AN1401: EFR32xG23 and EFM32PG23 to EFR32xG28 and EFM32PG28 Compatibility and Migration Guide

This application note is for users migrating an existing EFM32PG23, EFR32FG23, or EFR32ZG23 design to EFM32PG28, EFR32FG28, or EFR32ZG28. Migration entails both hardware and software considerations. MCU subsystem compatibility between these devices is covered in detail, but specific radio and security considerations are beyond the scope of this document.

KEY POINTS

- EFR32xG23 and EFM32PG23 and EFR32xG28 and EFM32PG28 maintain a high degree of peripheral and software compatibility.
- Migrating designs from xG23 to xG28 may require hardware and software changes depending on the device's feature set.
- Datasheets for each device detail specific performance metrics, which may differ between EFR32xG23 and EFM32PG23 and EFR32xG28 and EFM32PG28.

1. Introduction

This application note provides guidance on migrating from the xG23 family, which consists of the following:

- EFR32FG23
- EFR32ZG23
- EFM32PG23

...to the xG28 family, which consists of the following:

- EFR32FG28
- EFR32ZG28
- EFM32PG28

This application note highlights the microcontroller-level differences between xG23 and xG28. Certain orderable part numbers (OPNs) of EFR32FG28 and EFR32ZG28 are designed to be both pin- and hardware-compatible, as well as largely software-compatible, with EFR32FG23 and EFR32ZG23, thus requiring only minor designs changes. Although other xG28 OPNs differ sufficiently enough from xG23 to require new board designs, their use of the same hardware design concepts, such as power supply decoupling, and shared peripherals drastically simplifies upgrading to these new devices.

Four factors must be considered when porting a design from xG23 to xG28: pin compatibility, external hardware compatibility, peripheral compatibility, and software compatibility.

1.1 Pins and External Hardware

EFR32FG28 and EFR32ZG28 are available in 48-pin and 68-pin QFN packages. Certain xG28 devices are pin-compatiable with xG23 devices in the 48-pin QFN package. EFM32PG28 is only available in a 68-pin QFN package. For this reason, it is not pin-compatible with EFM32PG23.

A 40-pin QFN package option is not available for xG28. Boards designed for xG23 in this package will need to be modified to use one of the available package options when upgrading to xG28.

More information on the footprints, packages, and external hardware compatibility between xG23 devices and xG28 devices can be found in 2. Pin Compatibility and 3. External Hardware Migration.

1.2 Peripherals and Software

EFR32xG28 and EFM32PG28 MCU functionality is nearly identical to EFR32xG23 and EFM32PG23 such that the source code changes required to upgrade may be minimal to non-existent. 4. Peripheral Compatibility covers the peripheral differences between the two devices.

SDK features used in EFR32xG23 and EFM32PG23 firmware are available for EFR32xG28 and EFM32PG28; however existing binaries built for EFM32PG23, EFR32FG23, or EFR32ZG23 cannot be directly flashed onto EFM32PG28, EFR32FG28, or EFR32ZG28 and vice versa. Furthermore, even when no changes are required to user-developed source code that makes use of SDK APIs (e.g. emlib or RAIL), it is still necessary to rebuild an existing xG23 project against a new SDK release with xG28 support. 5. Software Compatibility covers this in greater detail.

2. Pin Compatibility

Pin compatibility between EFR32xG23 and EFR32xG28 depends on the selected device package and its feature set. Table 2.1 EFR32FG23/EFR32FG28 Pin Compatibility Overview on page 3 provides an overview of the pin compatibility between EFR32FG23 and EFR32FG28. See Table 2.2 EFR32ZG23/EFR32ZG28 Pin Compatibility Overview on page 4 for the same comparison between EFR32ZG23 and EFR32ZG28. Two devices are pin-compatible if both their pin definitions and package footprints are the same. See the *Ordering Information* section of the device datasheet in question for package-specific feature availability.

Note: The pin-compatibility definition above does not necessarily extend to radio transmit power and associated matching networks. See the device datasheet for specific radio matching network component values.

