!" #

$ %%% % & & & $'

(( ) $ %%% %
& & & $

- ,	. . /
$ -	! 0 1 $ *1 (
$, #	( ) 2 , $
,	3% 4#
$	2 5$. ! , $, - 6! 0 6, ! 0 6, 2
-	2 . 0 . ! 7$. -8 0 -9 9 .
: 8	; (
- , #	<&) / 2 =<&) / * %>
-	?7! , 4
-08 , #	
0! , #	&) %/ * %
@ , =@ @ >	& %<@A (%@
. $	! . %&) B. !) * &)
8	
8	3CD$ A %<D$ = ! >
	,
- $	&) C @?2 + !
, . -	&) C 2 + ! =E* E>
- 607	F * ( ) ; ; < <&) &) C

Note

The MSC_WDATA and MSC_ADDRB registers are not retained when entering EM2 or lower energy modes.

7.3.5.1 Double-Word Writes

EFM32GG devices have the ability to do double writes to the flash. This is enabled by setting WDOUBLE in MSC_WRITECTRL, and only has effect on the main pages of the flash. When double writes are enabled, MSC_WDATA accepts two words before a flash write is started. This is signaled by WDATAREADY in MSC_STATUS going high again after the first word is written. When the second word is written, the actual write begins.

7.3.5.2 Low-Power Erase

Because of the fast flash in EFM32GG these devices consume approximately twice the amount of current while doing erase operations when compared to other EFM32 devices. To limit the maximum erase current, the erase operation can be slowed down. Set LPERASE in MSC_WRITECTRL to double the erase time, halving the erase current.

7.3.5.3 Low-Power Write

The maximum write current can also be limited, by slowing down write operations by setting LPWRITE in MSC_WRITECTRL. For single-word writes, this has no effect on write time, but for consecutive writes, the write-time doubles. LPWRITE cannot be set while WDOUBLE is set.

7.3.5.4 Read-While-Write

Reading from the lower half of the flash is possible while writing/erasing from the upper half of the flash and vice versa. Enable read-while-write by setting RWWEN in MSC_WRITECTRL.

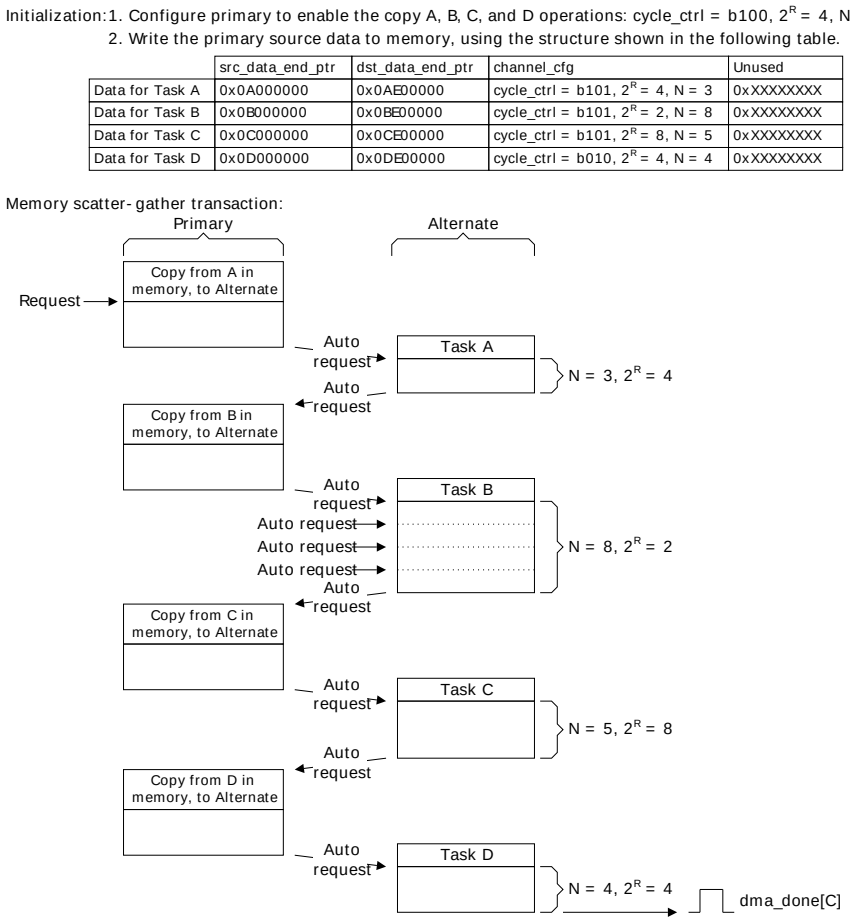
The information pages can be written and erased while running code from the lower half of the flash.

