10 (0x0A) Default: 24'h00001E					
0x0A	[4]	VREF_SLEEP	W/R	1'h1	
	[3]	COMP_SLEEP	W/R	1'h1	
	[2]	COMP_CLKENB	W/R	1'h1	
	[1]	COMP_ISOL	W/R	1'h1	
Register 12 (0x0C) Default: 24'h000001					
0x0C	[3]	WRAP_EXT	W/R	1'h0	
	[2]	UPDATE_ADDR_EXT	W/R	1'h0	
	[1:0]	BIT_EN_EXT	W/R	2'h1	
Register 13 (0x0D) Default: 24'h0FFFFF					
0x0D	[19:0]	TIMEOUT_EXT	W/R	20'hFFFFF	
Register 14 (0x0E) Default: 24'h000000					
0x0E	[12:0]	SRAM_START_ADDR_EXT	W/R	13'h0000	

Continued on next page

Continued from previous page

Reg Addr	Bit Field	Reg Name	Property	Size & Reset	Remark
Register 15 (0x0F) Default: 24'h001000					
0x0F	[15:8]	INT_RPLY_SHORT_ADDR	W/R	8'h10	
	[7:0]	INT_RPLY_REG_ADDR	W/R	8'h00	
Register 16 (0x10) Default: 24'h000000					
0x10	[22]	BOOT_FLAG_SLEEP	W/R	1'h0	
	[21]	BOOT_FLAG_ECC_ERROR	W/R	1'h0	
	[20]	BOOT_FLAG_WRONG_HEADER	W/R	1'h0	
	[19]	BOOT_FLAG_PWDN	W/R	1'h0	
	[18]	BOOT_FLAG_INVALID_CMND	W/R	1'h0	
	[17]	BOOT_FLAG_CHKSUM_ERROR	W/R	1'h0	
	[16]	BOOT_FLAG_SUCCESS	W/R	1'h0	
	[1:0]	BOOT_REG_PATTERN	W/R	2'h0	
Register 17 (0x11) Default: 24'h00002E					
0x11	[5]	FLASH_POWER_DO_VREFCOMP	W/R	1'h1	
	[3]	FLASH_POWER_DO_FLSH	W/R	1'h1	
	[2]	FLASH_POWER_IRQ_EN	W/R	1'h1	
	[1]	FLASH_POWER_SEL_ON	W/R	1'h1	
	[0]	FLASH_POWER_GO	W/R	1'h0	
Register 18 (0x12) Default: 24'h000000					
0x12	[6]	IRQ_PWR_ON_WUP	W/R	1'h0	
	[5:3]	SEL_PWR_ON_WUP	W/R	3'h0	
	[2]	FLASH_AUTO_USE_CUSTOM	W/R	1'h0	
	[1]	FLASH_AUTO_OFF	W/R	1'h0	
	[0]	FLASH_AUTO_ON	W/R	1'h0	
Register 19 (0x13) Default: 24'h000000					
0x13	[19:1]	PP_STR_LIMIT	W/R (NR)	19'h00000	
	[0]	PP_STR_EN	W/R (NR)	1'h0	
Register 20 (0x14) Default: 24'h000001					
0x14	[4]	PP_NO_ERR_DETECTION	W/R	1'h0	
	[3]	PP_USE_FAST_PROG	W/R	1'h0	
	[2]	PP_WRAP	W/R	1'h0	
	[1:0]	PP_BIT_EN_EXT	W/R	2'h1	
Register 21 (0x15) Default: 24'h000000					
0x15	[17:0]	PP_FLSH_ADDR	W/R	18'h00000	
Register 22 (0x16) Default: 24'h000000					
0x16	[18:0]	PP_LENGTH_STREAMED	W/R (NR)	19'h00000	
Register 23 (0x17) Default: 24'h000000					
0x17	[23]	PP_FLAG_END_OF_FLASH	W/R (NR)	1'h0	
	[22]	PP_FLAG_STR_LIMIT	W/R (NR)	1'h0	
	[21]	PP_FLAG_COPY_LIMIT	W/R (NR)	1'h0	
	[18:0]	PP_LENGTH_COPIED	W/R (NR)	19'h00000	
Register 24 (0x18) Default: 24'h000031					
0x18	[5:2]	CLK_RING_SEL	W/R	4'hC	
	[1:0]	CLK_DIV_SEL	W/R	2'h1	
Register 25 (0x19) Default: 24'h000C03					
0x19	[11]	DISABLE_BYPASS_MIRROR	W/R	1'h1	
	[10:7]	COMP_CTRL_I_1STG	W/R	4'h8	
	[6:3]	COMP_CTRL_I_2STG_BAR	W/R	4'h0	
	[2:0]	COMP_CTRL_VOUT	W/R	3'h3	

Continued on next page

Continued from previous page

Reg Addr	Bit Field	Reg Name	Property	Size & Reset	Remark
Register 27 (0x1B) Default: 24'h000000					
0x1B	[7:0]	IRQ_PAYLOAD	R	8'h00	
Register 30 (0x1E) Default: 24'h000000					
0x1E	[23:0]	FLS2LC_REG_WR_DATA	R	24'h000000	
Register 31 (0x1F) Default: 24'h000001					
0x1F	[0]	FORCE_RESETN	W/R	1'h1	
Register 32 (0x20) Default: 24'h001087					
0x20	[14:10]	FLSH_SET0	W/R	5'h04	
	[9:5]	FLSH_SET1	W/R	5'h04	
	[4:0]	FLSH_SNT	W/R	5'h07	
Register 33 (0x21) Default: 24'h001084					
0x21	[14:10]	FLSH_SPT0	W/R	5'h04	
	[9:5]	FLSH_SPT1	W/R	5'h04	
	[4:0]	FLSH_SPT2	W/R	5'h04	
Register 34 (0x22) Default: 24'h0000E7					
0x22	[9:5]	FLSH_SYT0	W/R	5'h07	
	[4:0]	FLSH_SYT1	W/R	5'h07	
Register 35 (0x23) Default: 24'h008C67					
0x23	[19:15]	FLSH_SRT0	W/R	5'h01	
	[14:10]	FLSH_SRT1	W/R	5'h03	
	[9:5]	FLSH_SRT2	W/R	5'h03	
	[4:0]	FLSH_SRT3	W/R	5'h07	
Register 36 (0x24) Default: 24'h001CE1					
0x24	[14:10]	FLSH_SRT4	W/R	5'h07	
	[9:5]	FLSH_SRT5	W/R	5'h07	
	[4:0]	FLSH_SRT6	W/R	5'h01	
Register 38 (0x26) Default: 24'h0D7788					
0x26	[19:16]	FLSH_SPIG	W/R	4'hD	
	[15:12]	FLSH_SRIG	W/R	4'h7	
	[11:8]	FLSH_SVR0	W/R	4'h7	
	[7:4]	FLSH_SVR1	W/R	4'h8	
	[3:0]	FLSH_SVR2	W/R	4'h8	
Register 39 (0x27) Default: 24'h011BC8					
0x27	[20:16]	FLSH_SHVE	W/R	5'h01	
	[15:11]	FLSH_SHVP	W/R	5'h03	
	[10:6]	FLSH_SHVCT	W/R	5'h0F	
	[5:0]	FLSH_SMV	W/R	6'h08	
Register 40 (0x28) Default: 24'h0000E7					
0x28	[9:5]	FLSH_SMVCT0	W/R	5'h07	
	[4:0]	FLSH_SMVCT1	W/R	5'h07	
Register 42 (0x2A) Default: 24'h000002					
0x2A	[5:0]	FLSH_SAB	W/R	6'h02	
Register 48 (0x30) Default: 24'hF00000					
0x30	[23:16]	STR_WR_CH1_ALT_ADDR	W/R	8'hF0	
Register 49 (0x31) Default: 24'h000000					
0x31	[23:16]	STR_WR_CH1_ALT_REG_WR	W/R	8'h00	
Register 50 (0x32) Default: 24'hC01FFF					
0x32	[23]	STR_WR_CH1_EN	W/R	1'h1	
	[22]	STR_WR_CH1_WRP	W/R	1'h1	

Continued on next page

Continued from previous page					
Reg Addr	**Bit Field**	**Reg Name**	**Property**	**Size & Reset**	**Remark**
	[21]	STR_WR_CH1_DBLB	W/R	1'h0	
	[12:0]	STR_WR_CH1_BUF_LEN	W/R	13'h1FFF	
Register 51 (0x33) Default: 24'h000000					
0x33	[12:0]	STR_WR_CH1_BUF_OFF	W/R	13'h0000	
Register 52 (0x34) Default: 24'hF00000					
0x34	[23:16]	STR_WR_CH0_ALT_ADDR	W/R	8'hF0	
Register 53 (0x35) Default: 24'h000000					
0x35	[23:16]	STR_WR_CH0_ALT_REG_WR	W/R	8'h00	
Register 54 (0x36) Default: 24'hC01FFF					
0x36	[23]	STR_WR_CH0_EN	W/R	1'h1	
	[22]	STR_WR_CH0_WRP	W/R	1'h1	
	[21]	STR_WR_CH0_DBLB	W/R	1'h0	
	[12:0]	STR_WR_CH0_BUF_LEN	W/R	13'h1FFF	
Register 55 (0x37) Default: 24'h000000					
0x37	[12:0]	STR_WR_CH0_BUF_OFF	W/R	13'h0000	
Register 58 (0x3A) Default: 24'h800000					
0x3A	[23]	BLK_WR_EN	W/R	1'h1	
Register 71 (0x47) Default: 24'h000000					
0x47	[23]	ACT_RST	W/R	1'h0	

Table 1: FLPv3L MBus Register File Mapping

5.2 MBus Register Descriptions

5.2.1 Register 0 (0x00)

Tcyc_read Reg 0x00, Bit Field: [23:19], Default: 5'h1F, W/R

Number of CLK_{main} cycles to set Read cycle. The default value corresponds to $\sim 32\mu$s.

T3us Reg 0x00, Bit Field: [18:13], Default: 6'h02, W/R

Number of CLK_{main} cycles to measure 3μs. The default value corresponds to $\sim 3\mu$s.

T5us Reg 0x00, Bit Field: [12:7], Default: 6'h04, W/R

Number of CLK_{main} cycles to measure 5μs. The default value corresponds to $\sim 5\mu$s.

T10us Reg 0x00, Bit Field: [6:0], Default: 7'h09, W/R

Number of CLK_{main} cycles to measure 10μs. The default value corresponds to $\sim 10\mu$s.

5.2.2 Register 1 (0x01)

Tcyc_prog Reg 0x01, Bit Field: [23:8], Default: 16'h007F, W/R

Number of CLK_{main} cycles to set Program cycle. The default value corresponds to $\sim$128μs.

Tprog Reg 0x01, Bit Field: [7:0], Default: 8'h09, W/R

Number of CLK_{main} cycles to set Program time. The default value corresponds to $\sim$10μs.

5.2.3 Register 2 (0x02)

Terase Reg 0x02, Bit Field: [15:0], Default: 16'h0100, W/R

Number of CLK_{main} cycles to set Erase time. The default value corresponds to $\sim$256μs.

5.2.4 Register 3 (0x03)

Thvcp_en Reg 0x03, Bit Field: [23:10], Default: 14'h03E8, W/R

Number of CLK_{main} cycles to set delay between HVCP_EN and MVCP_EN. The default value corresponds to $\sim$1ms.

Tben Reg 0x03, Bit Field: [9:0], Default: 10'h031, W/R

Number of CLK_{main} cycles to set flash initial setup time. The default value corresponds to $\sim$50μs.

5.2.5 Register 4 (0x04)

Tmvcp_en Reg 0x04, Bit Field: [23:12], Default: 12'h3E8, W/R

Number of CLK_{main} cycles to set delay between MVCP_EN and SC_EN. The default value corresponds to $\sim$1ms.

Tsc_en Reg 0x04, Bit Field: [11:0], Default: 12'h3E8, W/R

Number of CLK_{main} cycles to set delay after SC_EN. The default value corresponds to $\sim$1ms.

5.2.6 Register 5 (0x05)

Tcap Reg 0x05, Bit Field: [19:0], Default: 20'h007CF, W/R

Number of CLK_{main} cycles to set capacitor-charging time. The default value corresponds to $\sim$2ms.

5.2.7 Register 6 (0x06)

Tvref Reg 0x06, Bit Field: [16:0], Default: 17'h01F3F, W/R

Number of CLK_{main} cycles to set Vref settling time. The default value corresponds to $\sim$8ms.

Table 2: Summary of Timing Parameters

Name	Supposed Value	Default Value
T3us	3μs	6'h02
T5us	5μs	6'h04
T10us	10μs	7'h09
Tprog	10μs	8'h09
Terase	256μs	16'h0100
Tcyc_read	32μs	5'h1F
Tcyc_prog	128μs	16'h007F
Thvcp_en	1ms	14'h03E8
Tmvcp_en	1ms	12'h3E8
Tsc_en	1ms	12'h3E8
Tben	50μs	10'h031
Tcap	2ms	20'h007CF
Tvref	8ms	17'h01F3F

5.2.8 Register 7 (0x07)

SRAM_START_ADDR Reg 0x07, Bit Field: [12:0], Default: 13'h0000, W/R (NR)

If the operation specified in CMD need to access SRAM, it uses SRAM_START_ADDR as an initial value of the SRAM pointer and increments from there if needed. SRAM_START_ADDR's LSB indicates 1 word (32-bit).

5.2.9 Register 8 (0x08)

FLSH_START_ADDR Reg 0x08, Bit Field: [17:0], Default: 18'h00000, W/R (NR)

If the operation specified in CMD need to access Flash, it uses FLSH_START_ADDR as an initial value of the Flash pointer and increments from there if needed. FLSH_START_ADDR's LSB indicates 1 word (32-bit).

5.2.10 Register 9 (0x09)

LENGTH Reg 0x09, Bit Field: [18:6], Default: 13'h0000, W/R (NR)

Sets the length (= Number of words - 1) to be used in CMD operation.