Table 2.1. EFR32FG23/EFR32FG28 Pin Compatibility Overview

Package	EFR32FG23 OPN	EFR32FG28 OPN	Pin Compatible?
QFN40	EFR32FG23A010F256GM40 EFR32FG23A010F512GM40 EFR32FG23A020F256GM40 EFR32FG23A020F512GM40 EFR32FG23B010F128GM40 EFR32FG23B010F512IM40 EFR32FG23B020F128GM40 EFR32FG23B020F512IM40	N/A	No
QFN40 with HFCLKOUT	EFR32FG23A011F512GM40 EFR32FG23A021F512GM40 EFR32FG23B021F512IM40	N/A	No
QFN48 Sub-GHz	EFR32FG23A010F256GM48 EFR32FG23A010F512GM48 EFR32FG23A020F256GM48 EFR32FG23A020F512GM48 EFR32FG23B010F512IM48 EFR32FG23B020F512IM48	EFR32FG28A010F1024GM48 EFR32FG28A110F1024GM48 EFR32FG28A120F1024GM48 EFR32FG28B310F1024IM48 EFR32FG28B320F1024IM48	Yes
QFN48 with HFCLKOUT	EFR32FG23B021F512IM48	N/A	No
QFN48 Dual-Band	N/A	EFR32FG28A112F1024GM48 EFR32FG28A122F1024GM48 EFR32FG28B312F1024IM48 EFR32FG28B322F1024IM48	No
QFN68 Sub-GHz	N/A	EFR32FG28A010F1024GM68 EFR32FG28A110F1024GM68 EFR32FG28A120F1024GM68 EFR32FG28B310F1024IM68 EFR32FG28B320F1024IM68	No
QFN68 Dual-Band	N/A	EFR32FG28A112F1024GM68 EFR32FG28A122F1024GM68 EFR32FG28B312F1024IM68 EFR32FG28B322F1024IM68	No

Table 2.2. EFR32ZG23/EFR32ZG28 Pin Compatibility Overview

Package	EFR32ZG23 OPN	EFR32ZG28 OPN	Pin Compatible?
QFN40	EFR32ZG23A010F512GM40 EFR32ZG23A020F512GM40 EFR32ZG23B010F512IM40 EFR32ZG23B020F512IM40	N/A	No
QFN40 with HFCLKOUT	EFR32ZG23B011F512IM40 EFR32ZG23B021F512IM40	N/A	No
QFN48 Single-Band	EFR32ZG23A010F512GM48 EFR32ZG23A020F512GM48 EFR32ZG23B010F512IM48 EFR32ZG23B020F512IM48	EFR32ZG28A110F1024GM48 EFR32ZG28A120F1024GM48 EFR32ZG28B310F1024IM48 EFR32ZG28B320F1024IM48	Yes
QFN48 Dual-Band	N/A	EFR32ZG28A112F1024GM48 EFR32ZG28A122F1024GM48 EFR32ZG28B312F1024IM48 EFR32ZG28B322F1024IM48	No
QFN68 Single-Band	N/A	EFR32ZG28A110F1024GM68 EFR32ZG28A120F1024GM68 EFR32ZG28B310F1024IM68 EFR32ZG28B320F1024IM68	No
QFN68 Dual-Band	N/A	EFR32ZG28A112F1024GM68 EFR32ZG28A122F1024GM68 EFR32ZG28B312F1024IM68 EFR32ZG28B322F1024IM68	No

3. External Hardware Migration

External hardware components used for EFR32xG23 and EFM32PG23 designs — crystals, passive components used for the supply inputs, and the connector used to attach an external debugger — can be re-used when upgrading to EFR32xG28 and EFM32PG28.

In particular, 39 MHz crystals that have been validated for use in xG23 designs can continue to be used when moving to xG28. While HFXO specifications (excluding the HFCLKOUT option that is not present on xG28) are unchanged between xG23 and xG28, check with Silicon Labs for proprietary use cases that depend on other crystal values.

Excepting some improvements related to trace clock selection on EFR32xG28 and EFM32PG28, debug functionality is unchanged when moving from EFR32xG23 and EFM32PG23. The debug connector pin out remains the same, as do the electrical specifications of the debug interface. Pending any software or firmware upgrades that might be necessary in order to recognize xG28 devices, an existing debug solution used with xG23 ought to be able to support xG28.