7.3.5.5 Mass erase

A mass erase can be initiated from software using ERASEMAIN0 and ERASEMAIN1 in MSC_WRITECMD. These commands will start a masserase on the lower and upper half of the flash respectively. Prior to initiating a mass erase, MSC_MASSLOCK must be unlocked by writing 0x631A to it. After a mass erase has been started, this register can be locked again to prevent runaway code from accidentally triggering a mass erase.

The regular flash page lock bits will not prevent a mass erase. To prevent software from initiating mass erases, use the mass erase lock bits in the mass erase lock word (MLW).

Figure 8.4. Memory scatter-gather example



In Figure 8.4 (p. 58) :

Initialization

1. The host processor configures the primary data structure to operate in memory scatter-gather mode by setting cycle_ctrl to b100. Because a data structure for a single channel consists of four words then you must set 2^R to 4. In this example, there are four tasks and therefore N is set to 16.
2. The host processor writes the data structure for tasks A, B, C, and D to the memory locations that the primary src_data_end_ptr specifies.
3. The host processor enables the channel.

The memory scatter-gather transaction commences when the controller receives a request on dma_req[ ] or a manual request from the host processor. The transaction continues as follows:

Primary, copy A

1. After receiving a request, the controller performs four DMA transfers. These transfers write the alternate data structure for task A.
2. The controller generates an auto-request for the channel and then arbitrates.

Task A

3. The controller performs task A. After it completes the task, it generates an auto-request for the channel and then arbitrates.

Primary, copy B

4. The controller performs four DMA transfers. These transfers write the alternate data structure for task B.
5. The controller generates an auto-request for the channel and then arbitrates.

Task B

6. The controller performs task B. After it completes the task, it generates an auto-request for the channel and then arbitrates.

Primary, copy C

7. The controller performs four DMA transfers. These transfers write the alternate data structure for task C.

threshold is programmed using BUENRANGE and BUENTHRES in EMU_BUINACT. BUENRANGE decides the voltage range for the BOD, while BUENTHRES is used for tuning of the BOD threshold. Refer to Section 10.3.4.5 (p. 113) for details regarding BOD calibration.

Note

BUVINPEN in EMU_ROUTE is by default set. If Backup mode is not to be used, this bit should be cleared.

Note

The voltage on BU_VIN has to be above the threshold for the BOD sensing on BU_VIN to enter backup mode.

The BU_STAT pin can be used to indicate whether or not the system is in backup mode. To enable exporting of the backup mode status, set STATEN in EMU_BUCTRL and enable the GPIO clock. The BU_STAT pin is driven to BU_VIN when backup mode is active and to ground otherwise.

10.3.4.4 Leaving backup mode

To exit backup mode, the voltage on VDD_DREG has to be above the threshold programmed in EMU_BUACT. BUEXRANGE decides the voltage range for backup mode exit, while BUEXTHRES is used for tuning. When leaving backup mode, a system reset is triggered, resetting everything except the backup domain. When backup mode has been active, the BURST bit in RMU_RSTCAUSE is set.

