Release Notes

MSPM0G and MSPM0L Families: Functional Differences With Early Samples

ABSTRACT

This document provides advisories for known differences in functionality between early experimental samples and final production material for the following MSPM0 MCU device families:

- **MSPM0L 32MHz mixed-signal MCUs:**
  - MSPM0L110x
  - MSPM0L13xx
- **MSPM0G 80MHz mixed-signal MCUs:**
  - MSPM0G110x
  - MSPM0G150x
  - MSPM0G310x
  - MSPM0G350x

This document is an addendum to the technical reference manual(s) for the MSPM0G and MSPM0L device families. When working with early samples for these families, the advisories in this document must be taken into account during evaluation.

The advisories listed in this document are categorized into two types:

- Known design exceptions in early experimental samples with respect to the functional specifications given in the corresponding technical reference manual (see Section 1)
- Planned design enhancements to the functional specifications given in the corresponding technical reference manual which are intended for inclusion in future experimental sample revisions and production material (see Section 2)

1 Advisories for Early Samples

The following advisories are known design exceptions to functional specifications on early experimental samples. Advisories are categoried into debug only advisories and functional advisories. Debug only advisories only impact the debug capabilities of the device, and do not affect device performance when debug capabilities are not used.

1.1 Debug Only Advisories

The following advisories are known design exceptions to functional specifications on early experimental samples which affect only debug operation.

1.1.1 Debug Only Advisories Matrix

Table 1-1 lists the advisories for early samples which affect debug operation only.

Table 1-1. Debug Only Advisories Matrix

MODULE	DESCRIPTION	DEVICE FAMILY					
		MSPM0L110x	MSPM0L13xx	MSPM0G110x	MSPM0G150x	MSPM0G310x	MSPM0G350x
DEBUG	*DEBUGSS_01* Wait-in-reset command may cause the debug probe connection to fail	Y	Y	Y	Y	Y	Y
DEBUG	*DEBUGSS_02* Debug probe may not be able to connect to a device in STOP or STANDBY mode	Y	Y	Y	Y	Y	Y
DEBUG	*DEBUGSS_03* EnergyTrace+ is not supported in STOP and STANDBY mode	Y	Y	Y	Y	Y	Y

1.1.2 Debug Only Advisory Descriptions

The following advisories are known design exceptions to functional specifications which affect debug operation.

DEBUGSS_01

Wait-in-reset command may cause the debug probe connection to fail

Revisions Affected Revision 0

Details Sending a wait-in-reset command via the debug sub system mailbox (DSSM) may cause the debug probe to disconnect and not reconnect.

Workaround None.

DEBUGSS_02

Debug probe may not be able to connect to a device in STOP or STANDBY mode

Revisions Affected Revision 0

Details The debug sub system may not wake the device from STOP or STANDBY operating modes to service an AHB-AP request from a debug probe. This may cause a debug connection error.

This behavior may occur in the following situations:

Scenario 1: Connecting to a blank device

When attempting to connect to a blank device (a device with an unprogrammed reset vector and/or stack pointer), a debug connection error may occur if the device was powered on for some time before the debug connection attempt was made. The default boot behavior of an empty device (empty reset vector and stack pointer) is to enter the boot strap loader (if enabled), which times out into STANDBY mode if no BSL session is started within 10 seconds. Once the device enters STANDBY, the debug probe is not capable of waking the device to service an AHB-AP request, resulting in a debug error.

Scenario 2: Connecting to a device which is in STOP or STANDBY mode

When attempting to connect to a programmed device running application code which enters STOP or STANDBY mode, a debug connection error may occur if the application code enters STOP or STANDBY mode before the debug probe is able to connect.

Scenario 3: Disconnection from a device which entered STOP or STANDBY mode

When a debug connection is active, if the device is running a program that enters STOP or STANDBY for more than several seconds, the debug connection will time out and a debug error may occur. If the device remains in STOP or STANDBY and does not return to RUN or SLEEP mode, the debug probe may not be able to re-connect.

Workarounds The following workarounds may be used to establish a debug connection in the scenarios given above:

Scenario 1 Workarounds

Note

Workarounds require hardware access to the NRST pin.

1. When attempting to establish a debug connection with a blank device, toggle the NRST pin manually immediately before attempting to start the connection. The connection must be established within 10 seconds.

DEBUGSS_02 (continued)

Debug probe may not be able to connect to a device in STOP or STANDBY mode

2. Use version *0.20.00.06.eng* of the MSP support package for Code Composer Studio, and ensure that the NRST connection between the target device and the debug probe is made before attempting to start the connection.

Scenario 2 Workarounds

The following workarounds may be used to connect to a device in scenario 2:

1. When attempting to establish a debug connection with a device running application code which enters STOP or STANDBY for extended periods of time, assert a boot strap loader (BSL) entry sequence immediately before attempting to start the connection. The proper sequence is to raise the BSL_invoke pin high while toggling the NRST pin. The connection must be established within 10 seconds.

   Note

   Workaround requires hardware access to the NRST pin and the BSL_invoke pin. The NRST and BSL_invoke pins are connected to momentary switches on both the LaunchPad and the target socket (TS) board evaluation modules. Switch S1 is connected to the BSL_invoke pin (GPIO PA18), and switch S3 is connected to the NRST pin.
2. When developing code when enters STOP and STANDBY, optionally insert a delay of several seconds before initially entering STOP or STANDBY mode after a reset, and apply one of the Scenario 1 workarounds.