Power supply decoupling on xG28 is the same as is used on other Series 2 EFM32/EFR32 devices and is specifically covered in AN0002.2: EFM32 and EFR32 Wireless Gecko Series 2 Hardware Design Considerations. The same is also true of the load capacitor and inductor connected to the buck DCDC converter's VREGSW output node. In particular, there is no need to change the passive components connected to any of the supply inputs or the DCDC converter on an existing xG23 design that uses the 48-pin QFN package when upgrading to a pin-compatible xG28 variant.

When migrating an existing design with EFR32xG23 or EFM32PG23 in a smaller package to EFR32xG28 or EFM32PG28 in a larger package, there are no changes involving the DCDC converter. If the move is from a 40-pin xG23 to a 48-pin xG28, there are also no changes to power supply decoupling because the number of pins involved (one each of AVDD, DECOUPLE, DVDD and IOVDD) remains the same.

Upgrading to the 68-pin QFN brings two additional pins (for a total of three) to supply the common IOVDD rail. All of these must be decoupled with a local 0.1 µF capacitor. The main 1 µF bulk capacitor connected to the single IOVDD on the north side of the 40- and 48-pin packages (adjacent to the AVDD and DVDD supply inputs) can now be connected to whichever of the three IOVDD pins results in the most optimal board layout. As a rule, these and all other power supply decoupling capacitors must be placed as close to the package pins as possible.

The figure below illustrates placement options for the IOVDD decoupling capacitors in designs using the 68-pin QFN.

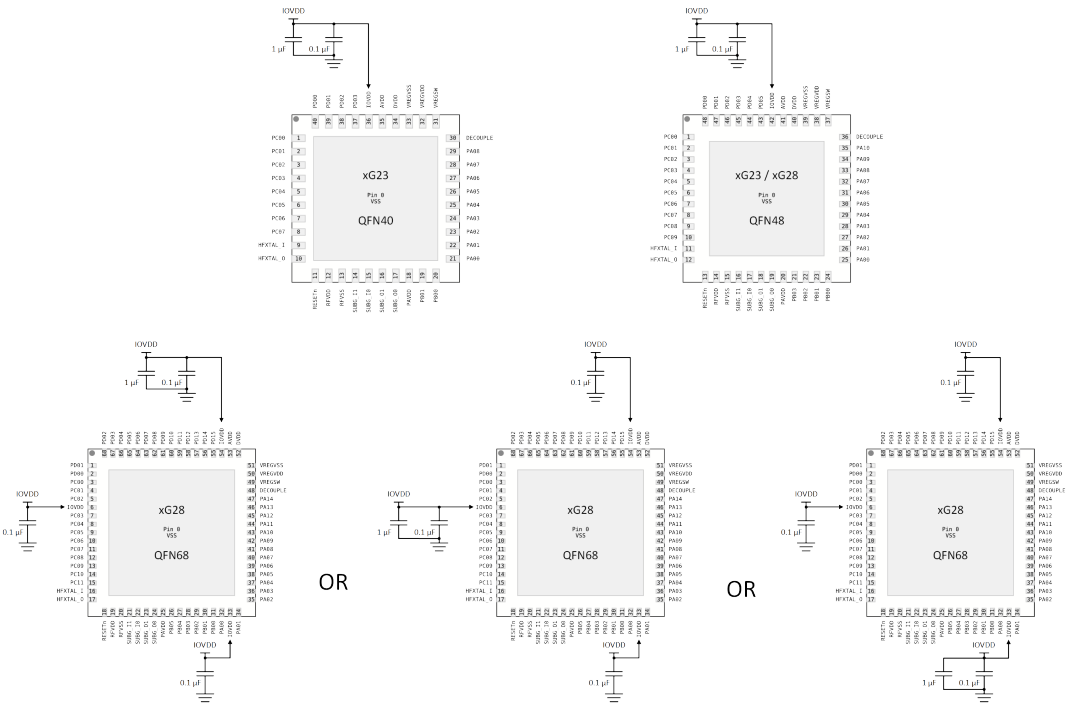

Figure 3.1. IOVDD Decoupling Capacitors

Note: In addition to the supplementary IOVDD pins for QFN68 package, on the MCU only EFM32PG28 QFN68 package, there is also an additional supply pin for DVDD (total of two). Both DVDD pins need to be tied to the same supply net, either the DCDC output node (if used) or other main supply. A main 4.7 µF bulk capacitor should be connected to the node on the north side near the DCDC output, and a 0.1 µF decouple capacitor placed at *each* DVDD pin located as close as possible to the package pins.