IRQ_EN Reg 0x09, Bit Field: [5], Default: 1'h0, W/R (NR)

Enables/Disables MBus IRQ message.
If 1, Flash layer will send an MBus IRQ message at the end of CMD operation. The IRQ data indicates

PASS/FAIL status of the operation.
If 0, Flash layer will not send an MBus IRQ message at the end of CMD operation.

CMD Reg 0x09, Bit Field: [4:1], Default: 4'h0, W/R (NR)

Specifies the operation to be executed. See Table 3

Table 3: CMD

CMD	Description	
4'h0	Invalid Command	-
4'h1	Copy from Flash to SRAM	Section 7
4'h2	Copy from SRAM to Flash	Section 8.1
4'h3	Copy from SRAM to Flash (Fast Program)	Section 8.2
4'h4	Erase Flash (Page Erase)	Section 9.1
4'h5	Reserved	-
4'h6	External Input Stream Write to SRAM	Section 10
4'h7	Erase Reference Array in Flash	Section 9.2
4'h8	Start Boot-Up	Section 12
4'h9	Measure Clock Frequency	Section 14
4'hA $\sim$ 4'hF	Reserved	-

GO Reg 0x09, Bit Field: [0], Default: 1'h0, W/R (NR)

Writing 1 to this register initiates the operation specified in CMD. Once the operation is done, this register is automatically reset to 0. User does NOT have to manually reset this to 0.
Writing 0 to this register is ignored.

5.2.11 Register 10 (0x0A)

VREF_SLEEP Reg 0x0A, Bit Field: [4], Default: 1'h1, W/R

Turns on/off Vref circuit inside the Flash.
If 1, Vref circuit is off.
If 0, Vref circuit is on.

COMP_SLEEP Reg 0x0A, Bit Field: [3], Default: 1'h1, W/R

Turns on/off the comparator inside Voltage Clamper.
If 1, the comparator is off.
If 0, the comparator is on.

COMP_CLKENB Reg 0x0A, Bit Field: [2], Default: 1'h1, W/R

Enables/Disables CLK_{comp}.
If 1, CLK_{comp} is disabled (CLK_{comp} not running).
If 0, CLK_{comp} is enabled (CLK_{comp} running).

COMP_ISOL Reg 0x0A, Bit Field: [1], Default: 1'h1, W/R

Enables/Disables isolation of Voltage Clamper.
If 1, Voltage Clamper becomes isolated.
If 0, Voltage Clamper becomes un-isolated.

5.2.12 Register 12 (0x0C)

WRAP_EXT Reg 0x0C, Bit Field: [3], Default: 1'h0, W/R

Enables/Disables wrapping during the External Streaming.
If 1, wrapping is enabled. Once the last address of SRAM is written, the next input streaming is written to the first address of SRAM.
If 0, wrapping is disabled. Once the last address of SRAM is written, following input streaming is ignored.

UPDATE_ADDR_EXT Reg 0x0C, Bit Field: [2], Default: 1'h0, W/R

Enables/Disables the update of SRAM_START_ADDR_EXT. If 1, SRAM_START_ADDR_EXT is updated to the next address. For example, if the current External Streaming ends with writing into ADDR[N], then SRAM_START_ADDR_EXT will be updated to N+1, so that the next External Streaming can start writing from ADDR[N+1]. If 0, SRAM_START_ADDR_EXT is not updated and will remain same.

BIT_EN_EXT Reg 0x0C, Bit Field: [1:0], Default: 2'h1, W/R

Enables/Disables DATA_EXT[1:0] pads. See Table 4.

Table 4: BIT_EN_EXT Setting

BIT_EN_EXT	DATA_EXT[1]	DATA_EXT[0]
2'h0	Disabled	Disabled
2'h1	Disabled	Enabled
2'h2	Enabled	Disabled
2'h3	Enabled	Enabled

If BIT_EN_EXT=2'h3, DATA_EXT[1] becomes MSB, and DATA_EXT[0] becomes LSB.

5.2.13 Register 13 (0x0D)

TIMEOUT_EXT Reg 0x0D, Bit Field: [19:0], Default: 20'hFFFFF, W/R

Number of CLK_{main} cycles that will trigger the time-out error during External Streaming. TIMEOUT_EXT=0 will NOT trigger the time-out error, and it will be in the streaming mode indefinitely until one of the following events occurs:

- The number of words streamed becomes equal to the value specified in LENGTH (plus one).
- User manually resets the controller (main FSM) by writing 0 to FORCE_RESETN.

5.2.14 Register 14 (0x0E)

SRAM_START_ADDR_EXT Reg 0x0E, Bit Field: [12:0], Default: 13'h0000, W/R

SRAM Address that External Streaming starts from.

5.2.15 Register 15 (0x0F)

INT_RPLY_SHORT_ADDR Reg 0x0F, Bit Field: [15:8], Default: 8'h10, W/R

Destination short address used for the controller (main FSM)'s IRQ.

INT_RPLY_REG_ADDR Reg 0x0F, Bit Field: [7:0], Default: 8'h00, W/R

Destination register index used for the controller (main FSM)'s IRQ.

5.2.16 Register 16 (0x10)

BOOT_FLAG_SLEEP Reg 0x10, Bit Field: [22], Default: 1'h0, W/R

Updated to 1 if a boot-up program finishes with TAIL_SLEEP.

BOOT_FLAG_ECC_ERROR Reg 0x10, Bit Field: [21], Default: 1'h0, W/R

Updated to 1 if a boot-up program gets aborted due to an ECC error (more than 2-bit error).

BOOT_FLAG_WRONG_HEADER Reg 0x10, Bit Field: [20], Default: 1'h0, W/R

Updated to 1 if a boot-up program gets aborted due to a wrong header.

BOOT_FLAG_PWDN Reg 0x10, Bit Field: [19], Default: 1'h0, W/R

Updated to 1 if a boot-up program finishes with TAIL_PWDN.

BOOT_FLAG_INVALID_CMND Reg 0x10, Bit Field: [18], Default: 1'h0, W/R

Updated to 1 if a boot-up program gets aborted due to an invalid command.

BOOT_FLAG_CHKSUM_ERROR Reg 0x10, Bit Field: [17], Default: 1'h0, W/R

Updated to 1 if a boot-up program gets aborted due to a checksum error.

BOOT_FLAG_SUCCESS Reg 0x10, Bit Field: [16], Default: 1'h0, W/R

Updated to 1 if a boot-up program finishes with TAIL_IDLE or TAIL_PWDN or TAIL_SLEEP.

BOOT_REG_PATTERN Reg 0x10, Bit Field: [1:0], Default: 2'h0, W/R

2-bit pattern used for distinguishing the first reset release. At the first reset release after POR, only if BOOT_DIS pad is floating or grounded, the controller (main FSM) updates this register to 2'h3. This is for triggering the auto boot-up sequence only at the first reset release.

5.2.17 Register 17 (0x11)

FLASH_POWER_DO_VREFCOMP Reg 0x11, Bit Field: [5], Default: 1'h1, W/R

Determines whether the auto power-up/down includes Voltage Clamper. This is effective only if the power-up/down is triggered by FLASH_POWER_GO, or if (FLASH_AUTO_ON=1 or FLASH_AUTO_OFF=1) and (FLASH_AUTO_USE_CUSTOM=1).
If 1, Voltage Clamper is included in the power-up/down sequence.
If 0, Voltage Clamper is not included in the power-up/down sequence.

FLASH_POWER_DO_FLSH Reg 0x11, Bit Field: [3], Default: 1'h1, W/R

Determines whether the auto power-up/down includes the Flash. This is effective only if the power-up/down is triggered by FLASH_POWER_GO, or if (FLASH_AUTO_ON=1 or FLASH_AUTO_OFF=1) and (FLASH_AUTO_USE_CUSTOM=1).
If 1, the Flash is included in the power-up/down sequence.
If 0, the Flash is not included in the power-up/down sequence.

FLASH_POWER_IRQ_EN Reg 0x11, Bit Field: [2], Default: 1'h1, W/R

Enables/Disables MBus IRQ message at the end of the power-up/down sequence triggered by FLASH_POWER_GO.
If 1, Flash layer sends an MBus IRQ message at the end of the power-up/down sequence triggered by FLASH_POWER_GO.
If 0, Flash layer does not send an MBus IRQ message at the end of the power-up/down sequence triggered by FLASH_POWER_GO.

FLASH_POWER_SEL_ON Reg 0x11, Bit Field: [1], Default: 1'h1, W/R

Chooses whether it turns on or off the selected blocks in FLASH_POWER_DO_VREFCOMP and FLASH_POWER_DO_FLSH during the power-up/down sequence triggered by FLASH_POWER_GO.
If 1, it turns on the selected blocks.
If 0, it turns off the selected blocks.

FLASH_POWER_GO Reg 0x11, Bit Field: [0], Default: 1'h0, W/R

Writing 1 to this register will initiate the power-up/down sequence as specified in FLASH_POWER_SEL_ON, FLASH_POWER_DO_VREFCOMP, FLASH_POWER_DO_FLSH. This register will be automatically reset to

0 at the end of the power-up/down sequence, so that user does not have to manually reset to 0. Writing 0 to this register is ignored.

5.2.18 Register 18 (0x12)

IRQ_PWR_ON_WUP Reg 0x12, Bit Field: [6], Default: 1'h0, W/R

Enables/Disables MBus IRQ message at the end of the auto-power-up/down sequence upon system wakeup.
If 1, Flash layer sends an MBus IRQ message at the end of the auto-power-up/down sequence upon system wakeup.
If 0, Flash layer does not send an MBus IRQ message at the end of the auto-power-up/down sequence upon system wakeup.

SEL_PWR_ON_WUP Reg 0x12, Bit Field: [5:3], Default: 3'h0, W/R

Enables/Disables the auto-power-up/down sequence upon system wakeup. See Table 5.

Table 5: SEL_PWR_ON_WUP Setting

SEL_PWR_ON_WUP	Description
3'h0	Auto-power-up/down upon system wakeup is disabled
3'h1	Turn-on/off Voltage Clamper only
3'h2	Reserved
3'h3	Turn-on/off both Voltage Clamper and the Flash
3'h4 $\sim$ 3'h7	Reserved

FLASH_AUTO_USE_CUSTOM Reg 0x12, Bit Field: [2], Default: 1'h0, W/R

Sets whether the controller (main FSM) can automatically selects what to turn-on/off during the auto-power-up/down sequence.
If 1, auto-power-up/down sequence includes only the blocks specified in FLASH_POWER_DO_VREFCOMP and FLASH_POWER_DO_FLSH.
If 0, the controller (main FSM) will automatically choose what to turn-on/off depending on the operation.

FLASH_AUTO_OFF Reg 0x12, Bit Field: [1], Default: 1'h0, W/R

Sets whether to do auto-power-down sequence at the end of a GO operation.
If 1, auto-power-down sequence will be initiated at the end of a GO operation.
If 0, auto-power-down sequence will not be initiated at the end of a GO operation.

FLASH_AUTO_ON Reg 0x12, Bit Field: [0], Default: 1'h0, W/R

Sets whether to do auto-power-up sequence at the end of a GO operation.
If 1, auto-power-up sequence will be initiated at the end of a GO operation.
If 0, auto-power-up sequence will not be initiated at the end of a GO operation.

5.2.19 Register 19 (0x13)

PP_STR_LIMIT Reg 0x13, Bit Field: [19:1], Default: 19'h00000, W/R (NR)

Sets the limit on the number of words to be ping-pong streamed. The unit is 'word (32-bit)'.
PP_STR_LIMIT=0 means no limit.

PP_STR_EN Reg 0x13, Bit Field: [0], Default: 1'h0, W/R (NR)

Enables/Disables ping-pong streaming.
If 1, the ping-pong streaming will start upon one of the following events:

- There is no GO operation going on, and Flash layer starts receiving an MBus memory streaming message.
- GO operation starts with CMD=4'h6 (External Streaming).

If 0, the ping-pong streaming is disabled.

5.2.20 Register 20 (0x14)

PP_NO_ERR_DETECTION Reg 0x14, Bit Field: [4], Default: 1'h0, W/R

Enables/Disables the detection of buffer-overrun error during ping-pong streaming. Other errors can be still detected.
If 1, the detection of buffer-overrun error is disabled.
If 0, the detection of buffer-overrun error is enabled.

PP_USE_FAST_PROG Reg 0x14, Bit Field: [3], Default: 1'h0, W/R

Enables/Disables the fast-program mode during ping-pong streaming. The fast-program holds PROG and NVSTR signals during the whole page program, resulting in higher power consumption with better throughput.
If 1, the fast-program is enabled.
If 0, the fast-program is disabled, and it uses the normal-program.

PP_WRAP Reg 0x14, Bit Field: [2], Default: 1'h0, W/R

Enables/Disables the Flash address wrapping during ping-pong streaming.
If 1, PP_FLSH_ADDR becomes 0 after the last Flash address.
If 0, PP_FLSH_ADDR does not become 0 after the last Flash address. Instead, it will stay at the last Flash address.

PP_BIT_EN_EXT Reg 0x14, Bit Field: [1:0], Default: 2'h1, W/R

Enables/Disables DATA_EXT[1:0] pads during ping-pong streaming with CMD=4'h6 (External Streaming). See Table 6.

Table 6: PP_BIT_EN_EXT Setting

PP_BIT_EN_EXT	DATA_EXT[1]	DATA_EXT[0]
2'h0	Disabled	Disabled
2'h1	Disabled	Enabled
2'h2	Enabled	Disabled
2'h3	Enabled	Enabled

If PP_BIT_EN_EXT=2'h3, DATA_EXT[1] becomes MSB, and DATA_EXT[0] becomes LSB.

5.2.21 Register 21 (0x15)

PP_FLSH_ADDR Reg 0x15, Bit Field: [17:0], Default: 18'h00000, W/R

Flash address pointer used during ping-pong streaming. LSB indicates a word (32-bit). It automatically increments during the ping-pong streaming. User may want to set this pointer before starting ping-pong streaming.