Figure 10.4. Entering and leaving backup mode

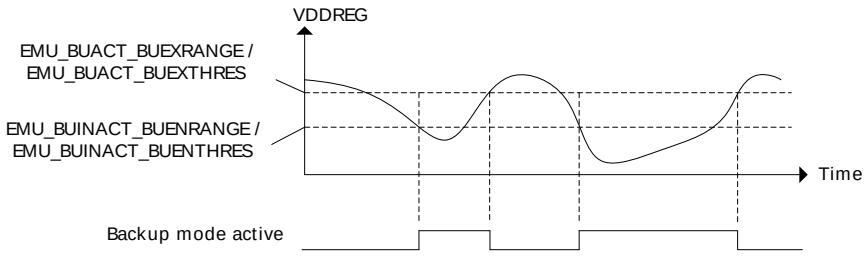


Figure 10.4 (p. 113) illustrates how the BOD sensing on VDD_DREG can be programmed to implement hysteresis on entering and exiting backup mode.

10.3.4.5 Threshold calibration

The thresholds for entering and exiting backup mode are configured in the EMU_BUINACT and EMU_BUACT registers, respectively. Calibration of these thresholds is performed during production test, but may also be performed using the DAC. The calibration values for the BODs sensing on unregulated power and BU_VIN, BUBODUNREG and BUBODBUVIN respectively, are available in EMU_BUBODVINCAL and EMU_BUBODUNREGCAL. These registers are written during production test and should not be modified except for calibrating the Backup BOD sensing on VDD_DREG, as described in the following section.

Setting BODCAL in EMU_BUCTRL will enable a mode where the BOD is sensing the DAC output, as depicted in Figure 10.5 (p. 114) . For the BODCAL bit to take effect, the backup power enable bit, EN in EMU_BUCTRL, has to be cleared. The procedure for BOD calibration is as follows:

- Clear EN and set BODCAL in EMU_BUCTRL.
- Store the values in EMU_BUBODVINCAL and EMU_BUBODUNREGCAL before clearing these registers.

14.5.37 EBI_TFTPIXEL - TFT Alpha Blending Result Pixel Register

Offset	Bit Position																													
0x090	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0000															
Access																	R															
Name																	DATA															

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)		
15:0	DATA	0x0000	R	Alpha Blending Result
	RGB result of Alpha Blending operation according to the format defined in RGBMODE.			

14.5.38 EBI_TFTMASK - TFT Masking Register

Offset	Bit Position																													
0x094	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0000															
Access																	RW															
Name																	TFTMASK															

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)		
15:0	TFTMASK	0x0000	RW	TFT Mask Value
	Sets the mask value. Data write transactions matching this value are suppressed.			

14.5.39 EBI_IF - Interrupt Flag Register

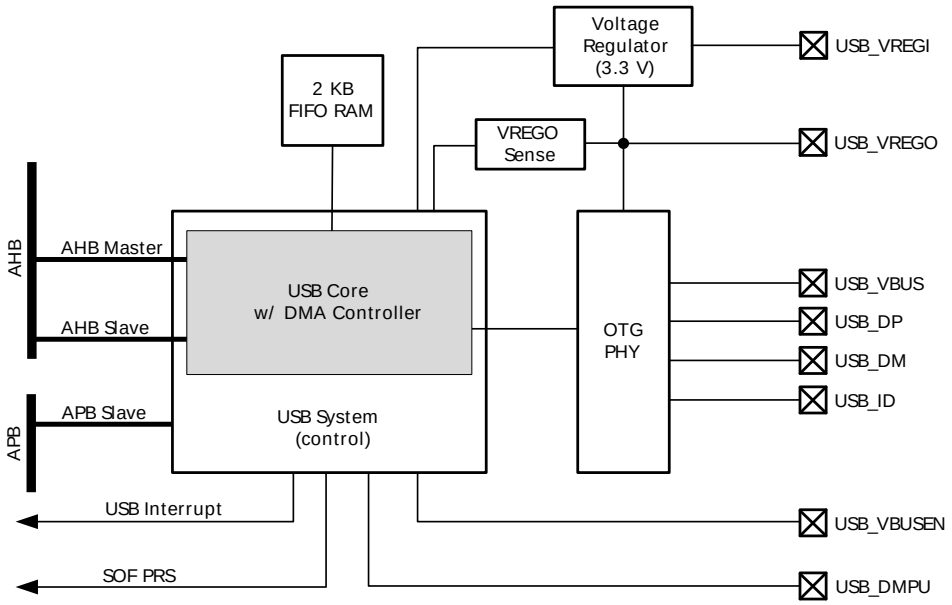
Offset	Bit Position																													
0x098	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0	0	0	0	0	0
Access																											R	R	R	R	R	R
Name																											DDJIT	DDEMPTY	VFPORCH	VBPORCH	HSYNC	VSYNC