4. Peripheral Compatibility

EFR32xG28 and EFM32PG28 are, effectively, variants of EFR32xG23 and EFM32PG23 with larger memories. All peripherals present on xG23 are present on xG28. Some peripherals have small enhancements that do not break firmware backwards compatibility. Additionally, some variants of xG28 are available with the Matrix Vector Processor (MVP) , a compute engine used to accelerate computationally intensive floating point operations, particularly the complex floating point matrix multiplication and addition used in machine learning (ML) algorithms. Peripheral enhancements are discussed in the subsections that follow.

4.1 Analog Comparator (ACMP)

The two analog comparators (ACMPs) on xG28 have enhanced biasing options beyond what is present on xG23. Specifically, the comparators now support bias modes 0 and 1 in addition to the existing modes 2 - 7. These new modes provide lower current options when running continuously, and are selected by writing 0 or 1 to the existing BIAS field in the ACMP Configuration Register (ACMP_CFG).

On xG28, however, the actual warm-up time for bias modes 0 - 3 needs to be extended to 60 µs to account for device-to-device variation relative to the self-timed warm-up reported by the ACMPRDY bit in the status register (ACMP_STATUS). No such requirement exists for using modes 2 - 3 on xG23, so this change will require the addition of a software delay (e.g. using a Sleeptimer callback) after ACMP initialization in firmware that is migrated to xG28.

4.2 Digital-to-Analog Converter (VDAC) PRS Sine Wave Control

EFR32xG23 and EFM32PG23 have a 2-channel, 12-bit voltage output digital-to-analog converter. Different options exist for software control of the outputs (e.g. direct setting via emlib or controlled updates using DMA), but in cases where a sine wave output is needed, the VDAC includes a hardware state machine with a 16-entry lookup table that can control the outputs automatically once configured and enabled. The waveform is output on channel 0 (a complementary waveform is also output on channel 1 if DIFF = 1 in VDAC_CFG) and can be started and stopped under software control by the SINEMODESTART and SINEMODESTOP bits in the VDAC Command Register (VDAC_CMD).

Control of the VDAC outputs with the PRS has been added to EFR32xG28 and EFM32PG28 via the OUTENPRS and SINEMODEPRS bits in the VDAC Configuration Register (VDAC_CFG). When OUTENPRS = 1, the VDAC outputs can be switched on (driven) and off (high-impedance) in response to the detected level of the ASYNCTRIGCH1 PRS input. Similarly, the sine wave output(s) can be started and stopped based on the ASYNCTRIGCH0 PRS input when SINEMODEPRS = 1. Figure 4.1 VDAC Sine Mode PRS Control on page 6 illustrates how the VDAC responds to the sine wave control and output buffer enable PRS inputs.