   Note

   Workaround requires hardware access to the NRST pin and special modification of application code.
3. If option 1 and 2 above are not feasible (for example, if the BSL_Invoke pin is not accessible, and software is already programmed into the device which enters STOP/STANDBY with no delay), a device may be recovered by issuing a mass erase DSSM command to erase the MAIN flash memory and bring the device back to Scenario 1.

   Note

   Workaround requires hardware access to the NRST pin, and requires use of debug sub system mailbox (DSSM) commands.

Scenario 3

Configure application code to wake up from STOP or STANDBY at a rate of once per second or faster to prevent an active debug probe connection from timing out when the device is in STOP or STANDBY mode.

Note

Workaround requires special modification of application code.

DEBUGSS_03

EnergyTrace+ is not supported in STOP and STANDBY mode

Revisions Affected Revision 0

Details The debug sub system does not support EnergyTrace+ in STOP and STANDBY operating modes.

Workarounds None.

1.2 Functional Advisories

The following advisories are known design exceptions to specified functionality of early experimental samples.

1.2.1 Functional Advisories Matrix

Table 1-2 lists the advisories for early samples by module.

Table 1-2. Advisories Matrix

MODULE	DESCRIPTION	DEVICE FAMILY					
		MSPM0L110x	MSPM0L13xx	MSPM0G110x	MSPM0G150x	MSPM0G310x	MSPM0G350x
PMCU	*PMCU_01* HFCLK startup monitor does not support the full HFCLK frequency range	-	-	Y	Y	Y	Y
PMCU	*PMCU_02* NRST 1-second POR counter does not reset	Y	Y	Y	Y	Y	Y
PMCU	*PMCU_03* Reset cause register does not indicate if a SHUTDNSTOREx parity error caused a POR	Y	Y	Y	Y	Y	Y
PMCU	*PMCU_04* COMP0 mode is used to set the startup time for all COMP modules	-	-	-	Y	-	Y
PMCU	*PMCU_05* Software-triggered BOOTRST will reset the RTC and LFCLK/LFXT state	-	-	Y	Y	Y	Y
PMCU	*PMCU_06* The frequency clock counter (FCC) does not support external input (FCC_IN)	Y	Y	Y	Y	Y	Y
PMCU	*PMCU_07* The temperature sensor in the PMCU is not available	Y	Y	Y	Y	Y	Y
UART	*UART_01* Idle line is not correctly generated in multiprocessor UART mode	Y	Y	Y	Y	Y	Y
UART	*UART_02* UART TX FIFO interrupts do not trigger at the configured FIFO depth	Y	Y	Y	Y	Y	Y
UART	*UART_03* UART baud rate may not be correctly generated in 3X oversampling mode	Y	Y	Y	Y	Y	Y
UART	*UART_04* UART0_TX and UART0_RX may not be functional when muxed to PA.0 and PA.1	Y	Y	Y	Y	Y	Y
UART	*UART_05* Uart may not generate the trigger to the DMA after the first packet is transmitted.						
SPI	*SPI_01* Receive timeout interrupt does not correctly trigger	Y	Y	Y	Y	Y	Y
SPI	*SPI_02* SPI CD line transition may not be correctly aligned to the data stream	Y	Y	Y	Y	Y	Y
CRC	*CRC_01* The CRC module registers are not retained in STOP and STANDBY mode	-	-	Y	Y	Y	Y
CRC	*CRC_02* Writes to the lower half-word of CRCSEED may be ignored	Y	Y	Y	Y	Y	Y
AES	*AES_01* AES block cipher modes with DMA are not supported	-	-	-	Y	Y	Y
AES	*AES_02* AESKEYWR bit must be written twice to take effect	-	-	-	Y	Y	Y
AES	*AES_03* DINWR bit must be written twice to take effect	-	-	-	Y	Y	Y
TRNG	*TRNG_01* False CMD_FAIL IRQ may be generated	-	-	-	Y	Y	Y
COMP	*COMP_01* COMP does not start properly when VREF is not enabled	-	Y	-	Y	-	Y
COMP	*COMP_02* COMP module does not enable VBOOST in on-demand mode	-	Y	-	Y	-	Y
OPA	*OPA_01* Burnout current source is not available	-	Y	-	Y	-	Y
OPA	*OPA_02* Read/write operations on the OPA CFG registers may return or store invalid data	-	Y	-	Y	-	Y
OPA	*OPA_03* OPA trim parameters are not applied automatically	-	Y	-	Y	-	Y
OPA	*OPA_04* OPA module does not enable VBOOST in on-demand mode	-	Y	-	Y	-	Y
OPA	*OPA_05* OPA module may not function properly when MFCLK is disabled	-	Y	-	Y	-	Y
DAC	*DAC_01* DAC does not start properly when VREF is not enabled	-	-	-	Y	-	Y
DAC	DAC_02 DAC cannot generate continuous waveform without CPU intervention.	-	-	-	Y	-	Y

Table 1-2. Advisories Matrix (continued)

MODULE	DESCRIPTION	DEVICE FAMILY					
		MSPM0L110x	MSPM0L13xx	MSPM0G110x	MSPM0G150x	MSPM0G310x	MSPM0G350x
DAC	DAC_03 If DAC DMATRIGEN is set before DMA is configured, DAC and DMA will stuck.	-	-	-	Y	-	Y
GPAMP	*GPAMP_01* GPAMP is not available	Y	Y	Y	Y	Y	Y
WWDT	*WWDT_01* WWDT counter does not halt in low power modes when STISM bit is set	Y	Y	-	-	-	-
BSL	*BSL_01* BSL may incorrectly enter UART mode instead of I2C mode due to noise on UART RX	Y	Y	Y	Y	Y	Y

1.2.2 Functional Advisory Descriptions

The following advisories are known design exceptions to functional specifications.