- SRP detection in EM2 (during host session off)
- Soft connect/disconnect
- Full OTG support
  - Compliant with On-The-Go and Embedded Host Supplement to the USB Revision 2.0 Specification, Revision 2.0
  - Compliant with USB On-The-Go Supplement, Revision 1.3
  - Supports Host Negotiation Protocol (HNP) and Session Request Protocol (SRP)
- On-chip PHY
  - Internal pull-up and pull-down resistors
  - Voltage comparators for monitoring VBUS voltage
  - A/B Device identification using ID line
  - Charge/discharge of VBUS for VBUS-pulsing
- Internal 3.3V Regulator
  - Output voltage: 3.3V
  - Output current: 50 mA
  - Input voltage range: 4.0 - 5.5V
  - Enabled automatically when input voltage applied
  - Low quiescent current: 100 uA
  - Dedicated input pin allows regulator to be used in OTG and host configurations
  - Output pin can be used to power the EFM32 itself as well as external components
  - Regulator voltage output sense feature for detecting USB plug/unplug events (also available in EM2/3)

15.3 USB System Description

A block diagram of the USB is shown in Figure 15.1 (p. 243) .

Figure 15.1. USB Block Diagram



The USB consists of a digital logic part, an endpoint RAM, PHY and a voltage regulator with output voltage sensor. The voltage regulator provides a stable 3.3 V supply for the PHY, but can also be used to power the EFM32 itself as well as external components.

The digital logic of the USB is split into two parts: system and core.

Bit	Name	Reset	Access	Description
16	SSTOP	0	W1	Set SSTOP Interrupt Flag
	Write to 1 to set the SSTOP interrupt flag.			
15	CLTO	0	W1	Set Clock Low Interrupt Flag
	Write to 1 to set the CLTO interrupt flag.			
14	BITO	0	W1	Set Bus Idle Timeout Interrupt Flag
	Write to 1 to set the BITO interrupt flag.			
13	RXUF	0	W1	Set Receive Buffer Underflow Interrupt Flag
	Write to 1 to set the RXUF interrupt flag.			
12	TXOF	0	W1	Set Transmit Buffer Overflow Interrupt Flag
	Write to 1 to set the TXOF interrupt flag.			
11	BUSHOLD	0	W1	Set Bus Held Interrupt Flag
	Write to 1 to set the BUSHOLD interrupt flag.			
10	BUSERR	0	W1	Set Bus Error Interrupt Flag
	Write to 1 to set the BUSERR interrupt flag.			
9	ARBLOST	0	W1	Set Arbitration Lost Interrupt Flag
	Write to 1 to set the ARBLOST interrupt flag.			
8	MSTOP	0	W1	Set MSTOP Interrupt Flag
	Write to 1 to set the MSTOP interrupt flag.			
7	NACK	0	W1	Set Not Acknowledge Received Interrupt Flag
	Write to 1 to set the NACK interrupt flag.			
6	ACK	0	W1	Set Acknowledge Received Interrupt Flag
	Write to 1 to set the ACK interrupt flag.			
5:4	Reserved	To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)		
3	TXC	0	W1	Set Transfer Completed Interrupt Flag
	Write to 1 to set the TXC interrupt flag.			
2	ADDR	0	W1	Set Address Interrupt Flag
	Write to 1 to set the ADDR interrupt flag.			
1	RSTART	0	W1	Set Repeated START Interrupt Flag
	Write to 1 to set the RSTART interrupt flag.			
0	START	0	W1	Set START Interrupt Flag
	Write to 1 to set the START interrupt flag.			