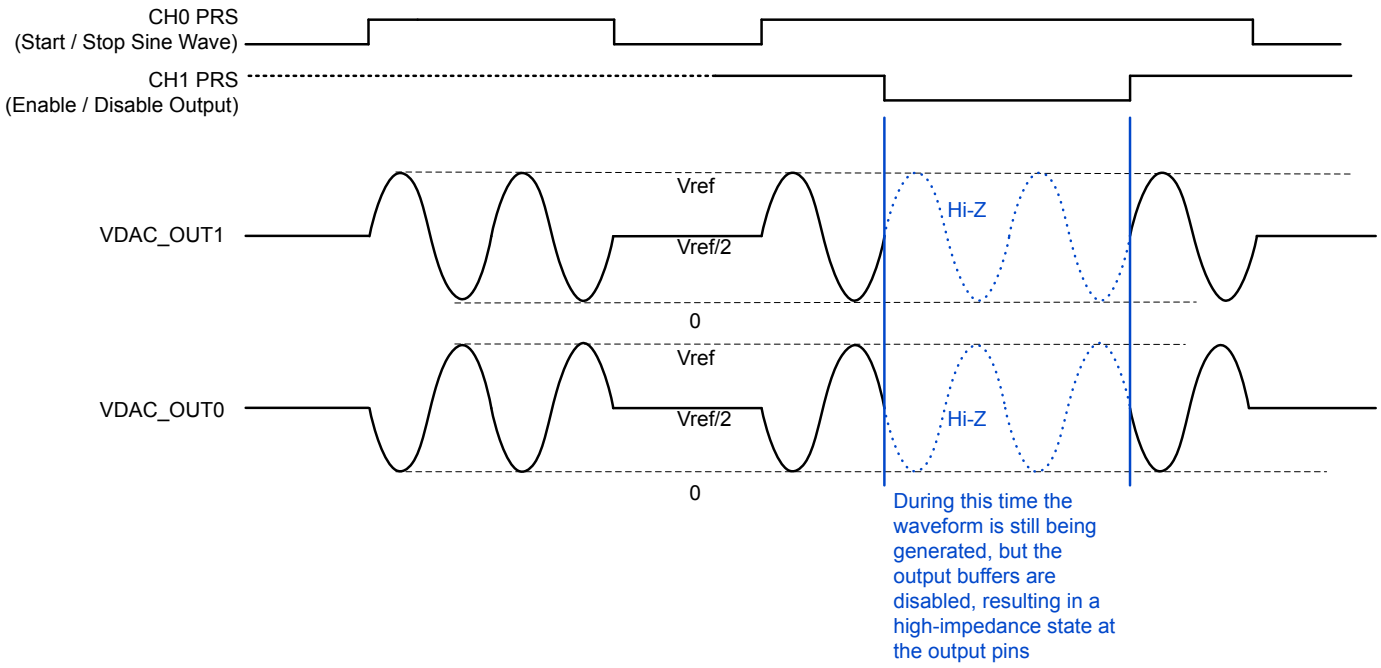

Figure 4.1. VDAC Sine Mode PRS Control

Note: Waveforms shown reflect DIFF = 1 and SINERESET = 1 in VDAC_CFG.

4.3 HFCLKOUT

Specific EFR32xG23 OPNs provide a high-quality, buffered sinusoidal clock output (HFCLKOUT) on a dedicated pin that can be used to clock another device. Most typically another radio might be used to provide network commissioning capability via a Bluetooth connection to a mobile device. This option is not available on any EFR32xG28 OPN.

When such functionality is required, especially to provide a clock to another device with a radio, it must be implemented with circuitry external to the EFR32xG28. In cases where simply having a clock is the benefit versus its specific signal quality, such as using it to drive the clock input of an external microcontroller without wireless capability, the Clock Management Unit (CMU) may be able to drive a suitable clock on a GPIO pin via its Export Clock Control Register (CMU_EXPORTCLKCTRL).

4.4 Liquid Crystal Display (LCD) Expansion

Improvements made to the EFR32xG28 and EFM32PG28 LCD interface relative to EFR32xG23 and EFM32PG23 are covered in Table 4.1 LCD Comparison on page 7. Forward compatibility of software written for xG23 that configures and controls the LCD using the relevant emlib APIs is not impacted by the added SEG and COM lines on xG28. Furthermore, even software for xG23 that writes directly to the LCD_SEGDn registers should not be affected by the added segment data bits on xG28. Because they are not implemented and thus reserved for future expansion, they should, by convention, be written to 0s on xG23.

Table 4.1. LCD Comparison

	EFR32xG28 and EFM32PG28	EFR32xG23 and EFM32PG23
Dedicated SEG Lines	24[1]	20
Dedicated COM Lines	4	4
Multiplexed COM/SEG Lines	4[2]	0
COM Multiplexing Options	1 / 2 / 3 /4	1 / 2 / 3 /4 / 6 / 8
Maximum Number of Segments	4 x 28 = 112 6 x 26 = 156 8 x 24 = 192	4 x 20 = 80

Note:

1. Two of the additional dedicated SEG lines are available on the 48-pin QFN (EFR32xG28 only); the other two are only available on the 68-pin QFN.
2. Only available on the 68-pin QFN.

4.5 Memory

Table 4.2 EFR32xG28 and EFM32PG28 vs. EFR32xG23 and EFM32PG23 Flash on page 8 compares flash on the two devices. While erase cycle endurance and data retention specifications remain unchanged, word programming and page erase times may vary slightly between the two devices. Mass erase time is increased because twice the flash must be erased on xG28.