PMCU_01

HFCLK startup monitor does not support the full HFCLK frequency range

Revisions Affected Revision 0

Details The HFCLK startup monitor (for testing HFXT and HFCLK_IN when HFXT is enabled or HFCLK_IN is selected) does not support testing HFCLK frequencies below 8MHz. If the HFCLK frequency is <8MHz, the HFCLK startup monitor may report a failed startup condition.

Workaround If HFCLK frequencies between 4MHz and 8MHz are used with early samples, the HFCLK startup monitor may be disabled by setting the HFCLKFLTCHK bit in the HFCLKCLKCFG register in SYSCTL.

PMCU_02

NRST 1-second POR counter does not reset

Revisions Affected Revision 0

Details The NRST pin may generate a POR even if the NRST pin is held for less than one second. A POR is generated incorrectly if the NRST pin is held for shorter intervals multiple times, such that the total time NRST is held is greater than 1 second.

Workaround None.

PMCU_03

Reset cause register does not indicate if a SHUTDNSTOREx parity error caused a POR

Revisions Affected Revision 0

Details A SHUTDNSTOREx memory parity error which occurs while the device is in SHUTDOWN mode will correctly generate a POR, but the reset cause register will indicate that the reset cause was a SHUTDOWN exit, not a SHUTDNSTOREx parity error.

Workaround None.

PMCU_04

COMP0 mode is used to set the startup time for all COMP modules

Revisions Affected Revision 0

Details The COMP0 mode (ultra-low power or high-speed) is used to set the startup time for all comparators. If COMP1/2 are enabled in a mode which is different than that of COMP0, the OUTRDY indication may be given at the incorrect time.

Workaround If COMP1/2 are used in a different mode than COMP0, software may delay the appropriate startup time before using the COMP, rather than depending upon the COMP1/2 OUTRDY indication.

PMCU_05

Software-triggered BOOTRST will reset the RTC and LFCLK/LFXT state

Revisions Affected	Revision 0
Details	A software-triggered BOOTRST will reset the RTC and the LFCLK/LFXT state. This will cause a loss of the RTC time base.
Workaround	If possible, use a software-triggered SYSRST instead.

PMCU_06

The frequency clock counter (FCC) does not support external input (FCC_IN)

Revisions Affected	Revision 0
Details	The frequency clock counter (FCC) in the PMCU does not support the use of a digital input pin to either clock the FCC counter (as the FCC source) or to set the trigger time over which pulses of the FCC source clock are counted.
Workaround	Use the internal FCC sources and an the internal FCC trigger signals.

PMCU_07

The temperature sensor in the PMCU is not available

Revisions Affected	Revision 0
Details	The temperature sensor is not available.
Workaround	None.

UART_01

Idle line is not correctly generated in multiprocessor UART mode

Revisions Affected	Revision 0
Details	The 11-bit UART idle period is not correctly inserted when the UART SENDIDLE bit is set.
Workaround	None.

UART_02

UART TX FIFO interrupts do not trigger at the configured FIFO depth

Revisions Affected	Revision 0
Details	The UART module TX FIFO interrupt only generates an interrupt when the TX FIFO is empty, regardless of the configured TX FIFO interrupt level.
Workaround	None.

UART_03

UART baud rate may not be correctly generated in 3X oversampling mode

Revisions Affected	Revision 0

UART_03 (continued)

UART baud rate may not be correctly generated in 3X oversampling mode

Details The UART module may not correctly generate the UART baud rate when the UART is used in 3X oversampling mode (for operation from a 32kHz clock source).

Workaround Use either the 8X or 16X oversampling mode for baud rate generation.

UART_04

UART0_TX and UART0_RX may not be functional when muxed to PA.0 and PA.1

Revisions Affected Revision 0

Details UART0 may not function properly when PA.0 and PA.1 (IOMUX PINCM1 and PINCM2, respectively) are used for UART0_TX and UART0_RX, respectively.

Workaround Select alternate pin locations for UART0_TX and UART0_RX.

UART_05

Uart may not generate the trigger to the DMA after the first packet is transmitted.

Revisions Affected Revision 0

Details When the UART is configured to be used with the DMA for transmitting packets, the UART may not generate the trigger to the DMA after the first packet is transmitted.

Workaround Test setup:

- MSPM0G device (READY/trimmed)
- MCLK=SYSOSC=32MHZ
- No low power modes
- UART0 is tested
- DMA channel 0 is tested

UART0 setup:

- Clock source is BUSCLK
- 115200 baud 8N1
- DMADONE and EOT interrupts enabled to indicate end of packet transmission
- LVL_1 used for FIFO trigger
- INT_EVENT2 TX FIFO trigger enabled

DMA CH0 setup:

- Single transfer mode
- Normal mode
- Byte source width and destination width
- Source increment, destination unchanged

When application software wants to write a packet, the DMA source address and DMA size is written, and then the DMA channel is enabled. The first time this happens after reset, the transfer runs to completion and the DMA channel correctly gets disabled automatically. On subsequent packet attempts, sometimes it completes correctly but other times the DMA never gets triggered after being enabled. The behavior is not consistent (sometimes a 5+ packets run before the triggers stop, other times only 1 or 2 packets get sent before triggers stop). Forcing a transfer by forcing "ISET" in INT_EVENT2 in the UART can cause the trigger to propagate to the DMA, but this should not be required and does not seem to always be required.