5.2.22 Register 22 (0x16)

PP_LENGTH_STREAMED Reg 0x16, Bit Field: [18:0], Default: 19'h00000, W/R (NR)

Indicates the number of words streamed so far during ping-pong streaming (the number of words written into SRAM). User may want to reset this register before starting a new ping-pong streaming.

5.2.23 Register 23 (0x17)

PP_FLAG_END_OF_FLASH Reg 0x17, Bit Field: [23], Default: 1'h0, W/R (NR)

Updated to 1 if the Flash reaches its last address during ping-pong streaming. User may want to reset this register before starting a new ping-pong streaming.

PP_FLAG_STR_LIMIT Reg 0x17, Bit Field: [22], Default: 1'h0, W/R (NR)

Updated to 1 if PP_LENGTH_STREAMED becomes equal to PP_STR_LIMIT. User may want to reset this register before starting a new ping-pong streaming.

PP_FLAG_COPY_LIMIT Reg 0x17, Bit Field: [21], Default: 1'h0, W/R (NR)

Updated to 1 if PP_LENGTH_COPIED becomes equal to PP_STR_LIMIT. User may want to reset this register before starting a new ping-pong streaming.

PP_LENGTH_COPIED Reg 0x17, Bit Field: [18:0], Default: 19'h00000, W/R (NR)

Indicates the number of words copied so far during ping-pong streaming (the number of words copied from SRAM to the Flash). User may want to reset this register before starting a new ping-pong streaming.

5.2.24 Register 24 (0x18)

CLK_RING_SEL Reg 0x18, Bit Field: [5:2], Default: 4'hC, W/R

Clock generator ring configuration.

CLK_DIV_SEL Reg 0x18, Bit Field: [1:0], Default: 2'h1, W/R

Clock generator divider configuration.

5.2.25 Register 25 (0x19)

DISABLE_BYPASS_MIRROR Reg 0x19, Bit Field: [11], Default: 1'h1, W/R

Enables/Disables a mirror circuit in Voltage Clamper.
If 1, the current to charge the output capacitor is determined by a current reference and two current mirrors.
If 0, the output capacitor is charged through one PMOS transistor directly from 3.6V supply. However, the output will be still regulated by a comparator.

COMP_CTRL_I_1STG Reg 0x19, Bit Field: [10:7], Default: 4'h8, W/R

It changes a current copy ratio of the NMOS current mirror (1x, 2x, 4x, 8x). The copied current is copied one more time through the PMOS mirror to charge the on-chip capacitor.

COMP_CTRL_I_2STG_BAR Reg 0x19, Bit Field: [6:3], Default: 4'h0, W/R

It changes a current copy ratio of the PMOS current mirror (1x, 2x, 4x, 8x). The control bits are inverted. This PMOS mirror copies current from the NMOS mirror to charge the on-chip capacitor.

COMP_CTRL_VOUT Reg 0x19, Bit Field: [2:0], Default: 3'h3, W/R

It controls the clamping voltage of the output capacitor. Higher tuning bit value gives higher output voltage.

5.2.26 Register 27 (0x1B)

IRQ_PAYLOAD Reg 0x1B, Bit Field: [7:0], Default: 8'h00, R

This register is internally used to hold the controller's last IRQ payload.

5.2.27 Register 30 (0x1E)

FLS2LC_REG_WR_DATA Reg 0x1E, Bit Field: [23:0], Default: 24'h000000, R

This register is internally used in the controller.

5.2.28 Register 31 (0x1F)

FORCE_RESETN Reg 0x1F, Bit Field: [0], Default: 1'h1, W/R

Writing 0 in this register immediately (and asynchronously) resets the controller (FLPv3L _CTRL). It does not automatically release the reset, thus the user must manually write 1 in this register to release the reset. It is intended to be used to terminate the indefinite listening mode (either in external streaming or in MBus streaming)

5.2.29 Register 32 (0x20)

FLSH_SET0 Reg 0x20, Bit Field: [14:10], Default: 5'h04, W/R

Flash ERASE Delay0 tuning

FLSH_SET1 Reg 0x20, Bit Field: [9:5], Default: 5'h04, W/R

Flash ERASE Delay1 tuning

FLSH_SNT Reg 0x20, Bit Field: [4:0], Default: 5'h07, W/R

Flash NVSTR Delay tuning

5.2.30 Register 33 (0x21)

FLSH_SPT0 Reg 0x21, Bit Field: [14:10], Default: 5'h04, W/R

Flash PROG Delay0 tuning

FLSH_SPT1 Reg 0x21, Bit Field: [9:5], Default: 5'h04, W/R

Flash PROG Delay1 tuning

FLSH_SPT2 Reg 0x21, Bit Field: [4:0], Default: 5'h04, W/R

Flash PROG Delay2 tuning

5.2.31 Register 34 (0x22)

FLSH_SYT0 Reg 0x22, Bit Field: [9:5], Default: 5'h07, W/R

Flash YE Delay0 tuning

FLSH_SYT1 Reg 0x22, Bit Field: [4:0], Default: 5'h07, W/R

Flash YE Delay1 tuning

5.2.32 Register 35 (0x23)

FLSH_SRT0 Reg 0x23, Bit Field: [19:15], Default: 5'h01, W/R

Flash READ Delay0 tuning

FLSH_SRT1 Reg 0x23, Bit Field: [14:10], Default: 5'h03, W/R

Flash READ Delay1 tuning

FLSH_SRT2 Reg 0x23, Bit Field: [9:5], Default: 5'h03, W/R

Flash READ Delay2 tuning

FLSH_SRT3 Reg 0x23, Bit Field: [4:0], Default: 5'h07, W/R

Flash READ Delay3 tuning

5.2.33 Register 36 (0x24)

FLSH_SRT4 Reg 0x24, Bit Field: [14:10], Default: 5'h07, W/R

Flash READ Delay4 tuning

FLSH_SRT5 Reg 0x24, Bit Field: [9:5], Default: 5'h07, W/R

Flash READ Delay5 tuning

FLSH_SRT6 Reg 0x24, Bit Field: [4:0], Default: 5'h01, W/R

Flash READ Delay6 tuning

5.2.34 Register 38 (0x26)

FLSH_SPIG Reg 0x26, Bit Field: [19:16], Default: 4'hD, W/R

Flash Program Iref Generation tuning

FLSH_SRIG Reg 0x26, Bit Field: [15:12], Default: 4'h7, W/R

Flash Read Iref Generation tuning

FLSH_SVR0 Reg 0x26, Bit Field: [11:8], Default: 4'h7, W/R

Flash Vref Generation tuning for Upper-Left PMOS

FLSH_SVR1 Reg 0x26, Bit Field: [7:4], Default: 4'h8, W/R

Flash Vref Generation tuning for Bottom-Left PMOS

FLSH_SVR2 Reg 0x26, Bit Field: [3:0], Default: 4'h8, W/R

Flash Vref Generation tuning for Bottom-Right PMOS

5.2.35 Register 39 (0x27)

FLSH_SHVE Reg 0x27, Bit Field: [20:16], Default: 5'h01, W/R

Flash HV Pump Diode-Chain tuning for Erase

FLSH_SHVP Reg 0x27, Bit Field: [15:11], Default: 5'h03, W/R

Flash HV Pump Diode-Chain tuning for Program

FLSH_SHVCT Reg 0x27, Bit Field: [10:6], Default: 5'h0F, W/R

Flash HV Pump CEN Timing tuning for HV_EN

FLSH_SMV Reg 0x27, Bit Field: [5:0], Default: 6'h08, W/R

Flash MV Pump Diode-Chain tuning

5.2.36 Register 40 (0x28)

FLSH_SMVCT0 Reg 0x28, Bit Field: [9:5], Default: 5'h07, W/R

Flash MV Pump CEN Timing tuning for MV_EN

FLSH_SMVCT1 Reg 0x28, Bit Field: [4:0], Default: 5'h07, W/R

Flash MV Pump CEN Timing tuning for NVSTR2

5.2.37 Register 42 (0x2A)

FLSH_SAB Reg 0x2A, Bit Field: [5:0], Default: 6'h02, W/R

Flash Analog Buffer tuning

5.2.38 Register 48 (0x30)

STR_WR_CH1_ALT_ADDR Reg 0x30, Bit Field: [23:16], Default: 8'hF0, W/R

Alert Address (8-bit) for MBus memory streaming (Channel 1). Alerts are sent whenever the end of the buffer is reached. If DBLB is active, an alert is also sent when the halfway point of the buffer is reached. See 'MBus Specification' document for more details.

5.2.39 Register 49 (0x31)

STR_WR_CH1_ALT_REG_WR Reg 0x31, Bit Field: [23:16], Default: 8'h00, W/R

When an alert message is sent for MBus memory streaming (Channel 1), STR_WR_CH1_ALT_REG_WR populates [31:24] in the MBus Data. See 'MBus Specification' document for more details.

5.2.40 Register 50 (0x32)

STR_WR_CH1_EN Reg 0x32, Bit Field: [23], Default: 1'h1, W/R

Enables/Disables MBus memory streaming (Channel 1).
If 1, MBus memory streaming (Channel 1) is enabled.
If 0, MBus memory streaming (Channel 1) is disabled.

STR_WR_CH1_WRP Reg 0x32, Bit Field: [22], Default: 1'h1, W/R

Enables/Disables memory wrapping during MBus memory streaming (Channel 1).
If 1, Enabled: Write Address Counter for MBus memory streaming (Channel 1) is reset to STR_WR_CH1_BUF_OFF when the end of the buffer is reached.
If 0, Disabled: Write Address Counter for MBus memory streaming (Channel 1) is unchanged when the end of the buffer is reached. Also, STR_WR_CH1_EN is changed to 0 at this point.

STR_WR_CH1_DBLB Reg 0x32, Bit Field: [21], Default: 1'h0, W/R

Double-buffering mode for MBus memory streaming (Channel 1).
If 1, it generates an alert message halfway through the buffer in addition to at the end of the buffer.
If 0, it generates only when the end of the buffer is reached.

STR_WR_CH1_BUF_LEN Reg 0x32, Bit Field: [12:0], Default: 13'h1FFF, W/R

It defines the length of the buffer for MBus memory streaming (Channel 1). It is 'Buffer Length - 1', and the unit is 'word (32-bit)'. Hence FLPv3L's default value (13'h1FFF) indicates 32kB (8192 words).

5.2.41 Register 51 (0x33)

STR_WR_CH1_BUF_OFF Reg 0x33, Bit Field: [12:0], Default: 13'h0000, W/R

It defines the buffer offset for MBus memory streaming (Channel 1). The unit is 'word (32-bit)'.

5.2.42 Register 52 (0x34)

STR_WR_CH0_ALT_ADDR Reg 0x34, Bit Field: [23:16], Default: 8'hF0, W/R

Alert Address (8-bit) for MBus memory streaming (Channel 0). Alerts are sent whenever the end of the buffer is reached. If DBLB is active, an alert is also sent when the halfway point of the buffer is reached. See 'MBus Specification' document for more details.

5.2.43 Register 53 (0x35)

STR_WR_CH0_ALT_REG_WR Reg 0x35, Bit Field: [23:16], Default: 8'h00, W/R

When an alert message is sent for MBus memory streaming (Channel 0), STR_WR_CH0_ALT_REG_WR populates [31:24] in the MBus Data. See 'MBus Specification' document for more details.

5.2.44 Register 54 (0x36)

STR_WR_CH0_EN Reg 0x36, Bit Field: [23], Default: 1'h1, W/R

Enables/Disables MBus memory streaming (Channel 0).
If 1, MBus memory streaming (Channel 0) is enabled.
If 0, MBus memory streaming (Channel 0) is disabled.

STR_WR_CH0_WRP Reg 0x36, Bit Field: [22], Default: 1'h1, W/R

Enables/Disables memory wrapping during MBus memory streaming (Channel 0).
If 1, Enabled: Write Address Counter for MBus memory streaming (Channel 0) is reset to STR_WR_CH0_BUF_OFF when the end of the buffer is reached.
If 0, Disabled: Write Address Counter for MBus memory streaming (Channel 0) is unchanged when the end of the buffer is reached. Also, STR_WR_CH0_EN is changed to 0 at this point.

STR_WR_CH0_DBLB Reg 0x36, Bit Field: [21], Default: 1'h0, W/R

Double-buffering mode for MBus memory streaming (Channel 0).
If 1, it generates an alert message halfway through the buffer in addition to at the end of the buffer.
If 0, it generates only when the end of the buffer is reached.

STR_WR_CH0_BUF_LEN Reg 0x36, Bit Field: [12:0], Default: 13'h1FFF, W/R

It defines the length of the buffer for MBus memory streaming (Channel 0). It is 'Buffer Length - 1', and the unit is 'word (32-bit)'. Hence FLPv3L's default value (13'h1FFF) indicates 32kB (8192 words).

5.2.45 Register 55 (0x37)

STR_WR_CH0_BUF_OFF Reg 0x37, Bit Field: [12:0], Default: 13'h0000, W/R

It defines the buffer offset for MBus memory streaming (Channel 0). The unit is 'word (32-bit)'.

5.2.46 Register 58 (0x3A)

BLK_WR_EN Reg 0x3A, Bit Field: [23], Default: 1'h1, W/R

Enables/Disables MBus memory bulk write.
If 1, MBus memory bulk write is enabled.
If 0, MBus memory bulk write is disabled.

5.2.47 Register 71 (0x47)

ACT_RST Reg 0x47, Bit Field: [23], Default: 1'h0, W/R

Action Register Reset. This is NOT implemented in the current version of Layer Ctrl.

6 Power-Up/Down

In order to do any operation that requires the Flash operation, the following blocks must be turned-on.

- Voltage Clamper
- Flash

A correct power-up sequence should be followed to turn on those. Similarly, a correct power-down sequence should be followed to turn off those. User must make sure that they are turned off before the system goes into sleep.