Table 4.2. EFR32xG28 and EFM32PG28 vs. EFR32xG23 and EFM32PG23 Flash

	EFR32xG28 and EFM32PG28	EFR32xG23 and EFM32PG23
Flash Size (kB)	1024 / 512[1]	512 / 256 / 128
Erase Page Size (kB)	8	8
Number of Erase Pages	128 / 64[1]	64 / 32 / 16
Page Lock Registers (MSC_PAGELOCKn)	4	2
Starting Address	0x08000000	0x08000000

Note:
1. Variants with 512 kB flash available on EFM32PG28 only.

A similar comparison is provided for RAM in Table 4.3 EFR32xG28 and EFM32PG28 vs. EFR32xG23 and EFM32PG23 RAM on page 8. The increased number of retention blocks results from a difference in the encoding of the RAMRETNCTRL field in the DMEM0 Retention Control Register (SYSCFG_DMEM0RETNCTRL) on xG28. Use `EMU_RamPowerDown()`, which takes a user-designated RAM address range, to avoid potential errors this change could cause.

Table 4.3. EFR32xG28 and EFM32PG28 vs. EFR32xG23 and EFM32PG23 RAM

	EFR32xG28 and EFM32PG28	EFR32xG23 and EFM32PG23
Main RAM Size (kB)	256 / 128	64 / 32
RAM Retention Block Size (kB)	16	16
Number of Retention Blocks	16 / 8	4 / 2
Starting Address	0x20000000	0x20000000

The starting addresses of flash and RAM are the same on both devices, which means that the bootloader starting address is also the same. The User Data and Device Information blocks remain the same size and reside at same addresses on both devices. Firmware projects that use a custom linker file will require changes to account for the increase in flash and RAM sizes.

4.6 SYSCFG Revision Information Encoding

While the SYSCFG block is primarily focused on memory-related functionality, such as ECC, it does provide registers that can be used for device identification. Therefore, it is important to note that the bit fields present in the two chip revision registers (SYSCFG_CHIPREVHW and SYSCFG_CHIPREV) have changed on EFR32xG28 and EFM32PG28. Both registers still hold the MINOR and MAJOR bit fields, but the width of these has changed. Furthermore, the 6-bit wide FAMILY field in these registers on EFR32xG23 and EFM32PG23 has been replaced with a 12-bit wide PARTNUMBER field on xG28.

Thus, on current versions of EFR32xG23 and EFM32PG23, SYSCFG_CHIPREVHW shows…

	Bit Position																															
	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													0x2								0x38						0x1					
Access													RW								RW						RW					
Name													MINOR								FAMILY						MAJOR					

…whereas EFR32xG28 and EFM32PG28 would show:

Offset	Bit Position																															
0x014	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													0x1				0x0				0x11											
Access													R				R				RW											
Name													MAJOR				MINOR				PARTNUMBER											

Because of the relocation of the MAJOR and MINOR bit fields and the change from the FAMILY bit field to the PARTNUMBER bit field, firmware should avoid reading these registers directly to identify the device on which it is executing. Instead, use a relevant emlib call, such as `SYSTEM_ChipRevisionGet()`, to retrieve and decode this information.

Note: Regardless of the device, SYSCFG_CHIPREV is not hardwired but is updated during reset and may show a slightly different value than SYSCFG_CHIPREVHW.

4.7 Trace Clock Enhancements

EFR32xG23 and EFM32PG23 is limited to using SYSCLK divided-by-1, 2, or 4 as the clock for the Trace Port Interface Unit (TPIU), which formats data driven on the external trace interface. Source clock selection and prescaling improvements have been made to EFR32xG28 and EFM32PG28 and are shown in Table 4.4 CMU Debug Trace Clock Control (CMU_TRACECLKCTRL) Register on page 10.