UART_05 (continued)

Uart may not generate the trigger to the DMA after the first packet is transmitted.

Clearing EVENT2.IMASK.TXINT when UART TX DMA done interrupt is serviced and setting it again after enabling DMA

SPI_01

Receive timeout interrupt does not correctly trigger

Revisions Affected Revision 0

Details The SPI module receive timeout interrupt (RTOUT) is incorrectly set when the SPI module receives data, regardless of whether the data was received before or after the specified timeout period.

Workaround None.

SPI_02

SPI CD line transition may not be correctly aligned to the data stream

Revisions Affected Revision 0

Details The SPI module command/data (CD) line low-to-high transition may occur before transmission of the command is complete.

Workaround

Option 1: Restricted clock configuration

Configure the SPI bus clock (SCLK) to be one-half of the SPI peripheral clock frequency. The command/data line transition timing is correct under this condition.

Option 2: Software control of command/data signal

Disable the command/data control line feature of the SPI peripheral and implement a software-controlled command/data signal by using the CD pin in GPIO mode and splitting the command and data SPI transmissions into two separate SPI transmissions.

CRC_01

The CRC module registers are not retained in STOP and STANDBY mode

Revisions Affected Revision 0

Details The CRC module register contents are lost when the device enters STOP or STANDBY mode.

Workaround Save any CRC context before entering STOP or STANDBY mode, and restore the context following the exit from low power mode.

CRC_02

Writes to the lower half-word of CRCSEED may be ignored

Revisions Affected Revision 0

Details When the CRC module is configured with ENDIANNESS=1 (big endian) and POLYSIZE=1 (CRC-16-CCITT), writes to the lower half-word of the CRCSEED register are ignored.

CRC_02 (continued)

Writes to the lower half-word of CRCSEED may be ignored

Workaround To set the CRC seed value correctly, write the 16-bit seed to the upper half-word of the CRCSEED register instead of the lower half-word. Note that due to the ENDIANESS adjustment, the byte order will still be reversed when read back in CRCOUT.

AES_01

AES block cipher modes with DMA are not supported

Revisions Affected Revision 0

Details The AES module block cipher modes (CMEN=1) for encrypting/decrypting more than 1 block at a time with the DMA controller are not supported.

Workaround Use single block mode with CPU reads and writes.

AES_02

AESKEYWR bit must be written twice to take effect

Revisions Affected Revision 0

Details A software write to set the AESKEYWR bit in the AESASTAT register will not take effect on the first write. The AESKEYWR bit is used by software to indicate that a usable key is present in the AES key memory (for example, to indicate that a decryption key generated in the previous operation should be used for an upcoming decryption operation).

Workaround Set the AESKEYWR bit in the AESASTAT register twice for the status to take effect.

AES_03

DINWR bit must be written twice to take effect

Revisions Affected Revision 0

Details A software write to set the DINWR bit in the AESASTAT register will not take effect on the first write. The DINWR bit is used by software to indicate that usable data is present in the STATE memory (for example, to start OFB and CFB block cipher operations after loading the initialization vector into the STATE memory).

Workaround Set the DINWR bit in the AESASTAT register twice for the status to take effect.

TRNG_01

False CMD_FAIL IRQ may be generated

Revisions Affected Revision 0

Details Writing to a TRNG configuration register before a CMD is written to the TRNG may cause an unexpected command fail interrupt request (IRQ_CMD_FAIL) to be generated.

Workaround Mask the TRNG command fail interrupt (IRQ_CMD_FAIL) when writing to TRNG configuration registers until a CMD is written to the TRNG. Clear the IRQ_CMD_FAIL interrupt status after writing the CMD and before enabling the IRQ_CMD_FAIL interrupt.

COMP_01

COMP does not start properly when VREF is not enabled

Revisions Affected Revision 0

Details The COMP module(s) do not start properly unless the internal VREF module is enabled and ready, regardless of whether the VREF is used by the COMP module.

Workaround

Scenario 1: Only internal VREF is used (no external VREF used)

Enable the internal VREF and wait for the VREF ready indication (READY bit set in the CTL1 register of the VREF) before enabling the COMP module(s). Note that an external capacitor (C_{VREF}) is required between the VREF pin and VSS for the internal VREF to start properly.

Scenario 2: External VREF is planned to be used

The internal VREF and external VREF share the same device pin. When the internal VREF is enabled, an external capacitor (C_{VREF}) is required on the VREF pin. When an external VREF is used, the internal VREF is normally left disabled and the external reference is connected to the VREF pin, together with a bulk capacitor which is suitably matched to the reference. Follow the procedure below to use the COMP(s) with an external VREF configuration:

1. Ensure an external capacitor is populated on the VREF pin (a value in the range of C_{VREF} is recommended)
2. Disable any external reference connected to the VREF pin
3. Enable the internal VREF and wait for the VREF ready indication (READY bit set in the CTL1 register of the VREF)
4. Enable the COMP module(s)
5. Disable the internal VREF
6. Enable the external reference

COMP_02

COMP module does not enable VBOOST in on-demand mode

Revisions Affected Revision 0

Details The COMP module(s) require the analog mux VBOOST circuit in the PMCU to function properly. When the VBOOST circuit is in ONDEMAND mode (only enabled when a COMP, GPAMP, or OPA is enabled), the VBOOST circuit may not become enabled when a COMP module is enabled. This may prevent the COMP from running properly.