If Auto Power-Up/Down is enabled (Section 6.3), the controller (main FSM) automatically handles the power-up/down sequence. Figure 5 shows internal waveforms used to turn-on/off Voltage Clamper.

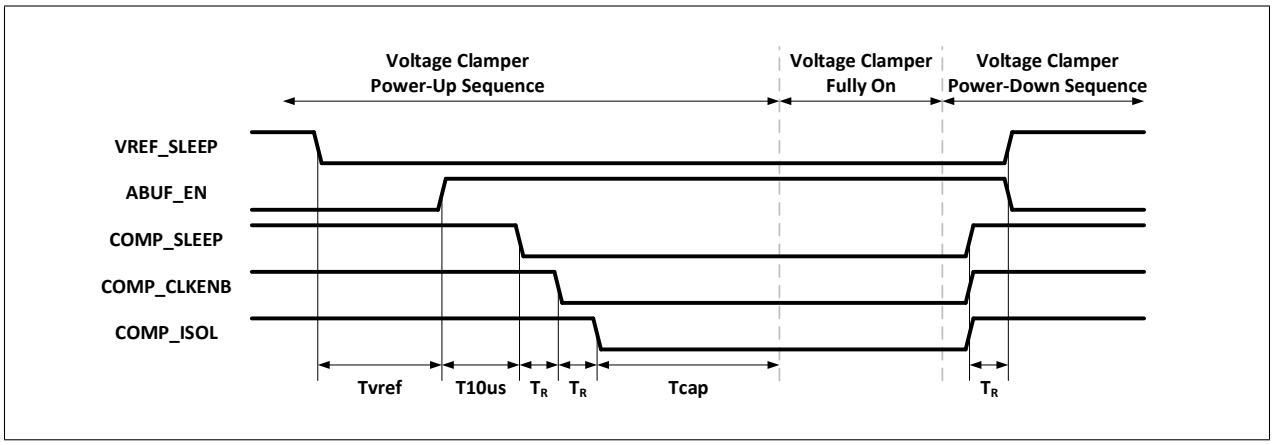

Figure 5: Voltage Clamper Power-Up/Down Waveform

VREF_SLEEP turns on/off a voltage reference circuit residing in the Flash. This voltage reference circuit provides the reference voltage to Voltage Clamper through an analog buffer.

ABUF_EN turns on/off this analog buffer residing in the Flash. Once this analog buffer becomes enabled, the reference voltage output from the Flash becomes valid. ABUF_EN is controlled by the controller (main FSM), so that user cannot manually control this signal.

COMP_SLEEP turns on/off Voltage Clamper.

COMP_CLKENB enables/disables the Voltage Clamper clock (CLK_{comp}). CLK_{comp} is generated from the Clock Generator (Section 13).

COMP_ISOL controls the isolation of Voltage Clamper. This is required because of level converter circuits inside Voltage Clamper.

T_R indicates a few CLKmain cycles required to update Register File (generally 10 20 CLK_{main} cycles). Tvref, T10us, Tcap are all set by MBus Register File. Tvref is required to wait until the voltage reference output becomes stabilized. Tcap is required to charge up the on-chip decap (shown in Figure 3).

Figure 6 shows internal waveforms used to turn-on/off the Flash.

All signals shown in Figure 6 are controlled by the controller (main FSM), so that user cannot manually control these signals.

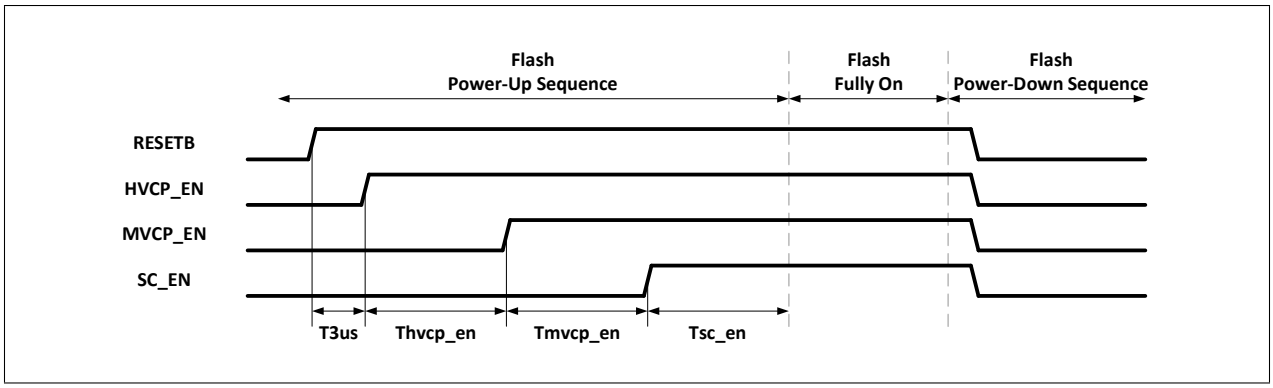

Figure 6: Flash Power-Up/Down Waveform

RESETB is a reset signal used in the Flash.

HVCP_EN is a high-voltage charge-pump enable signal.

MVCP_EN is a medium-voltage charge-pump enable signal.

SC_EN is a switch-cap enable signal.

T3us, T5us, Thvcp_en, Tmvcp_en, Tsc_en are all set by Register File.

Voltage Clamper must be fully on before the Flash becomes turned-on. Also, the Flash must be turned-off before Voltage Clamper becomes powered-down.

The Flash must be fully on before starting any Flash-related operation.

6.1 Manual Power-Up/Down

Signals that can be manually controlled is very limited. The only signals to which user has a direct access are:

- VREF_SLEEP
- COMP_SLEEP
- COMP_CLKENB
- COMP_ISOL

It is NOT recommended to manually control these signals. Since without using ABUF_EN, which is solely controlled by the controller (main FSM), Voltage Clamper cannot be fully on, thus no Flash operation can be performed correctly.

6.2 Semi-Auto Power-Up/Down

User can turn-on/off Voltage Clamper and the Flash using FLASH_POWER_GO signal. Follow instructions below.

6.2.1 Turn on Voltage Clamper and the Flash

1. Set Register 0x11 like below. This can be done using one MBus Register Write message.
   FLASH_POWER_DO_VREFCOMP = 1'h1
   FLASH_POWER_DO_FLSH = 1'h1
   FLASH_POWER_IRQ_EN = 1'h1
   FLASH_POWER_SEL_ON = 1'h1
   FLASH_POWER_GO = 1'h1
2. Once the power-up sequence is done, Flash layer will send an MBus interrupt message. See Table 7 for details on the interrupt payload.

6.2.2 Turn on Voltage Clamper Only

1. Set Register 0x11 like below. This can be done using one MBus Register Write message.
   FLASH_POWER_DO_VREFCOMP = 1'h1
   FLASH_POWER_DO_FLSH = 1'h0
   FLASH_POWER_IRQ_EN = 1'h1
   FLASH_POWER_SEL_ON = 1'h1
   FLASH_POWER_GO = 1'h1
2. Once the power-up sequence is done, Flash layer will send an MBus interrupt message. See Table 7 for details on the interrupt payload.

6.2.3 Turn on the Flash Only

1. Set Register 0x11 like below. This can be done using one MBus Register Write message.
   FLASH_POWER_DO_VREFCOMP = 1'h0
   FLASH_POWER_DO_FLSH = 1'h1
   FLASH_POWER_IRQ_EN = 1'h1
   FLASH_POWER_SEL_ON = 1'h1
   FLASH_POWER_GO = 1'h1
2. Once the power-up sequence is done, Flash layer will send an MBus interrupt message. See Table 7 for details on the interrupt payload.

6.2.4 Turn off Voltage Clamper and the Flash

1. Set Register 0x11 like below. This can be done using one MBus Register Write message.
   FLASH_POWER_DO_VREFCOMP = 1'h1
   FLASH_POWER_DO_FLSH = 1'h1
   FLASH_POWER_IRQ_EN = 1'h1
   FLASH_POWER_SEL_ON = 1'h0
   FLASH_POWER_GO = 1'h1
2. Once the power-down sequence is done, Flash layer will send an MBus interrupt message. See Table 7 for details on the interrupt payload.

6.2.5 Turn off Voltage Clamper Only

1. Set Register 0x11 like below. This can be done using one MBus Register Write message.
   FLASH_POWER_DO_VREFCOMP = 1'h1
   FLASH_POWER_DO_FLSH = 1'h0
   FLASH_POWER_IRQ_EN = 1'h1
   FLASH_POWER_SEL_ON = 1'h0
   FLASH_POWER_GO = 1'h1
2. Once the power-down sequence is done, Flash layer will send an MBus interrupt message. See Table 7 for details on the interrupt payload.

6.2.6 Turn off the Flash Only

1. Set Register 0x11 like below. This can be done using one MBus Register Write message.
   FLASH_POWER_DO_VREFCOMP = 1'h0
   FLASH_POWER_DO_FLSH = 1'h1
   FLASH_POWER_IRQ_EN = 1'h1
   FLASH_POWER_SEL_ON = 1'h0
   FLASH_POWER_GO = 1'h1
2. Once the power-down sequence is done, Flash layer will send an MBus interrupt message. See Table 7 for details on the interrupt payload.

FLASH_POWER_GO will be reset to 0 automatically once the power-up/down sequence is complete only if FLASH_POWER_IRQ_EN = 1'h1.

If user does not want to get the MBus interrupt message, set FLASH_POWER_IRQ_EN = 1'h0. However, in this case, FLASH_POWER_GO will not be reset to 0 automatically.

Table 7: Interrupt Payloads for Semi-Auto Power-Up/Down

Interrupt Payload	Description
8'hB5	Power-Up sequence completed
8'hBB	Power-Down sequence completed

6.3 Auto Power-Up/Down

The controller can handle the power-up/down sequence automatically if Auto Power-Up/Down is enabled. In order to enable or disable, follow the instructions below:

- To enable, set Register 0x12 like below.
  IRQ_PWR_ON_WUP = 1'h0
  SEL_PWR_ON_WUP = 3'h0
  FLASH_AUTO_USE_CUSTOM = 1'h0
  FLASH_AUTO_OFF = 1'h1
  FLASH_AUTO_ON = 1'h1
- To disable, set Register 0x12 like below.
  IRQ_PWR_ON_WUP = 1'h0
  SEL_PWR_ON_WUP = 3'h0

FLASH_AUTO_USE_CUSTOM = 1'h0
FLASH_AUTO_OFF = 1'h0
FLASH_AUTO_ON = 1'h0

If enabled, the controller powers-up Voltage Clamper and the Flash before starting an operation, such as GO operations or ping-pong streaming. Once the operation is done, then the controller powers-down Voltage Clamper and the Flash.

User can separately enable/disable power-up and power-down sequences.

- If FLASH_AUTO_ON = 1'h1 and FLASH_AUTO_OFF = 1'h0, only auto power-up is enabled. The controller turns on Voltage Clamper and the Flash before starting an operation. However, the controller does NOT turn off Voltage Clamper and the Flash after the operation is done.
- If FLASH_AUTO_ON = 1'h0 and FLASH_AUTO_OFF = 1'h1, only auto power-down is enabled. The controller does NOT turn on Voltage Clamper and the Flash before starting an operation. Hence, the operation may not be performed correctly unless user does the correct power-up sequence beforehand. The controller turns off Voltage Clamper and the Flash after the operation is done.

User can also choose blocks to be included in the auto power-up/down sequence.

- If FLASH_AUTO_USE_CUSTOM = 1'h1:
  - Auto power-up/down sequence turns-on/off Voltage Clamper only if FLASH_POWER_DO_VREFCOMP = 1'h1.
  - Auto power-up/down sequence turns-on/off the Flash only if FLASH_POWER_DO_FLSH = 1'h1.
- If FLASH_AUTO_USE_CUSTOM = 1'h0:
  - Auto power-up/down sequence turns-on/off both Voltage Clamper and the Flash.

6.4 Auto Power-Up upon System Wakeup

User can configure Flash layer so that the controller automatically turns on Voltage Clamper and the Flash upon the system wakeup. In order to enable this feature, user shall set SEL_PWR_ON_WUP (in Register 0x12) properly. See Table 8.

Table 8: Configuration for Auto Power-Up upon System Wakeup

SEL_PWR_ON_WUP	**Description**
3'h0	Auto-power-up/down upon system wakeup is disabled
3'h1	Turn-on/off Voltage Clamper only upon system wakeup
3'h2	Reserved
3'h3	Turn-on/off both Voltage Clamper and the Flash upon system wakeup
3'h4 $\sim$ 3'h7	Reserved

If IRQ_PWR_ON_WUP = 1'h1, then Flash layer sends an MBus interrupt message once the auto power-up upon system wakeup is complete.

Table 9: Interrupt Payloads for Auto Power-Up upon System Wakeup

Interrupt Payload	Description
8'hB5	Auto Power-Up upon System Wakeup completed

7 Copy from Flash to SRAM

In order to copy data from the Flash to SRAM, make sure Voltage Clamper and the Flash are turned-on or it uses Auto Power-Up/Down (Section 6).

Then, follow instructions below:

1. Set SRAM_START_ADDR. This is the SRAM address at which the first 32-bit of the data will be stored.
2. Set FLSH_START_ADDR. This is the Flash address at which the first 32-bit of the data is stored.
3. Set LENGTH. This is the length of the data to be copied. Use 'number of words (32-bit) minus 1'
4. Set IRQ_EN. If this is set to 1'h1, it will send an MBus interrupt message once the operation is done. See Table 10 for details on the interrupt payloads.
5. Set CMD = 4'h1
6. Set GO = 1'h1

Step 3 $\sim$ 6 can be done using one MBus Register Write Message. As soon as GO becomes 1'h1, it starts the operation. At the end of the operation, GO is automatically reset to 0 by the controller (main FSM).