16.5.13 I2Cn_IFC - Interrupt Flag Clear Register

Offset	Bit Position																															
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																0	0	0	0	0	0	0	0	0	0	0			0	0	0	0
Access																W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1			W1	W1	W1	W1
Name																SSTOP	CLTO	BITO	RXUF	TXOF	BUSHOLD	BUSERR	ARBLOST	MSTOP	NACK	ACK			TXC	ADDR	RSTART	START

Bit	Name	Reset	Access	Description
31:17	Reserved	To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)		
16	SSTOP	0	W1	Clear SSTOP Interrupt Flag

BIT8DV in USARTn_CTRL can be used to specify the value of the 9th bit without writing to the transmit buffer with USARTn_TXDATAX or USARTn_TXDOUBLEX, giving higher efficiency in multi-processor mode, as the 9th bit is only set when writing address frames, and 8-bit writes to the USART can be used when writing the data frames.

17.3.2.9 Collision Detection

The USART supports a basic form of collision detection. When the receiver is connected to the output of the transmitter, either by using the LOOPBK bit in USARTn_CTRL or through an external connection, this feature can be used to detect whether data transmitted on the bus by the USART did get corrupted by a simultaneous transmission by another device on the bus.

For collision detection to be enabled, CCEN in USARTn_CTRL must be set, and the receiver enabled. The data sampled by the receiver is then continuously compared with the data output by the transmitter. If they differ, the CCF interrupt flag in USARTn_IF is set. The collision check includes all bits of the transmitted frames. The CCF interrupt flag is set once for each bit sampled by the receiver that differs from the bit output by the transmitter. When the transmitter output is disabled, i.e. the transmitter is tristated, collisions are not registered.

17.3.2.10 SmartCard Mode

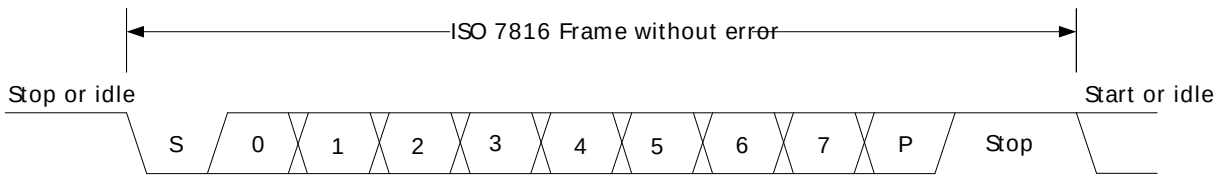
In SmartCard mode, the USART supports the ISO 7816 I/O line T0 mode. With exception of the stop-bits (guard time), the 7816 data frame is equal to the regular asynchronous frame. In this mode, the receiver pulls the line low for one baud, half a baud into the guard time to indicate a parity error. This NAK can for instance be used by the transmitter to re-transmit the frame. SmartCard mode is a half duplex asynchronous mode, so the transmitter must be tristated whenever not transmitting data.

To enable SmartCard mode, set SCMODE in USARTn_CTRL, set the number of databits in a frame to 8, and configure the number of stopbits to 1.5 by writing to STOPBITS in USARTn_FRAME.

The SmartCard mode relies on half duplex communication on a single line, so for it to work, both the receiver and transmitter must work on the same line. This can be achieved by setting LOOPBK in USARTn_CTRL or through an external connection. The TX output should be configured as open-drain in the GPIO module.