Workaround Configure the VBOOST circuit in ONALWAYS mode before enabling COMP module(s).

OPA_01

Burnout current source is not available

Revisions Affected Revision 0

Details The burnout current source in the OPA module is not available.

Workaround None.

OPA_02

Read/write operations on the OPA CFG registers may return or store invalid data

Revisions Affected Revision 0

Details When reading the CFG register in the OPA module, the data returned to the bus by the OPA may be invalid.

When writing to the CFG register in the OPA module, half-word and byte writes may not be properly applied.

Workaround To read back the CFG register, follow the alternate procedure below:

1. Perform a 32-bit word read of the CFG register
2. Perform the following logical translation for the fields in the read word:
   a. The 3-bit PSEL field is BIT6-4 of the read word instead of BIT5-3
   b. The 4-bit NSEL field is BIT11-8 of the read word instead of BIT9-6
   c. The 3-bit MSEL field is BIT14-12 of the read word instead of BIT12-10
   d. The 3-bit GAIN field is BIT18-16 of the read word instead of BIT15-13

When writing to the CFG register, only perform 32-bit word writes using the bit field locations defined in the technical reference manual.

OPA_03

OPA trim parameters are not applied automatically

Revisions Affected Revision 0

Details OPA trim parameters are not applied to the OPA automatically during device start-up. This may result in reduced OPA performance.

Workaround

Option 1: Programming with driver library

Use the latest driver library in the software development kit (SDK). The trim will be loaded by the OPA drivers during initialization of the OPA.

Option 2: Register level programming

Before using an OPA, apply the OPA trim parameters for that OPA via application software using the following procedure:

1. Read the trim parameters from the lower 12 bits (BIT11-0) of the OPA trim source register (see the Table 1-3 below).
2. Write the trim parameters to the lower 12 bits (BIT11-0) of the OPA trim destination register (offset 0x1C00 in the OPA register map) together with the key value of 0x12 in the upper byte (BIT31-24).

Table 1-3. OPA_03 Trim Source Memory

Device Family	OPAx	Trim Source Address
MSPM0L13xx	OPA0	0x400B.1450
	OPA1	0x400B.1454
MSPM0Gx50x	OPA0	0x400B.1480
	OPA1	0x400B.1484

OPA_03 (continued)

OPA trim parameters are not applied automatically

The code example below applies the trim for OPA0 on an MSPM0L130x device (the OPA0_BASE definition is available in the device-specific header file included in the software development kit).

```
uint32_t oaTrim;

oaTrim = *(uint32_t*)(0x400B1450) & 0x0FFF;
*(uint32_t*)(OPA0_BASE + 0x1C00) = oaTrim | 0x12000000;
```

OPA_04

OPA module does not enable VBOOST in on-demand mode

Revisions Affected Revision 0

Details The OPA module(s) require the analog mux VBOOST circuit in the PMCU to function properly. When the VBOOST circuit is in ONDEMAND mode (only enabled when a COMP, GPAMP, or OPA is enabled), the VBOOST circuit may not become enabled when an OPA module is enabled. This may prevent an OPA from running properly.

Workaround Configure the VBOOST circuit in ONALWAYS mode before enabling OPA module(s).

OPA_05

OPA module may not function properly when MFCLK is disabled

Revisions Affected Revision 0

Details The OPA module(s) may not function properly when MFCLK is left disabled.

Workaround Enable MFCLK by setting the MCLKCFG.USEMFTICK bit in SYSCTL before enabling the OPA module(s). The MFCLK may also be enabled by using the SYSCTL driver library function *void DL_SYSCTL_enableMFCLK (void)*.

DAC_01

DAC does not start properly when VREF is not enabled

Revisions Affected Revision 0

Details The DAC module does not start properly unless the internal VREF module is enabled and ready, regardless of whether the VREF is used by the DAC module.

Workaround

Scenario 1: Only internal VREF is used (no external VREF used)

Enable the internal VREF and wait for the VREF ready indication (READY bit set in the CTL1 register of the VREF) before enabling the DAC module. Note that an external capacitor (C_{VREF}) is required between the VREF pin and VSS for the internal VREF to start properly.

Scenario 2: External VREF is planned to be used

The internal VREF and external VREF share the same device pin. When the internal VREF is enabled, an external capacitor (C_{VREF}) is required on the VREF pin. When an external VREF is used, the internal VREF is normally left disabled and the external reference is connected to the VREF pin, together with a bulk capacitor which is suitably

DAC_01 (continued)

DAC does not start properly when VREF is not enabled

matched to the reference. Follow the procedure below to use the DAC with an external VREF configuration:

1. Ensure an external capacitor is populated on the VREF pin (a value in the range of C_{VREF} is recommended)
2. Disable any external reference connected to the VREF pin
3. Enable the internal VREF and wait for the VREF ready indication (READY bit set in the CTL1 register of the VREF)
4. Enable the DAC module
5. Disable the internal VREF
6. Enable the external reference

DAC_02

DAC cannot generate continuous waveform without CPU intervention.

Revisions Affected	Revision 0
Details	DAC DMATRIGEN is cleared while DMA transfer is done, DMA will stop the transfer from RAM to DAC FIFO. If we want to generate continuous waveforms, this shall not be cleared by hardware. Fix it in PG2.0.
Workaround	CPU to trigger DMA after DMA transfer is done everytime.

DAC_03

If DAC DMATRIGEN is set before DMA is configured, DAC and DMA will stuck.