Table 10: Interrupt Payloads for Copy from Flash to SRAM

Interrupt Payload	Description
8'h2B	Copy from Flash to SRAM completed

Figure 7 shows internal waveforms during Flash Read operation. T_1 indicates one CLK_{main} cycle, and the other timing parameters, such as Tben, are controlled through MBus Register File.

Each SE pulse reads out 8-bit data from the Flash. Thus, it needs four SE pulses for the controller (main FSM) to get a 32-bit data. This is done internally and automatically by the controller (main FSM).

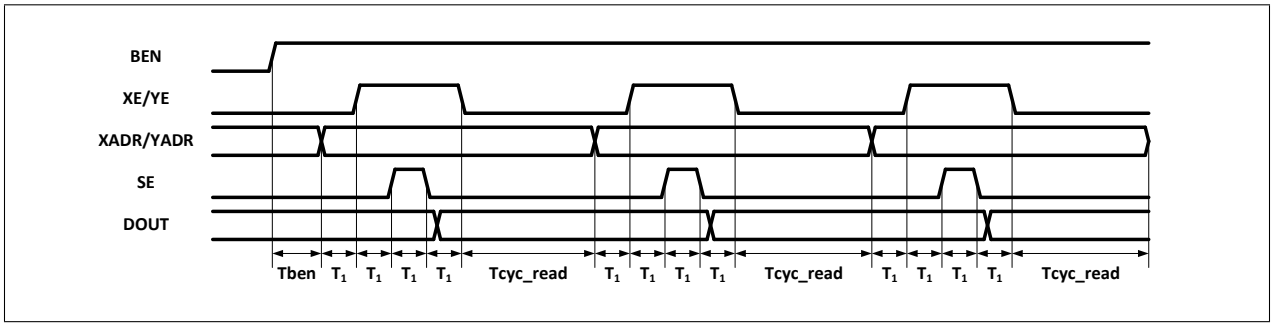

Figure 7: Flash Read Waveform

8 Copy from SRAM to Flash

Copy from SRAM to the Flash requires programming operation. Flash layer provides two-types of programming: Normal Program and Fast Program. While Normal Program is energy-efficient, Fast Program provides a better throughput at the cost of higher power consumption.

8.1 Normal Program

In order to copy data from SRAM to the Flash using Normal Program, make sure Voltage Clamper and the Flash are turned-on or it uses Auto Power-Up/Down (Section 6).

Then, follow instructions below:

1. Set SRAM_START_ADDR. This is the SRAM address at which the first 32-bit of the data is stored.
2. Set FLSH_START_ADDR. This is the Flash address at which the first 32-bit of the data will be stored.
3. Set LENGTH. This is the length of the data to be copied. Use 'number of words (32-bit) minus 1'
4. Set IRQ_EN. If this is set to 1'h1, it will send an MBus interrupt message once the operation is done. See Table 11 for details on the interrupt payloads.
5. Set CMD = 4'h2
6. Set GO = 1'h1

Step 3 $\sim$ 6 can be done using one MBus Register Write Message. As soon as GO becomes 1'h1, it starts the operation. At the end of the operation, GO is automatically reset to 0 by the controller (main FSM).

Table 11: Interrupt Payloads for Copy from SRAM to Flash (Normal Program)

Interrupt Payload	Description
8'h3F	Copy from SRAM to Flash (Normal Program) completed

Figure 8 shows internal waveforms during Flash Normal Program operation. T_1 indicates one CLKmain cycle, and the other timing parameters, such as Tben, T5us, T10us, Tprog, Tcyc_prog are controlled through MBus Register File. Each PROG pulse writes 8-bit data to the Flash. Thus, it needs four PROG pulses for the controller (main FSM) to write a 32-bit data. This is done internally and automatically by the controller (main FSM).

8.2 Fast Program

In order to copy data from SRAM to the Flash using Fast Program, make sure Voltage Clamper and the Flash are turned-on or it uses Auto Power-Up/Down (Section 6). Then, follow instructions below:

1. Set SRAM_START_ADDR. This is the SRAM address at which the first 32-bit of the data is stored.
2. Set FLSH_START_ADDR. This is the Flash address at which the first 32-bit of the data will be stored.
3. Set LENGTH. This is the length of the data to be copied. Use 'number of words (32-bit) minus 1'

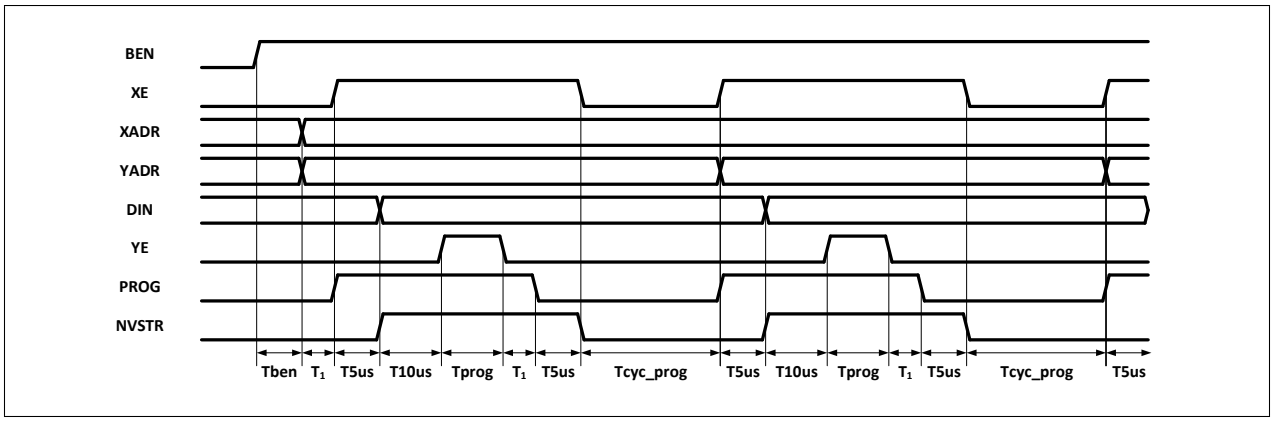

Figure 8: Flash Normal Program Waveform

4. Set IRQ_EN. If this is set to 1'h1, it will send an MBus interrupt message once the operation is done. See Table 12 for details on the interrupt payloads.
5. Set CMD = 4'h3
6. Set GO = 1'h1

Step 3 $\sim$ 6 can be done using one MBus Register Write Message. As soon as GO becomes 1'h1, it starts the operation. At the end of the operation, GO is automatically reset to 0 by the controller (main FSM).

Table 12: Interrupt Payloads for Copy from SRAM to Flash (Fast Program)

Interrupt Payload	Description
8'h5D	Copy from SRAM to Flash (Fast Program) completed

Figure 9 shows internal waveforms during Flash Fast Program operation. T_1 indicates one CLK_{main} cycle, and the other timing parameters, such as Tben, T5us, T10us, Tprog, Tcyc_prog are controlled through Register File. Each PROG pulse writes 8-bit data to the Flash. Thus, it needs four PROG pulses for the controller (main FSM) to write a 32-bit data. This is done internally and automatically by the controller (main FSM).

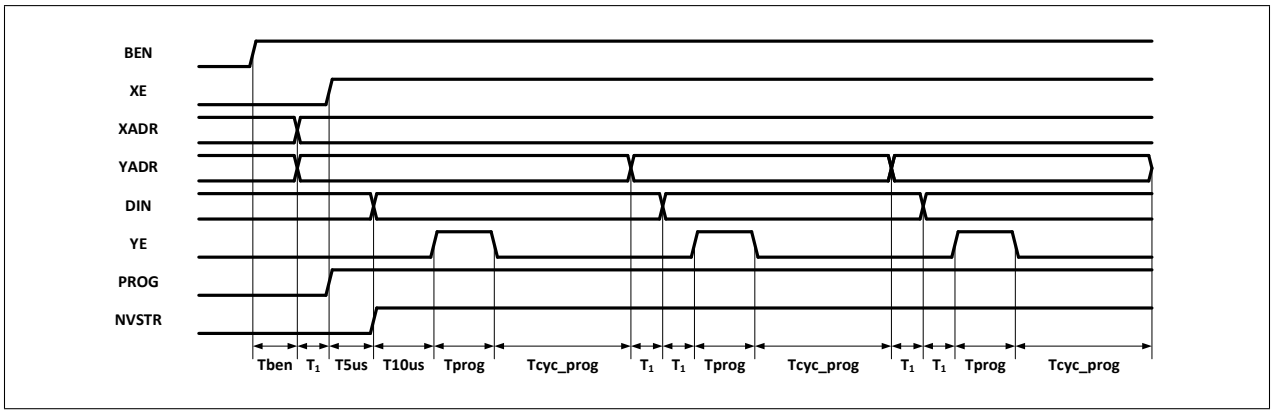

Figure 9: Flash Fast Program Waveform

9 Erase Flash

9.1 Page Erase

In order to erase a page (1kB) in the Flash, make sure Voltage Clamper and the Flash are turned-on or it uses Auto Power-Up/Down (Section 6). Then, follow instructions below:

1. Set FLSH_START_ADDR. This is the Flash address which corresponds to the first 32-bit of the page. See Figure 4 for the details on the addresses. For example, In FLPv3L, FLSH_START_ADDR shall be set to 0x3FF00 to erase Page#1023.
2. LENGTH in Register 0x07 is not used and will be ignored.
3. Set IRQ_EN. If this is set to 1'h1, it will send an MBus interrupt message once the operation is done. See Table 13 for details on the interrupt payloads.
4. Set CMD = 4'h4
5. Set GO = 1'h1

Step 3 $\sim$ 5 can be done using one MBus Register Write Message. As soon as GO becomes 1'h1, it starts the operation.

At the end of the operation, GO is automatically reset to 0 by the controller (main FSM).

Table 13: Interrupt Payloads for Flash Page Erase

Interrupt Payload	Description
8'h4F	Flash Page Erase completed

Figure 10 shows internal waveforms during Flash Erase operation. T_1 indicates one CLK_{main} cycle, and the other timing parameters, such as Tben, T5us, Terase are controlled through MBus Register File.

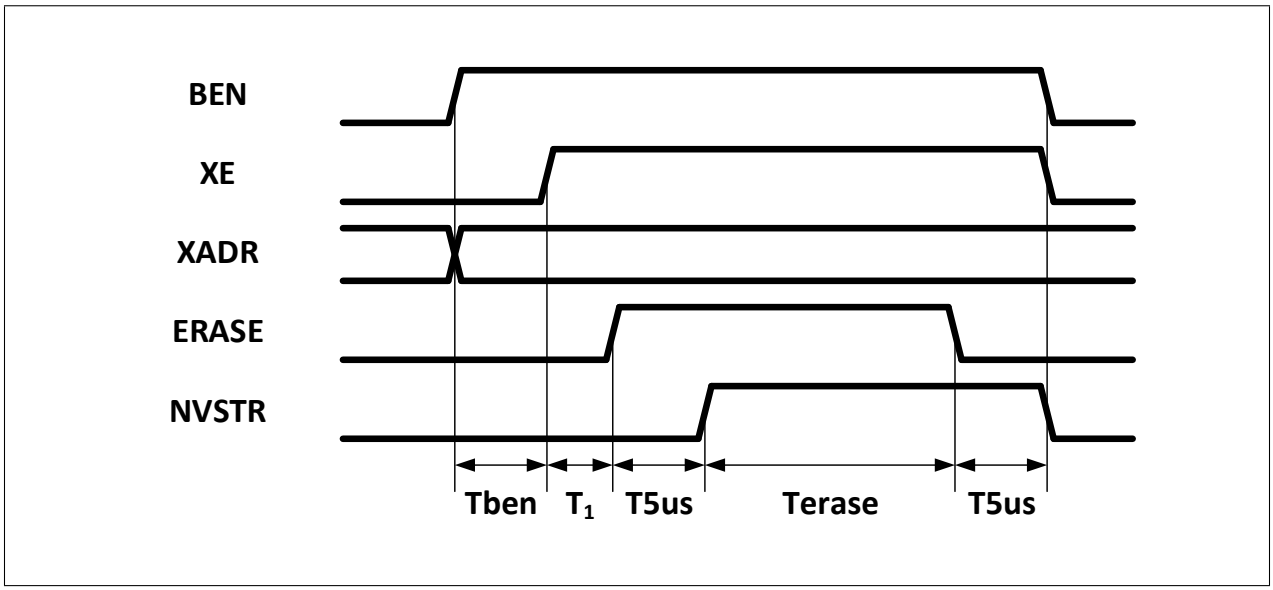

Figure 10: Flash Page Erase Waveform

9.2 Reference Array Erase

FLPv3L has sixteen (16) internal banks and each bank has its reference cell array. These reference cell arrays are used to generate a reference voltage for the sense amplifier during Read operation. In order to erase a reference cell array in the Flash, make sure Voltage Clamper and the Flash are turned-on or it uses Auto Power-Up/Down (Section 6). Then, follow instructions below:

1. Set FLSH_START_ADDR. This is the Flash address which corresponds to the first 32-bit of the bank, and should be one of the followings.
    - Bank#0: 0x00000
    - Bank#1: 0x04000
    - Bank#2: 0x08000
    - Bank#3: 0x0C000
    - Bank#4: 0x10000
    - Bank#5: 0x14000
    - Bank#6: 0x18000
    - Bank#7: 0x1C000
    - Bank#8: 0x20000
    - Bank#9: 0x24000
    - Bank#10: 0x28000
    - Bank#11: 0x2C000
    - Bank#12: 0x30000
    - Bank#13: 0x34000
    - Bank#14: 0x38000
    - Bank#15: 0x3C000
2. LENGTH in Register 0x07 is not used and will be ignored.
3. Set IRQ_EN. If this is set to 1'h1, it will send an MBus interrupt message once the operation is done. See Table 14 for details on the interrupt payloads.
4. Set CMD = 4'h7
5. Set GO = 1'h1

Step 3 $\sim$ 5 can be done using one MBus Register Write Message. As soon as GO becomes 1'h1, it starts the operation.