When no parity error is identified by the receiver, the data frame is as shown in Figure 17.12 (p. 464) . The frame consists of 8 data bits, a parity bit, and 2 stop bits. The transmitter does not drive the output line during the guard time.

Figure 17.12. USART ISO 7816 Data Frame Without Error



If a parity error is detected by the receiver, it pulls the line I/O line low after half a stop bit, see Figure 17.13 (p. 465) . It holds the line low for one bit-period before it releases the line. In this case, the guard time is extended by one bit period before a new transmission can start, resulting in a total of 3 stop bits.

Offset	Bit Position																															
0x2C0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0xXX							0xXX							0xXX					
Access													RW							RW							RW					
Name													MEASUREDLY							SAMPLEDLY							EXTIME					

Bit	Name	Reset	Access	Description
31:20	Reserved	To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)		
19:13	MEASUREDLY	0xXX	RW	Set measure delay
	Configure measure delay. Sensor measuring is delayed for MEASUREDLY+1 EXCLK cycles.			
12:6	SAMPLEDLY	0xXX	RW	Set sample delay
	Configure sample delay. Sampling will occur after SAMPLEDLY+1 SAMPLECLK cycles.			
5:0	EXTIME	0xXX	RW	Set excitation time
	Configure excitation time. Excitation will last EXTIME+1 EXCLK cycles.			

25.5.28 LESENSE_CHx_INTERACT - Scan configuration (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 20) .

Offset	Bit Position																															
0x2C4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													X	X	X	0xX		0xX		X	0xXXX											
Access													RW	RW	RW	RW		RW		RW	RW											
Name													ALTEX	SAMPLECLK	EXCLK	EXMODE		SETIF		SAMPLE	ACMPTHRES											

Bit	Name	Reset	Access	Description
31:20	Reserved	To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)		
19	ALTEX	X	RW	Use alternative excite pin
	If set, alternative excite pin will be used for excitation			
18	SAMPLECLK	X	RW	Select clock used for timing of sample delay

Value	Mode	Description
0	LFACLK	LFACLK will be used for timing
1	AUXHFRCO	AUXHFRCO will be used for timing

Bit	Name	Reset	Access	Description
17	EXCLK	X	RW	Select clock used for excitation timing

Value	Mode	Description
0	LFACLK	LFACLK will be used for timing
1	AUXHFRCO	AUXHFRCO will be used for timing

31.5.7 AES_IFC - Interrupt Flag Clear Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0
Access																																W1
Name																																DONE

Bit	Name	Reset	Access	Description
31:1	Reserved	To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)		
0	DONE	0	W1	Encryption/Decryption Done Interrupt Flag Clear
	Write to 1 to clear encryption/decryption done interrupt flag			

31.5.8 AES_DATA - DATA Register

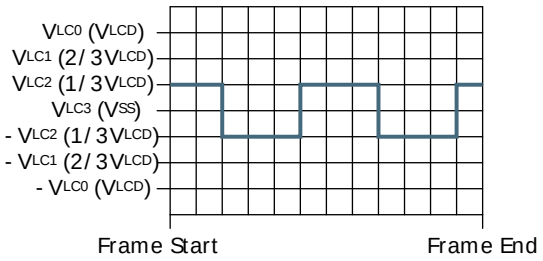
Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Access	RW																															
Name	DATA																															

Bit	Name	Reset	Access	Description
31:0	DATA	0x00000000	RW	Data Access
	Access data through this register.			