Revisions Affected	Revision 0
Details	If DAC DMATRIGEN is set before DMA is configured, DAC DMA request will be overlooked, thus DAC and DMA will stuck.
Workaround	Configure DMA first before triggering DMA with DMATRIGEN.

GPAMP_01

GPAMP is not available

Revisions Affected	Revision 0
Details	The GPAMP peripheral is not available.
Workaround	None.

WWDT_01

WWDT counter does not halt in low power modes when STISM bit is set

Revisions Affected	Revision 0
Details	The window watchdog timer (WWDT) counter continues counting in low power modes even when the STISM bit is set.
Workaround	None.

BSL_01

BSL may incorrectly enter UART mode instead of I2C mode due to noise on UART RX

Revisions Affected	Revision 0
Details	During BSL entry, UART mode may be incorrectly selected if there is noise present on the BSL UART RX line, even if a host is attempting to connect in I2C mode at the same time. Once UART mode is selected by the BSL, I2C communication is not possible in the active BSL session.
Workaround	Ensure a stable idle state on the BSL UART RX pin when invoking the BSL by actively driving the BSL UART RX pin to an idle state (logic high) or populating a pullup resistor between the pin and VDD.

2 Advisories for Planned Enhancements

The following advisories are planned design enhancements to the functional specifications given in the corresponding technical reference manual which are intended for inclusion in future experimental sample revisions and production material.

2.1 Enhancement Advisories Matrix

Table 2-1 lists the planned enhancement advisories for future samples and production devices by module.

Table 2-1. Enhancement Advisories Matrix

MODULE	DESCRIPTION	DEVICE FAMILY					
		MSPM0L110x	MSPM0L13xx	MSPM0G110x	MSPM0G150x	MSPM0G310x	MSPM0G350x
PMCU	*PMCU_ENH_01* The CLK_OUT signal will be available on a high-speed digital IO.	Y	Y	-	-	-	-
PMCU	*PMCU_ENH_02* SYSCTL will support mapping of error conditions to an NMI on the processor.	Y	Y	Y	Y	Y	Y
UART	*UART_ENH_01* UART peripherals will support majority vote.	Y	Y	Y	Y	Y	Y
SPI	*SPI_ENH_01* SPI peripherals will support the use of CS1/2/3 in target mode.	Y	Y	Y	Y	Y	Y
I2C	*I2C_ENH_01* I2C peripherals in target mode will support generating a start condition interrupt only on a matching I2C address.	Y	Y	Y	Y	Y	Y
I2C	*I2C_ENH_02* I2C peripherals in target mode will support generating a stop condition interrupt only on I2C bus transactions where the I2C module was previously addressed.	Y	Y	Y	Y	Y	Y
I2C	*I2C_ENH_03* I2C peripherals will support a TREQ interrupt.	Y	Y	Y	Y	Y	Y
I2C	*I2C_ENH_04* I2C peripherals will support the 10-bit I2C target addressing mode.	Y	Y	Y	Y	Y	Y
I2C	*I2C_ENH_05* I2C peripherals will support additional SMBUS features.	Y	Y	Y	Y	Y	Y
I2C	*I2C_ENH_06* I2C peripherals will support clock stretching in target mode when waking up SYSOSC.	Y	Y	Y	Y	Y	Y
TIM	*TIM_ENH_01* TIMA0 timer will support 6 capture compare (CC) blocks and complementary output pairs.	-	-	Y	Y	Y	Y
TIM	*TIM_ENH_02* TIMA1 will support complementary output pairs.	-	-	Y	Y	Y	Y
TIM	*TIM_ENH_03* TIMG10 QEI timer in PD1 will be replaced by TIMG8 QEI timer in PD0.	-	-	Y	Y	Y	Y
TIM	*TIM_ENH_04* TIMH0 24-bit timer in PD1 will be replaced by TIMG12 32-bit timer in PD0.	-	-	Y	Y	Y	Y
TIM	*TIM_ENH_05* TIMG6 and TIMG7 general purpose 16-bit timers will be added in PD1.	-	-	Y	Y	Y	Y
TIM	*TIM_ENH_06* FMP revmoed in TIMA and two clear events added	-	-	Y	Y	Y	Y
CRC	*CRC_ENH_01* The CRC will support byte swapping (endian reversal) on the output as well as the input.	Y	Y	Y	Y	Y	Y
COMP	*COMP_ENH_01* COMP peripherals will support a blanking mode.	-	Y	-	Y	-	Y
ARCH	*ARCH_ENH_01* The device configuration (NONMAIN) memory will support additional security configuration options.	Y	Y	Y	Y	Y	Y
DMA	*DMA_ENH_01* The DMA will only support repeat-mode on channel 0.	Y	Y	-	-	-	-
DMA	*DMA_ENH_02* The DMA will only support repeat-mode on channels 0, 1, and 2.	-	-	Y	Y	Y	Y
DMA	*DMA_ENH_03* The DMA will support interrupting an active channel to service other channels.	Y	Y	Y	Y	Y	Y
DMA	*DMA_ENH_04* The DMA will support early IRQ generation.	Y	Y	Y	Y	Y	Y
DMA	*DMA_ENH_05* The DMA will support a stride transfer mode.	Y	Y	Y	Y	Y	Y

Table 2-1. Enhancement Advisories Matrix (continued)

MODULE	DESCRIPTION	DEVICE FAMILY					
		MSPM0L110x	MSPM0L13xx	MSPM0G110x	MSPM0G150x	MSPM0G310x	MSPM0G350x
DMA	*DMA_ENH_06* The DMA will support simple cascading of DMA channels.	Y	Y	Y	Y	Y	Y
SRAM	*SRAM_ENH_01* The SRAM memory will support ECC (SEC/DED)	-	-	-	Y	Y	Y
IOMUX	*IOMUX_ENH_01* The IOMUX will support reduced IO configuration on certain SPI pins when used in SPI mode.	Y	Y	Y	Y	Y	Y

2.2 Enhancement Advisory Descriptions

The following advisories are enhancements currently planned for future samples and production material.