At the end of the operation, GO is automatically reset to 0 by the controller (main FSM).

Table 14: Interrupt Payloads for Flash Reference Array Erase

Interrupt Payload	Description
8'h6F	Flash Reference Array Erase completed

Figure 11 shows internal waveforms during Flash Erase operation. Tben and Terase are controlled through MBus Register File.

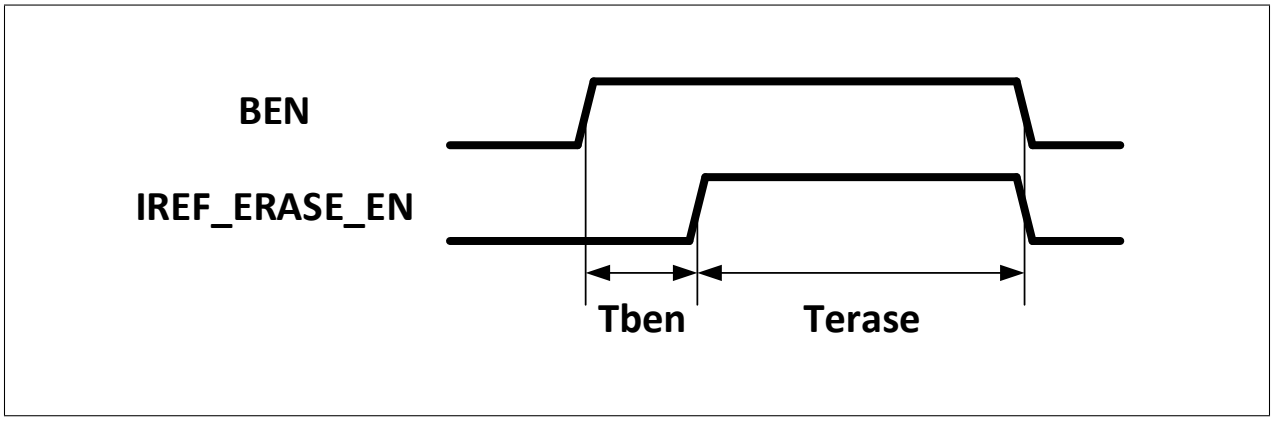

Figure 11: Flash Reference Array Erase Waveform

10 External Streaming

(Note that this section only explains about the standard (non-ping-pong) external streaming. For the external ping-pong streaming, see Section 11.2.

External streaming is a way to directly write into SRAM using CLK_EXT, DATA_EXT[1:0] pads. It does not use the Flash itself, so user does not have to turn on the Flash or Voltage Clamper. Obviously, the streamed data is only stored in SRAM. User has to use "Copy from SRAM to the Flash (Section 8)" to write into the Flash. In order to start the external streaming, follow instructions below:

1. Set Register 0x0C
   - WRAP_EXT = 1'h0
   - UPDATE_ADDR_EXT = 1'h1
   - BIT_EN_EXT: Enable/Disable DATA_EXT[1:0] pads. See Table 15. If both DATA_EXT[1] and DATA_EXT[0] are enabled, DATA_EXT[1] becomes MSB.
2. Set SRAM_START_ADDR_EXT. This is the SRAM address at which the first 32-bit will be stored.
3. Set TIMEOUT_EXT. Number of CLKmain cycles that will trigger the time-out error during External Streaming.
4. Set LENGTH. This is the length of data to be streamed. Use 'number of words (32-bit) minus 1'
5. Set IRQ_EN. If this is set to 1'h1, it will send an MBus interrupt message once the operation is done. See Table 16 for details on the interrupt payloads.
6. Set CMD = 4'h6
7. Set GO = 1'h1

Step 4 $\sim$ 7 can be done using one MBus Register Write Message. As soon as GO becomes 1'h1, it starts the operation. At the end of the operation, GO is automatically reset to 0 by the controller (main FSM).

Table 15: BIT_EN_EXT Setting

BIT_EN_EXT	DATA_EXT[1]	DATA_EXT[0]
2'h0	Disabled	Disabled
2'h1	Disabled	Enabled
2'h2	Enabled	Disabled
2'h3	Enabled	Enabled

TIMEOUT_EXT=0 will NOT trigger the time-out error, and it will be in the streaming mode indefinitely until one of the following events occurs:

- The number of words streamed becomes equal to the value specified in LENGTH (plus one).
- User manually resets the controller (main FSM) by writing 0 to FORCE_RESETN

If UPDATE_ADDR_EXT = 1'h1, SRAM_START_ADDR_EXT is automatically updated at the end of the External Streaming. The next External Streaming (triggered by another GO command) starts writing from the successive SRAM address.

Table 16: Interrupt Payloads for External Streaming

Interrupt Payload	Description
8'hF5	External Streaming Time-Out Error
8'hF7	External Streaming Too-Fast Error
8'hF9	External Streaming Successful
8'hFB	Has not yet received the number of words specified in LENGTH, but the address reaches the last SRAM Address while WRAP_EXT = 1'h0
8'hFD	External Streaming Successful. Has received the number of words specified in LENGTH, and the address reaches the last SRAM Address while WRAP_EXT = 1'h0

11 Ping-Pong Streaming

Ping-pong streaming writes incoming data into SRAM, as well as the Flash, at the same time, utilizing the two sub-SRAMs shown in Figure 3. Thus, user has to make sure Voltage Clamper and the Flash are turned-on or it uses Auto Power-Up/Down (Section 6).

Due to the nature of the ping-pong streaming, SRAM data keeps being overwritten if the amount of the streamed data exceeds the SRAM capacity.

There are two types of ping-pong streaming: MBus ping-pong streaming and External ping-pong streaming.

11.1 MBus Ping-Pong Streaming

MBus ping-pong streaming writes data from MBus Memory Streaming messages into SRAM and the Flash. In order to do this, follow instructions below:

1. Set PP_STR_LIMIT. This is the limit on the number of words to be ping-pong streamed. The unit is 'word (32-bit)'. Once the number of streamed data reaches PP_STR_LIMIT, any other following data will be ignored and not be written into SRAM or the Flash.
2. Set PP_STR_EN = 1'h1
3. Set PP_FLSH_ADDR. This is the Flash address at which the first 32-bit data will be written. It must be a start address of a page.
4. Set IRQ_EN. If this is set to 1'h1, it will send an MBus interrupt message once the operation is done. See Table 17 for details on the interrupt payloads.
5. (Optional) Set PP_NO_ERR_DETECTION = 1'h1. This will not detect Buffer Overrun error. The recommended value is 1'h0 (detect Buffer Overrun error).
6. (Optional) Set PP_USE_FAST_PROG = 1'h1. This will use Fast Program shown in Section 8.2. Otherwise, it will use Normal Program shown in Section 8.1.
7. (Optional) Set PP_WRAP = 1'h1. This will enable address wrapping in the Flash (i.e, the Flash address becomes 0 after writing into the last address of the Flash). If PP_WRAP = 1'h0, it will stop writing into the Flash once it writes into the last address of the Flash, and any other following data will be ignored and not be written into the Flash.

With this configuration, it will start the ping-pong operation as soon as there is an incoming MBus Memory Streaming message.

The ping-pong operation stops when one of the following event occurs.

- The MBus Memory Streaming message ends, and the amount of data streamed is less than that specified in PP_STR_LIMIT. In this case, there is no MBus interrupt message even if IRQ_EN = 1'h1.
- The MBus Memory Streaming message ends, and the amount of data streamed is equal to or greater than that specified in PP_STR_LIMIT.
- If there is any error. The interrupt payload should indicate the type of the error.

Once the ping-pong operation is done, it will update the following registers in MBus Register File.

- PP_FLAG_END_OF_FLASH: Updated to 1 if the Flash reaches its last address during ping-pong streaming.
- PP_FLAG_STR_LIMIT: Updated to 1 if PP_LENGTH_STREAMED becomes equal to PP_STR_LIMIT.
- PP_FLAG_COPY_LIMIT: Updated to 1 if PP_LENGTH_COPIED becomes equal to PP_STR_LIMIT.
- PP_LENGTH_STREAMED: Updated to the number of words streamed so far during ping-pong streaming (the number of words written into SRAM).
- PP_LENGTH_COPIED: Updated to the number of words copied so far during ping-pong streaming (the number of words copied from SRAM to the Flash).

Table 17: Interrupt Payloads for MBus Ping-Pong Streaming

Interrupt Payload	Description
8'hE2	Ping-Pong streaming successful and it reaches its given limit in PP_STR_LIMIT
8'hE5	Buffer Overrun Error

11.2 External Ping-Pong Streaming

External ping-pong streaming writes data from CLK_EXT, DATA_EXT[1:0] pads into SRAM and the Flash. In order to do this, follow instructions below:

1. Set PP_STR_LIMIT. This is the limit on the number of words to be ping-pong streamed. The unit is 'word (32-bit)'. Once the number of streamed data reaches PP_STR_LIMIT, any other following data will be ignored and not be written into SRAM or the Flash.
2. Set PP_STR_EN = 1'h1
3. Set PP_FLSH_ADDR. This is the Flash address at which the first 32-bit data will be written. It must be a start address of a page.
4. PP_BIT_EN_EXT: Enable/Disable DATA_EXT[1:0] pads. See Table 18. If both DATA_EXT[1] and DATA_EXT[0] are enabled, DATA_EXT[1] becomes MSB.
5. (Optional) Set PP_NO_ERR_DETECTION = 1'h1. This will not detect Buffer Overrun error. The recommended value is 1'h0 (detect Buffer Overrun error).
6. (Optional) Set PP_USE_FAST_PROG = 1'h1. This will use Fast Program shown in Section 8.2. Otherwise, it will use Normal Program shown in Section 8.1.
7. (Optional) Set PP_WRAP = 1'h1. This will enable address wrapping in the Flash (i.e, the Flash address becomes 0 after writing into the last address of the Flash). If PP_WRAP = 1'h0, it will stop writing into the Flash once it writes into the last address of the Flash, and any other following data will be ignored and not be written into the Flash.
8. LENGTH is not used and will be ignored.
9. Set IRQ_EN. If this is set to 1'h1, it will send an MBus interrupt message once the operation is done. See Table 19 for details on the interrupt payloads.
10. Set CMD = 4'h6
11. Set GO = 1'h1

Step 8 $\sim$ 11 can be done using one MBus Register Write Message. As soon as GO becomes 1h1, it starts the ping-pong operation, and any following External Streaming will be written into SRAM as well as the Flash.

The ping-pong operation stops when one of the following event occurs.

- The amount of data streamed becomes equal to that specified in PP_STR_LIMIT.
- User writes 0 to PP_STR_EN. This will immediately terminate the ping-pong streaming. PP_STR_EN will be set to 1 automatically by the controller (main FSM).
- If there is any error. The interrupt payload should indicate the type of the error.

At the end of the operation, GO is automatically reset to 0 by the controller (main FSM).

Table 18: PP_BIT_EN_EXT Setting

PP_BIT_EN_EXT	DATA_EXT[1]	DATA_EXT[0]
2'h0	Disabled	Disabled
2'h1	Disabled	Enabled
2'h2	Enabled	Disabled
2'h3	Enabled	Enabled

Table 19: Interrupt Payloads for External Ping-Pong Streaming

Interrupt Payload	Description
8'hE2	Ping-Pong streaming successful and it reaches its given limit in PP_STR_LIMIT
8'hE4	External Ping-Pong Streaming stopped by user
8'hE5	Buffer Overrun Error
8'hE6	External Streaming Too-Fast during External Ping-Pong Streaming

Once the ping-pong operation is done, it will update the following registers in Register File.

- PP_FLAG_END_OF_FLASH: Updated to 1 if the Flash reaches its last address during ping-pong streaming.
- PP_FLAG_STR_LIMIT: Updated to 1 if PP_LENGTH_STREAMED becomes equal to PP_STR_LIMIT.
- PP_FLAG_COPY_LIMIT: Updated to 1 if PP_LENGTH_COPIED becomes equal to PP_STR_LIMIT.
- PP_LENGTH_STREAMED: Updated to the number of words streamed so far during ping-pong streaming (the number of words written into SRAM).
- PP_LENGTH_COPIED: Updated to the number of words copied so far during ping-pong streaming (the number of words copied from SRAM to the Flash).

12 Boot-Up

12.1 Boot-Up Operation

12.1.1 Auto Boot-Up

Auto Boot-Up can be triggered only when the pad BOOT_DIS is floating or grounded. Auto Boot-Up is disabled if the pad BOOT_DIS is held high at 1.2V.

Auto Boot-Up automatically follows the below sequence.

1. Trigger the MBus interrupt controller (mbus_int_ctrl) upon POR to wake up the whole system.
2. Check BOOT_REG_PATTERN value. If it is 2'h3, it stops the boot-up sequence, and the controller (main FSM) goes into IDLE state. If it is not 2'h3, it proceeds to the next step.
3. Set BOOT_REG_PATTERN=2'h3.
4. Turn on Voltage Clamper and the Flash following the sequence described in Section 6.
5. Read out from the very first word (ADDR#0) in the Flash.
    - If the read-out data is matched with Header (=0x6AB0C3CB), proceed to the next step.
    - If the read-out data is NOT matched with Header (=0x6AB0C3CB), and if COMP_CTRL_I_1STG is less than 4'hE, then turn off Voltage Clamper and the Flash, increment COMP_CTRL_I_1STG by 1, then re-start from 4.
    - If the read-out data is NOT matched with Header (=0x6AB0C3CB), and if COMP_CTRL_I_1STG is 4'hE or larger, then set BOOT_FLAG_WRONG_HEADER=1, power-off Voltage Clamper and the Flash (Section 6), send out an MBus flag message, followed by the MBus sleep message.
6. Read out from the next address from the flash memory, and executes the code following the Boot-Up ISA (See Section 12.2). Repeat this until one of the following conditions is met.
    - If the read-out data is TAIL_IDLE, it sets BOOT_FLAG_SUCCESS=1, and the controller (main FSM) will go into IDLE state.
    - If the read-out data is TAIL_PWDN, it sets BOOT_FLAG_SUCCESS=1 and BOOT_FLAG_PWDN=1. It turns-off Voltage Clamper and the Flash, and the controller (main FSM) will go into IDLE state.
    - If the read-out data is TAIL_SLEEP, it sets BOOT_FLAG_SUCCESS=1 and BOOT_FLAG_SLEEP=1. It turns-off Voltage Clamper the Flash and then sends out the MBus sleep message.
    - If there is a checksum error, it sets BOOT_FLAG_CHKSUM_ERROR=1. It turns-off Voltage Clamper and the Flash and then sends out an MBus flag message, followed by the MBus sleep message.
    - If the read-out data is an invalid command (after **ECC** correction, if any), it sets BOOT_FLAG_INVALID_CMND=1. It turns-off Voltage Clamper and the Flash and then sends out an MBus flag message, followed by the MBus sleep message.
    - If the read-out data is has more than 2-bit error, hence not correctable using the **ECC**, it sets BOOT_FLAG_ECC_ERROR=1. It turns-off Voltage Clamper and the Flash and then sends out an MBus flag message, followed by the MBus sleep message.