31.5.9 AES_XORDATA - XORDATA Register

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Access	RW																															
Name	XORDATA																															

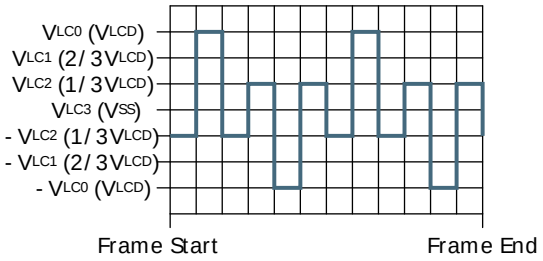
Figure 33.30. LCD 1/3 Bias and Triplex Multiplexing - LCD_SEG0-LCD_COM0



1/3 bias and triplex multiplexing - LCD_SEG0-LCD_COM1

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.64 \times V_{LCD_OUT}$
- The LCD display pixel that is connected to LCD_SEG0 and LCD_COM1 will be ON with this waveform

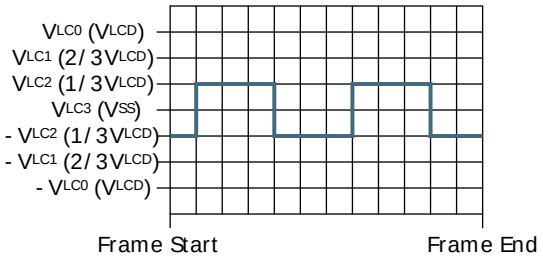
Figure 33.31. LCD 1/3 Bias and Triplex Multiplexing - LCD_SEG0-LCD_COM1



1/3 bias and triplex multiplexing - LCD_SEG0-LCD_COM2

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.33 \times V_{LCD_OUT}$
- The LCD display pixel that is connected to LCD_SEG0 and LCD_COM2 will be OFF with this waveform

Figure 33.32. LCD 1/3 Bias and Triplex Multiplexing - LCD_SEG0-LCD_COM2



33.3.3.6 Waveforms with 1/3 Bias and Quadruplex Multiplexing

In this mode, each frame is divided into 8 periods. All COM lines can be multiplexed with all segment lines. Figures show 1/3 bias and quadruplex multiplexing (waveforms show two frames).

Table of Contents

1. Energy Friendly Microcontrollers 2
1.1. Typical Applications 2
1.2. EFM32GG Development 2
2. About This Document 3
2.1. Conventions 3
2.2. Related Documentation 4
3. System Overview 5
3.1. Introduction 5
3.2. Block Diagram 5
3.3. Features 6
3.4. Energy Modes 7
3.5. Product Overview 8
3.6. Device Revision 10
4. System Processor 12
4.1. Introduction 12
4.2. Features 12
4.3. Functional Description 13
5. Memory and Bus System 15
5.1. Introduction 15
5.2. Functional Description 16
5.3. Access to Low Energy Peripherals (Asynchronous Registers) 20
5.4. Flash 22
5.5. SRAM 23
5.6. Device Information (DI) Page 23
6. DBG - Debug Interface 25
6.1. Introduction 25
6.2. Features 25
6.3. Functional Description 25
6.4. Debug Lock and Device Erase 26
6.5. Register Map 28
6.6. Register Description 28
7. MSC - Memory System Controller 30
7.1. Introduction 30
7.2. Features 31
7.3. Functional Description 31
7.4. Register Map 38
7.5. Register Description 38
8. DMA - DMA Controller 48
8.1. Introduction 48
8.2. Features 48
8.3. Block Diagram 49
8.4. Functional Description 50
8.5. Examples 70
8.6. Register Map 71
8.7. Register Description 72
9. RMU - Reset Management Unit 97
9.1. Introduction 97
9.2. Features 97
9.3. Functional Description 97
9.4. Register Map 102
9.5. Register Description 102
10. EMU - Energy Management Unit 105
10.1. Introduction 105
10.2. Features 105
10.3. Functional Description 106
10.4. Register Map 117
10.5. Register Description 117
11. CMU - Clock Management Unit 126
11.1. Introduction 126
11.2. Features 126
11.3. Functional Description 127
11.4. Register Map 136
11.5. Register Description 137
12. WDOG - Watchdog Timer 159
12.1. Introduction 159
12.2. Features 159
12.3. Functional Description 159
12.4. Register Map 161
12.5. Register Description 161
13. PRS - Peripheral Reflex System 164
13.1. Introduction 164