PMCU_ENH_01

The CLK_OUT signal will be available on a high-speed digital IO.

Supported Revisions Revision 1 and greater

Details The PMCU CLK_OUT signal, which may be used to output a system clock externally to a pin, will be made available on a high-speed IO pin in addition to the standard-drive IO pins.

PMCU_ENH_02

SYSCTL will support mapping of error conditions to an NMI on the processor.

Supported Revisions Revision 1 and greater

Details The SYSCTL module will support generation of a non-maskable interrupt (NMI) to the Cortex-M0+ processor upon the following system error conditions:

- Detection of uncorrectable memory errors (e.g. SRAM parity errors, flash ECC DED errors)
- An SRAM invalid address access violation
- Detection of early BOR (when the BOR level is set to BOR1/2/3)
- A watchdog timer violation occurs
- A non-fatal low frequency clock error occurs

An NMI interrupt management register set will be provided, including an interrupt index (IIDX) register to indicate the index highest priority pending NMI source.

UART_ENH_01

UART peripherals will support majority vote.

Supported Revisions Revision 1 and greater

Details The UART module will support majority-vote capture on the RX line for improved robustness.

SPI_ENH_01

SPI peripherals will support the use of CS1/2/3 in target mode.

Supported Revisions	Revision 1 and greater
Details	The SPI module will support the use of chip select 1, 2, and 3 (in addition to chip select 0) when operating in target mode.

I2C_ENH_01

I2C peripherals in target mode will support generating a start condition interrupt only on a matching I2C address.

Supported Revisions	Revision 1 and greater
Details	Revision 0 early experimental samples will generate an I2C start condition interrupt for any start condition seen on the I2C bus. Revision 1 will support generating a start condition interrupt only when the I2C address on the I2C bus matches the target address programmed in the I2C module. This enhancement can reduce CPU interrupts and power consumption when the I2C controller is addressing other modules/devices.

I2C_ENH_02

I2C peripherals in target mode will support generating a stop condition interrupt only on I2C bus transactions where the I2C module was previously addressed.

Supported Revisions	Revision 1 and greater
Details	Revision 0 early experimental samples will generate an I2C stop condition interrupt for any stop condition seen on the I2C bus. Revision 1 will support generating a stop condition interrupt only when the I2C address used to initiate the transaction matched the target address programmed in the I2C module. This enhancement can reduce CPU interrupts and power consumption when the I2C controller is addressing other modules/devices.

I2C_ENH_03

I2C peripherals will support a TREQ interrupt.

Supported Revisions	Revision 1 and greater
Details	The I2C module will include additional support for handling I2C bus restart conditions.

I2C_ENH_04

I2C peripherals will support the 10-bit I2C target addressing mode.

Supported Revisions	Revision 1 and greater
Details	The I2C module will support 10-bit I2C target addresses in addition to standard 7-bit target addresses.

I2C_ENH_05

I2C peripherals will support additional SMBUS features.

Supported Revisions

Revision 1 and greater

Details

The I2C module will support additional SMBUS features, including:

- Address resolution protocol with target arbitration and default device address
- Packet error check generation and verification
- Host protocol
- Extended timeout detection and handling features, including:
  - Controller and target timeout detection
  - Cumulative clock low extended timeout for controller and target
  - Bus idle detection
  - Bus release on timeout

I2C_ENH_06

I2C peripherals will support clock stretching in target mode when waking up SYSOSC.

Supported Revisions

Revision 1 and greater

Details

The I2C module will support stretching the I2C clock line (SCL) in target mode per the I2C specification when a start condition is seen on the bus and the device does not have a fast clock running to support I2C operation. When configured to generate an asynchronous fast clock request, an I2C module may request SYSOSC to be enabled to support correct I2C processing, and the I2C module will stretch SCL until the fast clock is available so that the I2C transaction may continue properly.

TIM_ENH_01

TIMA0 timer will support 6 capture compare (CC) blocks and complementary output pairs.

Supported Revisions

Revision 1 and greater

Details

The TIMA0 (PD1 domain) advanced timer will include 6 capture compare (CC) blocks, with:

- Four complementary PWM output pairs (mapped to the first 4 capture compare blocks)
- Two internal outputs (mapped to the last 2 capture compare blocks)

The TIMA2 (PD1 domain) advanced timer will be removed.

TIM_ENH_02

TIMA1 will support complementary output pairs.

Supported Revisions

Revision 1 and greater

Details

The TIMA0 (PD1 domain) advanced timer will support two complementary output pairs, mapped to the two capture compare (CC) blocks.

TIM_ENH_03

TIMG10 QEI timer in PD1 will be replaced by TIMG8 QEI timer in PD0.

Supported Revisions	Revision 1 and greater
Details	The TIMG10 general purpose timer with quadrature encoder support will be removed from the PD1 power domain, and general purpose timer TIMG8 with quadrature encoder support will be added in the PD0 power domain.

TIM_ENH_04

TIMH0 24-bit timer in PD1 will be replaced by TIMG12 32-bit timer in PD0.