12.1.2 Manual Boot-Up

Manual Boot-Up can be triggered regardless of the pad BOOT_DIS setting. In order to start the Manual Boot-Up, use the GO operation (CMD=4'h8). It will follow the sequence described in Section 12.1.1 with the following exceptions.

- It does not trigger the MBus interrupt controller. The assumption is that the whole system is already up and running.
- It does not check the BOOT_REG_PATTERN value, so the boot-up sequence is always executed regardless of the BOOT_REG_PATTERN value. However, if BOOT_REG_PATTERN is not 2'h3, then it will set BOOT_REG_PATTERN=2'h3.

12.2 Boot-Up ISA (Instruction Set Architecture)

This section describes how the Flash memory contents (i.e., boot-up code) are to be structured. Every header/command/tail/data described here is 32-bit. Since the flash memory in Flash layer has 8-bit I/O, the following conversions are made in the controller (main FSM).

- DATA[0][31:0] = {FLASH_DATA[3][7:0], FLASH_DATA[2][7:0], FLASH_DATA[1][7:0], FLASH_DATA[0][7:0]}
- DATA[1][31:0] = {FLASH_DATA[7][7:0], FLASH_DATA[6][7:0], FLASH_DATA[5][7:0], FLASH_DATA[4][7:0]}
- ...
- DATA[N][31:0] = {FLASH_DATA[4N+3][7:0], FLASH_DATA[4N+2][7:0], FLASH_DATA[4N+1][7:0], FLASH_DATA[4N][7:0]}

A boot-code has the following requirements.

- DATA[0][31:0] must be Header (=0x6AB0C3CB)
- DATA[1][31:0] must be either one of the Commands or one of the Tails.
- A valid Command or Tail must follow the Command structure described in Section 12.2.2 or Section 12.2.3.
- A boot-up code must be ended with one of the Tails.

If any of these requirements are not met, one of the error handling described in Section 12.2.4 is initiated.

12.2.1 Header

DATA[0] must be Header (=0x6AB0C3CB).

12.2.2 Commands

REG_WRITE

Syntax

ADDR+0 : {4'h1, 4'h1, **N**[17:11], **ECC**[5], **N**[10:4], **ECC**[4], **N**[3:1], **ECC**[3], **N**[0], **ECC**[2:0]}

ADDR+1 : {28'h0, **DEST_SHORT_PREFIX_0**[3:0]}

ADDR+2 : {**REG_IDX_0**[7:0], **REG_DATA_0**[23:0]}

ADDR+3 : **REG_CHKSUM_0**[31:0]

ADDR+4 : {28'h0, **DEST_SHORT_PREFIX_1**[3:0]}

ADDR+5 : {**REG_IDX_1**[7:0], **REG_DATA_1**[23:0]}

ADDR+6 : **REG_CHKSUM_1**[31:0]

...

ADDR+3N+1 : {28'h0, **DEST_SHORT_PREFIX_N**[3:0]}

ADDR+3N+2 : {**REG_IDX_N**[7:0], **REG_DATA_N**[23:0]}

ADDR+3N+3 : **REG_CHKSUM_N**[31:0]

Description

- This command writes **REG_DATA_n** into **REG_IDX_n** register ($0 \leq n \leq$ **N**) on the layer whose short-prefix is **DEST_SHORT_ADDR_n**.
- If **DEST_SHORT_PREFIX_n**=4'h0, it will write into the register on this layer (Flash layer). Otherwise, it will write into the register on the layer whose short-prefix is **DEST_SHORT_PREFIX_n**.
- **REG_CHKSUM_n**[31:0] is the Checksum defined below:
  REG_CHKSUM_n[31:0]={28'h0, **DEST_SHORT_PREFIX_n**[3:0]} + {**REG_IDX_n**[7:0], **REG_DATA_n**[23:0]}
- **N**[19:0] cannot be 20'hFFFFF.
- For **ECC**[5:0], see Section 12.2.6 for details.

MEM_COPY

Syntax

ADDR+0 : {4'h1, 4'h2, **N**[17:11], **ECC**[5], **N**[10:4], **ECC**[4], **N**[3:1], **ECC**[3], **N**[0], **ECC**[2:0]}

ADDR+1 : {28'h0, **DEST_SHORT_PREFIX**[3:0]}

ADDR+2 : **DEST_MEM_START_ADDR**[31:0]

ADDR+3 : **DATA_0**[31:0]

ADDR+4 : **DATA_1**[31:0]

ADDR+5 : **DATA_2**[31:0]

...

ADDR+N+3 : **DATA_N**[31:0]

ADDR+N+4 : **MEM_CHKSUM**[31:0]

Description

- This command will copy **DATA_0** **DATA_N** into the SRAM in the layer whose short-prefix is **DEST_SHORT_PREFIX**
- The start prefix of the destination SRAM is given by **DEST_MEM_START_ADDR**
- **DEST_SHORT_PREFIX** cannot be 0x0 or 0xF.
- "**DEST_SHORT_PREFIX** = 'Flash layer-short-prefix'" is not dened.
- For word-alignment, it is required that **DEST_MEM_START_ADDR**[1:0] = 2b00
- **MEM_CHKSUM**[31:0] is the Checksum defined below:
  MEM_CHKSUM[31:0]={28'h0, **DEST_SHORT_ADDR**[3:0]} + **DEST_MEM_START_ADDR**[31:0] + **DATA_0**[31:0] + **DATA_1**[31:0] + ... + **DATA_N**[31:0]
- **N**[19:0] cannot be 20'hFFFFF.
- For **ECC**[5:0], see Section 12.2.6 for details.

ENUMERATION

Syntax [ADDR+0]: {4'h1, 4'hE, 7'h0, **ECC**[5], 7'h0, **ECC**[4], **SHORT_PREFIX**[3:1], **ECC**[3], **SHORT_PREFIX**[0], **ECC**[2:0]}

Description

- This command will send out an MBus enumeration message, trying to set the next layer's short-prefix to **SHORT_PREFIX**.
- **SHORT_PREFIX**[3:0] cannot be 0x0 or 0x1 or 0xF.
- It cannot enumerate Flash layer itself due to a limitation of the MBus implementation.
- It is recommended that the user use NOP command right after this ENUMERATION command, to provide some time for the MBus transaction.
- For **ECC**[5:0], see Section 12.2.6 for details.

NOP

Syntax [ADDR+0]: {4'h1, 4'hD, **N**[17:11], **ECC**[5], **N**[10:4], **ECC**[4], **N**[3:1], **ECC**[3], **N**[0], **ECC**[2:0]}

Description

- This command will make the controller (main FSM) stay idle for (**N**+1) clock cycles.
- **N**[19:0] cannot be 20'hFFFFF.
- For **ECC**[5:0], see Section 12.2.6 for details.

12.2.3 Tails

TAIL_IDLE

Syntax [ADDR+0]: {4'hF, 4'h0, 7'h0, **ECC**[5], 7'h0, **ECC**[4], 3'h0, **ECC**[3], 1'h0, **ECC**[2:0]}

Description

- This command will make the controller (main FSM) go into the IDLE state and stay there.
- Everything stays on, including Voltage Clamper and the Flash.
- For **ECC**[5:0], see Section 12.2.6 for details.

TAIL_PWDN

Syntax [ADDR+0]: {4'hF, 4'hF, 7'h0, **ECC**[5], 7'h0, **ECC**[4], 3'h0, **ECC**[3], 1'h0, **ECC**[2:0]}

Description

- This command will turn-off Voltage Clamper and the Flash, and then make the controller (main FSM) go into the IDLE state and stay there.
- Everything except Voltage Clamper and the Flash memory will stay on. For example, the controller (main FSM) and the Layer Ctrl stay on.
- For **ECC**[5:0], see Section 12.2.6 for details.

TAIL_SLEEP

Syntax [ADDR+0]: {4'hF, 4'hC, 7'h0, **ECC**[5], 7'h0, **ECC**[4], 3'h0, **ECC**[3], 1'h0, **ECC**[2:0]}

Description

- This command will turn-off Voltage Clamper and the Flash and then send out the MBus sleep message.
- For **ECC**[5:0], see Section 12.2.6 for details.

12.2.4 Error Handling

If there is an error, it will do one of the followings depending on the error type.

- If the Header is wrong, it sets BOOT_FLAG_WRONG_HEADER=1, turns-off Voltage Clamper and the Flash, and then sends out an MBus flag message, followed by the MBus sleep message.
- If there is a checksum error, it sets BOOT_FLAG_CHKSUM_ERROR=1. It turns-off Voltage Clamper and the Flash and then sends out an MBus flag message, followed by the MBus sleep message.
- If the read-out data is an invalid command (after ECC correction, if any), it sets BOOT_FLAG_INVALID_CMND=1. It turns-off Voltage Clamper and the Flash and then sends out an MBus flag message, followed by the MBus sleep message.
- If the read-out data has more than 2-bit error, hence not correctable using the ECC, it sets BOOT_FLAG_ECC_ERROR=1. It turns-off the ash and then sends out an MBus flag message, followed by the MBus sleep message.

Note that, the 'invalid command error' and the 'ECC error' may be indistinguishable in some cases.

The MBus flag message has the following contents:

MBus Addr: 8'h10
MBus Data: {8'h07, Reg0x10[23:0]}

12.2.5 Flag Registers

Upon the end of the booting program, the boot-flag register (Register 0x10) values will be updated. These registers are always-on, so they are reset by POR reset. These register values can be changed using Register Write functionality of the Layer Ctrl.

- CASE 1: Successful Boot-Up ending with TAIL_IDLE
  - BOOT_FLAG_SUCCESS=1
- CASE 2: Successful Boot-Up ending with TAIL_PWDN
  - BOOT_FLAG_SUCCESS=1
  - BOOT_FLAG_PWDN=1
- CASE 3: Successful Boot-Up ending with TAIL_SLEEP
  - BOOT_FLAG_SUCCESS=1

  - BOOT_FLAG_SLEEP=1
- CASE 4: Checksum Error
  - BOOT_FLAG_CHKSUM_ERROR=1
- CASE 5: Invalid Command/Tail (Wrong Syntax)
  - BOOT_FLAG_INVALID_CMND=1
- CASE 6: Wrong Header
  - BOOT_FLAG_WRONG_HEADER=1
- CASE 7: **ECC** Error (More than 1-bit Error)
  - BOOT_FLAG_ECC_ERROR=1

Note that, the 'invalid command error (CASE 5)' and the '**ECC** error (CASE 7)' may be indistinguishable in some cases.

12.2.6 ECC

All Commands and Tails are protected with **ECC**. **ECC** used here is Hamming (32, 26) and it is SEC-DED.

- **ECC**[0] = $\wedge$Bit[31:1]
- **ECC**[1] = Bit[3] $\wedge$ Bit[5] $\wedge$ Bit[7] $\wedge$ Bit[9] $\wedge$ Bit[11] $\wedge$ Bit[13] $\wedge$ Bit[15] $\wedge$ Bit[17] $\wedge$ Bit[19] $\wedge$ Bit[21] $\wedge$ Bit[23] $\wedge$ Bit[25] $\wedge$ Bit[27] $\wedge$ Bit[29] $\wedge$ Bit[31]
- **ECC**[2] = Bit[3] $\wedge$ Bit[6] $\wedge$ Bit[7] $\wedge$ Bit[10] $\wedge$ Bit[11] $\wedge$ Bit[14] $\wedge$ Bit[15] $\wedge$ Bit[18] $\wedge$ Bit[19] $\wedge$ Bit[22] $\wedge$ Bit[23] $\wedge$ Bit[26] $\wedge$ Bit[27] $\wedge$ Bit[30] $\wedge$ Bit[31]
- **ECC**[3] = Bit[5] $\wedge$ Bit[6] $\wedge$ Bit[7] $\wedge$ Bit[12] $\wedge$ Bit[13] $\wedge$ Bit[14] $\wedge$ Bit[15] $\wedge$ Bit[20] $\wedge$ Bit[21] $\wedge$ Bit[22] $\wedge$ Bit[23] $\wedge$ Bit[28] $\wedge$ Bit[29] $\wedge$ Bit[30] $\wedge$ Bit[31]
- **ECC**[4] = Bit[9] $\wedge$ Bit[10] $\wedge$ Bit[11] $\wedge$ Bit[12] $\wedge$ Bit[13] $\wedge$ Bit[14] $\wedge$ Bit[15] $\wedge$ Bit[24] $\wedge$ Bit[25] $\wedge$ Bit[26] $\wedge$ Bit[27] $\wedge$ Bit[28] $\wedge$ Bit[29] $\wedge$ Bit[30] $\wedge$ Bit[31]
- **ECC**[5] = Bit[17] $\wedge$ Bit[18] $\wedge$ Bit[19] $\wedge$ Bit[20] $\wedge$ Bit[21] $\wedge$ Bit[22] $\wedge$ Bit[23] $\wedge$ Bit[24] $\wedge$ Bit[25] $\wedge$ Bit[26] $\wedge$ Bit[27] $\wedge$ Bit[28] $\wedge$ Bit[29] $\wedge$ Bit[30] $\wedge$ Bit[31]

12.2.7 Compiler and Examples

In order to facilitate the programming, a simple compiler script has been made. It is located in `m3_hdk/scripts/compileBootFLX` and the file name is `compileBootFLX`. A sample program is provided in the same directory, and the file name is `prog`. Details about how-to-use can be found in `prog` and `compileBootFLX` files.