Supported Revisions	Revision 1 and greater
Details	The TIMH0 24-bit timer will be removed from the PD1 power domain, and a general purpose timer TIMG12 with a 32-bit counter and 2 capture compare blocks will be added in the PD0 power domain to support 32-bit counting in STOP and STANDBY modes.

TIM_ENH_05

TIMG6 and TIMG7 general purpose 16-bit timers will be added in PD1.

Supported Revisions	Revision 1 and greater
Details	Two 16-bit timers, TIMG6 and TIMG7, will be added in the PD1 power domain. Both timers will have two capture compare blocks each.

TIM_ENH_06

FMP revmoed in TIMA and two clear events added

Supported Revisions	Revision 1 and greater
Details	FMP(Fault Min Period) revmoed in TIMA and two additional fault lactching clear events (zero and load) added, the names of two events in SDK are MSPGPTIMER_FCTL_FL_LATCH_Z_CLR and MSPGPTIMER_FCTL_FL_LATCH_LD_CLR.

CRC_ENH_01

The CRC will support byte swapping (endian reversal) on the output as well as the input.

Supported Revisions	Revision 1 and greater
Details	The CRC module will have support for optionally swapping the byte order of the CRC digest output.

COMP_ENH_01

COMP peripherals will support a blanking mode.

Supported Revisions	Revision 1 and greater
Details	The COMP module will support configuring a blanking window (output mask window) to prevent output glitches during the specified window.

ARCH_ENH_01

The device configuration (NONMAIN) memory will support additional security configuration options.

Supported Revisions	Revision 1 and greater
Details	The device configuration memory (NONMAIN flash) will provide additional flexibility for configuring the device security policies, including: • Support for fully disabling application debug and/or TI failure analysis while allowing factory reset and/or mass erase • Support for 16-bit pattern-match fields to enable lower security states, with all other values leading to a maximally secure state

DMA_ENH_01

The DMA will only support repeat-mode on channel 0.

Supported Revisions	Revision 1 and greater
Details	The DMA module will be updated to only support repeat-mode on channel 0.

DMA_ENH_02

The DMA will only support repeat-mode on channels 0, 1, and 2.

Supported Revisions	Revision 1 and greater
Details	The DMA module will be updated to only support repeat-mode on channels 0, 1, and 2.

DMA_ENH_03

The DMA will support interrupting an active channel to service other channels.

Supported Revisions	Revision 1 and greater
Details	The DMA module will support a burst block mode for suspending an active (triggered) channel after a configurable number of transfers in order to service other pending channels, after which the suspended channel will resume processing of its trigger.

DMA_ENH_04

The DMA will support early IRQ generation.

Supported Revisions	Revision 1 and greater
Details	The DMA module will support generating an early interrupt request to the CPU to indicate that a transfer will complete within a configurable number of transfers (1, 2, 4, 8, 32, 64, 128, 256). This feature will be available on repeat-capable channels only.

DMA_ENH_05

The DMA will support a stride transfer mode.

Supported Revisions	Revision 1 and greater
Details	The DMA module will support a stride transfer mode for re-organizing the order of data between the source and destination. This feature will be available on repeat-capable channels only.

DMA_ENH_06

The DMA will support simple cascading of DMA channels.

Supported Revisions	Revision 1 and greater
Details	The DMA module will support cascading of DMA channels. Channels will support being triggered upon the completion of activity on another channel.

SRAM_ENH_01

The SRAM memory will support ECC (SEC/DED)

Supported Revisions	Revision 1 and greater
Details	The system memory (SRAM) will include error correction code (ECC) logic in addition to parity logic. The ECC logic will provide support for single error correct and double error detect on the SRAM, evaluated upon each memory access.

IOMUX_ENH_01

The IOMUX will support reduced IO configuration on certain SPI pins when used in SPI mode.

Supported Revisions	Revision 1 and greater
Details	The IOMUX will not support the following IO settings on SPI CIPO pins when the pins are configured for SPI functionality: • Output inversion • Pseudo open-drain (output-high translated to high-impedance)

IOMUX_ENH_01 (continued)

The IOMUX will support reduced IO configuration on certain SPI pins when used in SPI mode.

The IOMUX will not support the following IO setting on SPI CLK pins when the pins are configured for SPI functionality:

- Output inversion

3 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

DATE	REVISION	NOTES
2022-02-23	0.5	• Added BSL advisory 1 • Added DEBUGSS advisory 3 • Updating COMP advisory 1 and DAC advisory 1 with a workaround procedure for using an external voltage reference • Added OPA advisory 4 • Updated DEBUGSS advisory 2 with EVM switch descriptions • Updated OPA advisory 3 • Added OPA advisory 5
2022-01-25	0.4	• Added DEBUGSS advisory 2 • Added UART advisory 4 • Added DAC advisory 1 • Added WWDT advisory 1
2022-01-10	0.3	• Added PMCU advisory 7 • Added DEBUGSS advisory 1 • Added UART advisory 3 • Added SPI advisory 2 • Added CRC advisory 2 • Added AES advisories 2 and 3 • Added TRNG advisory 1 • Added COMP advisories 1 and 2 • Added OPA advisories 1, 2, and 3 • Added GPAMP advisory
2021-12-16	0.2	• Added SRAM enhancement advisory 1 • Added DMA enhancement advisories1-6 • Added IOMUX enhancement advisory 1 • Added CRC advisory 1 • Removed DAC enhancement advisory 1
2021-12-14	0.1	Initial release

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265