13 Clock Generator

This is a clock generator that generates two clocks: CLK_{main} and CLK_{comp}. CLK_{main} is used in the Layer Ctrl and the controller (main FSM). CLK_{comp} is used in the voltage clamper. In order to provide robust operation, CLK_{comp} must be faster than CLK_{main}. More faster CLK_{comp} (compared to CLK_{main}) provides more current regulation (hence more robust flash operation).

CLK_{main} and CLK_{comp} share the same clock ring. CLK_{main} can be further slowed down using the dividers, but there is no divider for CLK_{comp}.

Both CLK_{main} and CLK_{comp} are at 1.2V, and power-gated with the controller (main FSM) and the Layer Ctrl.

13.1 Power Domains

VDD_0P6 Power-Gated Domain: Used to operate the clock ring and dividers.

VDD_1P2 Power-Gated Domain: Used to produce the clock outputs.

13.2 Operation and Tuning

It generates CLK_{main} whenever the system is active (Layer Ctrl is running). There is no need to manually turn on and off the clock generator.

It generates CLK_{comp} whenever the voltage clamper turns on. This is automatically handled by the controller (main FSM) depending on the flash operation. There is no need to manually turn on and off the clock generator.

The clock frequencies can be tuned using CLK_RING_SEL and CLK_DIV_SEL. Table 20 shows simulated clock frequencies depending on those tuning bits. Default values are shown in red & italic.

Table 20: Clock Generator Frequencies

CLK_RING_SEL	Power (μW)	CLK_{comp} (MHz)	CLK_{main} (MHz)			
			CLK_DIV_SEL =2'h0	CLK_DIV_SEL =***2'h1***	CLK_DIV_SEL =2'h2	CLK_DIV_SEL =2'h3
4'h0	6.999	32.90	8.22	4.11	2.06	1.03
4'h1	6.575	29.60	7.39	3.70	1.85	0.92
4'h2	6.158	26.70	6.67	3.34	1.67	0.83
4'h3	5.902	24.40	6.11	3.06	1.53	0.76
4'h4	5.332	23.00	5.74	2.87	1.44	0.72
4'h5	5.229	21.30	5.33	2.67	1.33	0.67
4'h6	4.964	19.80	4.94	2.47	1.24	0.62
4'h7	4.853	18.50	4.63	2.32	1.16	0.58
4'h8	5.017	22.00	5.50	2.75	1.38	0.69
4'h9	4.300	17.50	4.37	2.19	1.09	0.55
4'hA	3.517	12.90	3.22	1.61	0.81	0.40
4'hB	3.238	11.20	2.80	1.40	0.70	0.35
4'hC	***2.786***	***9.36***	2.34	***1.17***	0.59	0.29
4'hD	1.986	5.25	1.31	0.66	0.33	0.16
4'hE	1.815	4.32	1.08	0.54	0.27	0.14
4'hF	1.601	3.17	0.79	0.40	0.20	0.10

14 Clock Frequency Measurement

Flash layer provides a way to indirectly measure the frequency of CLK_{main}. It is not an accurate method to measure the clock frequency but should be good enough to get a reasonable estimation.

In order to measure CLK_{main} frequency, follow instructions below.

1. Set FLASH_AUTO_OFF = 0.
2. Write data shown below in Register 0x07. This can be done using one MBus Register Write Message.
   LENGTH = 0 (not used)
   IRQ_EN = 1'h1
   CMD = 4'h9
   GO = 1'h1
3. Wait until Flash layer sends out an MBus interrupt message. The interrupt payloads and their meanings are shown in Table 21.

User shall measure the time starting at the end of the MBus Register Write message (step 2) until the start of the MBus interrupt message.

T_D = Time@(End of MBus Reg Write Msg in step 2) - Time@(Start of MBus Interrupt Msg in step 3)

And the CLK_{main} period and frequency can be estimated as shown below.

$$\text{Period} = T_D / 1048576$$

$$\text{Frequency} = 1 / \text{Period} = 1048576 / T_D$$

Table 21: Interrupt Payloads for Clock Frequency Measurement

Interrupt Payload	Description
8'h08	Clock Frequency Measurement Completed

15 List of Interrupt Payloads

Table 22 shows a list of interrupt payloads and their meanings.

Table 22: List of Interrupt Payloads

Interrupt Payload	Description	Related Sections
8'h08	Clock Frequency Measurement Completed	Section 14
8'h2B	Copy from Flash to SRAM completed	Section 7
8'h3F	Copy from SRAM to the Flash (Normal Program) completed	Section 8.1
8'h4F	Flash Page Erase completed	Section 9.1
8'h5D	Copy from SRAM to the Flash (Fast Program) completed	Section 8.2
8'h6F	Flash Reference Array Erase completed	Section 9.2
8'hB5	Power-Up sequence completed	Section 6.2 Section 6.4
8'hBB	Power-Down sequence completed	Section 6.2
8'hE2	Ping-Pong streaming successful and it reaches its given limit in PP_STR_LIMIT	Section 11.1 Section 11.2
8'hE4	External Ping-Pong Streaming stopped by user	Section 11.2
8'hE5	Buffer Overrun Error during Ping-Pong Streaming	Section 11.1 Section 11.2
8'hE6	External Streaming Too-Fast during Ping-Pong Streaming	Section 11.2
8'hF5	External Streaming Time-Out Error	Section 10
8'hF7	External Streaming Too-Fast Error	Section 10
8'hF9	External Streaming Successful	Section 10
8'hFB	During External Streaming, it has not yet received the number of words specified in LENGTH, but the address reaches the last SRAM Address while WRAP_EXT = 1'h0	Section 10
8'hFD	External Streaming Successful. It has received the number of words specified in LENGTH, and the address reaches the last SRAM Address while WRAP_EXT = 1'h0	Section 10

16 Voltage Clamper

16.1 Description

The voltage clamper charges an output capacitor connected to VDD_2P5_FLS (also VCAP pad) with limited current. The capacitor is charged by the current determined by current reference and two current mirrors (NMOS and PMOS).

Charging current = Current multiplication of PMOS mirror $\times$ Current multiplication of NMOS mirror $\times$ VREF / 1.93MΩ

As VCAP becomes higher than a threshold, the switch between to the PMOS mirror and VCAP is turned off, and the output capacitor will not be charged any more. VCAP is divided down to VDIV by a diode stack and CTRL_VOUT<2:0> (=COMP_CTRL_VOUT), and VDIV is compared to VREF by a comparator to control the switch between the PMOS mirror and VCAP.

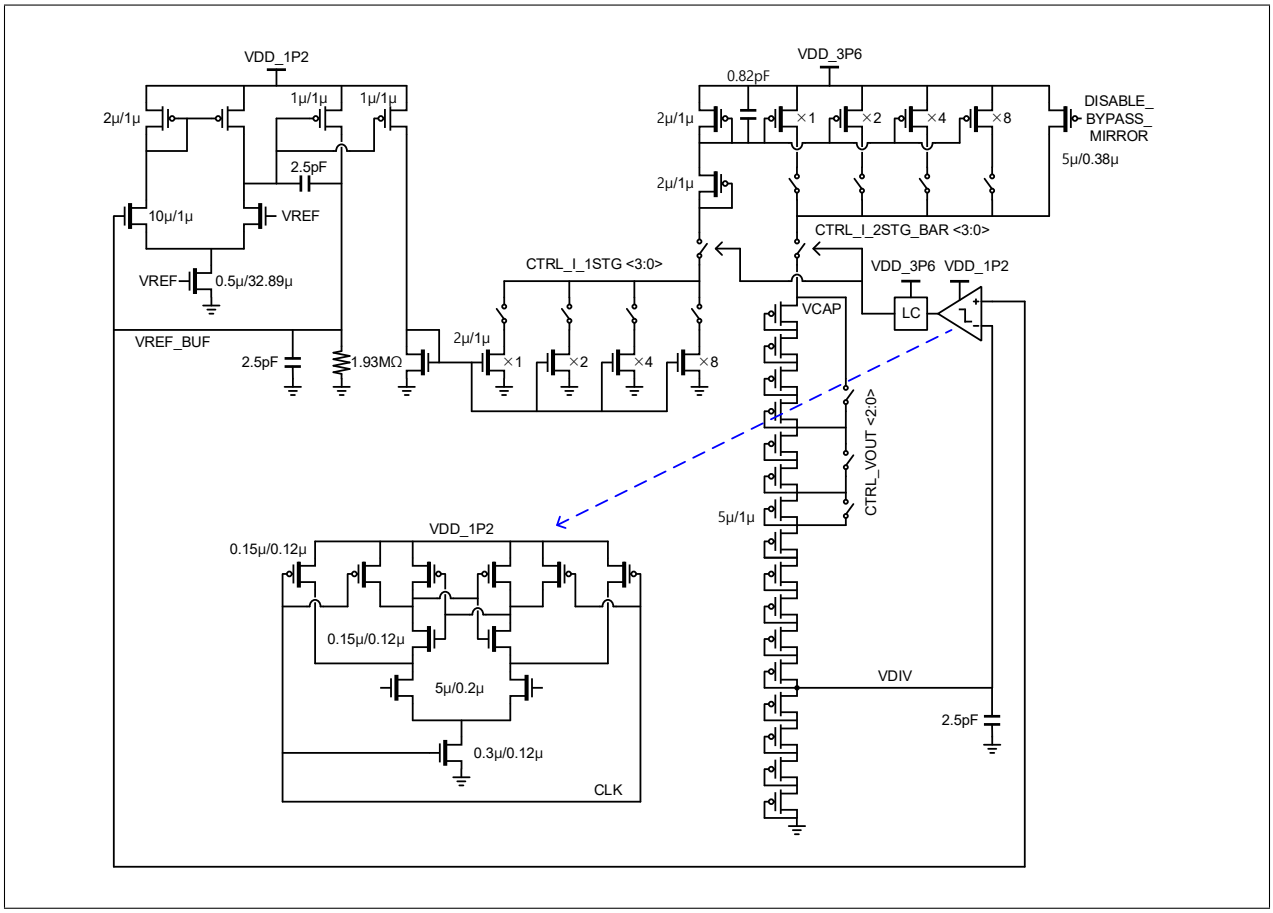

Figure 12: Schematic of Voltage Clamper

16.2 Simulation Results

16.2.1 Voltage Clamp

Phase 1: 3μA @ 25μs
Phase 2: 50μA @ 5μs

On-Chip Cap: 3nF

Comparator Clock: 2MHz (10 comparison @ Phase 2)

Temp=27℃ & VDD_1P2=1.2V & VDD_3P6=3.8V

VREF = 860mV

Table 23: Simulation Results of Voltage Clamp

Corner	TT	FF	SS	FS	SF
Max. V_{CAP} (V)	2.57	2.57	2.58	2.58	2.57
Min. V_{CAP} (V)	2.50	2.51	2.50	2.50	2.50
Ripple V_{CAP} (V)	68.7m	60.1m	73.8m	74.3m	64.1m
Max. V_{MIRROR} (V)	3.25	3.30	3.19	3.22	3.28
Min. V_{MIRROR} (V)	2.98	3.00	2.96	2.95	3.02
Max. I_{VDD_3P6} (A)	17.6μ	22.8μ	14.3μ	17.7μ	17.5μ
Power Overhead (W)	5.08μ	6.46μ	4.39μ	4.87μ	5.74μ
Power Overhead (%)	10.7	13.6	9.25	10.3	12.1

16.2.2 Voltage Divider (w/ nwell-to-psub dio)

Temp=27℃ & VCAP=2.6V

Table 24: Simulation Results of Voltage Divider

Corner	TT	FF	SS	FS	SF
Error (V)	μ=0.294m & σ=2.92m				
Error (%)	μ=0.034 & σ=0.337				
Current (A)	187n	412n	82.2n	98.2n	353n

16.2.3 Current Generator

Temp=27℃ & VDD_1P2=1.2V & VDD_3P6=3.8V

Table 25: Simulation Results of Current Generator

Corner	TT	FF	SS	FS	SF
Min. I_{UP} (A)	549n	707n	447n	551n	547n
Min. I_{UP} (A)	μ=569n & σ=100n				
Max. I_{UP} (A)	117μ	150μ	94.9μ	117μ	116μ
Max. I_{UP} (A)	μ=120μ & σ=15.9μ				

16.2.4 Current Reference

Temp=27℃ & VDD_1P2=1.2V

Table 26: Simulation Results of Current Reference

Corner	TT	FF	SS	FS	SF
DC Gain (dB)	46.7	45.6	47.8	47.6	45.8
Unity-Gain Bandwidth (Hz)	68.5k	95.9k	49.2k	81.3k	57.2k
Phase Margin (°)	74.2	73.5	75.5	71.0	77.1
Voltage Error (V)	-6.14m	-6.70m	-5.84m	-5.39m	-6.99m
Voltage Error (V)	μ=-6.22m & σ=5.89m				
Regulated Current (A)	442n	560n	366n	442n	442n
Power (W)	530n	672n	440n	531n	530n

16.2.5 Comparator

Temp=27℃ & VDD_1P2=1.2V

Table 27: Simulation Results of Comparator

Corner	TT	FF	SS	FS	SF
Input Offset Voltage (V)	μ=545μ & σ=6.48m				
Power @ 1MHz (W)	77.9n	171n	41.4n	81.9n	78.5n
Power @ 10MHz (W)	331n	400n	307n	340n	328n