Table 4.4. CMU Debug Trace Clock Control (CMU_TRACECLKCTRL) Register

Offset	Bit Position																													
0x080	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																											0x0				0x1	
Access																											RW				RW	
Name																											PRESC				CLKSEL	

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated.*		
5:4	PRESC	0x0	RW	**TRACECLK Prescaler**
	Clock prescaler for debug trace logic.			
	Value	Mode		Description
	0	DIV1		TRACECLK source is divided by 1
	1	DIV2		TRACECLK source is divided by 2
	2	**DIV3**		**TRACECLK source is divided by 3 (new on xG28)**
	3	DIV4		TRACECLK source is divided by 4
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated.*		
1:0	**CLKSEL**	**0x1**	**RW**	**Clock Select (new on xG28)**
	Selects the TPIU clock source. Changing this while trace is enabled results in bus fault.			
	Value	Mode		Description
	0	DISABLE		TRACE clock disabled
	1	SYSCLK		SYSCLK drives TRACE
	2	HFRCOEM23		HFRCOEM23 drives TRACE
	3	HFRCODPLLRT		HFRCODPLLRT drives TRACE

5. Software Compatibility

A high degree of software compatibility is retained between EFR32xG23 and EFM32PG23 and EFR32xG28 and EFM32PG28. However, in all cases, firmware written originally for xG23 must be recompiled against the patch release of a new Gecko SDK that supports xG28. Even though software compatibility is highly retained between the two device families, the recompiled firmware cannot be flashed interchangeably between EFR32xG23 and EFM32PG23 and EFR32xG28 and EFM32PG28.

5.1 EMLIB

Because emlib is a peripheral abstraction API, it handles the small differences between devices within the EFR32 families, including specific revisions. Certain emlib subcomponents are impacted by the differences between EFR32xG23 and EFM32PG23 and EFR32xG28 and EFM32PG28.

The following sections provide a summary of certain software differences between xG23 and xG28. Please note that this is not necessarily a complete list of the differences, and users are advised to consult the Gecko SDK and its relevant documentation for additional details.

5.1.1 `em_acmp.c` and `em_acmp.h`

The addition of bias modes 0 and 1 to the ACMPs on EFR32xG28 and EFM32PG28 does not change the underlying abstraction code in `em_acmp.c` and `em_acmp.h`. Because the `biasProg` member of the existing `ACMP_Init_TypeDef` structure is of type `uint32_t`, initialization firmware migrated from EFR32xG23 and EFM32PG23 can simply specify a value of 0 or 1 to use the new bias modes. If current draw and comparator response time, especially in low-energy modes, is acceptable on xG28 with the `biasProg` setting used in existing xG23 firmware, no changes are necessary.

As noted in 4.1 Analog Comparator (ACMP), a longer delay is required than what is provided by the ACMP self-timed warm-up mechanism when using bias modes 0 - 3 on EFR32xG28 and EFM32PG28. Firmware migrated from EFR32xG23 and EFM32PG23 must take this into consideration and add an appropriate software delay, e.g. by using Sleeptimer.

5.1.2 `em_lcd.c` and `em_lcd.h`

As discussed in 4.4 Liquid Crystal Display (LCD) Expansion, forward compatibility of LCD firmware written for EFR32xG23 and EFM32PG23 is preserved on EFR32xG28 and EFM32PG28 despite the addition of extra dedicated SEG and multiplexed COM/SEG lines so long as said firmware follows convention by using emlib and/or writing 0s to unimplemented LCD_SEGDn register bits.

If migration to xG28 entails use of a display with additional COM lines, the `LCD_Mux_TypeDef` structure (used in `LCD_Init_TypeDef` structure) includes support for the new sextaplex and octaplex muxing options:

```
typedef enum {
  lcdMuxStatic     = LCD_DISPCTRL_MUX_STATIC,
  lcdMuxDuplex     = LCD_DISPCTRL_MUX_DUPLEX,
  lcdMuxTriplex    = LCD_DISPCTRL_MUX_TRIPLEX,
  lcdMuxQuadruplex = LCD_DISPCTRL_MUX_QUADRUPLEX,
  lcdMuxSextaplex  = LCD_DISPCTRL_MUX_SEXTAPLEX,
  lcdMuxOctaplex   = LCD_DISPCTRL_MUX_OCTAPLEX,
} LCD_Mux_TypeDef;
```

Similarly, should an upgrade to xG28 include a display with additional segments, these can be manipulated through existing emlib LCD APIs, such as `LCD_SegmentEnable()`, `LCD_SegmentSet()`, and `LCD_SegmentSetLow()`.

5.1.3 `em_prs.c` and `em_prs.h`

No changes were made to `em_prs.h` or `em_prs.c` to accommodate VDAC sine wave generator PRS control on EFR32xG28 and EFM32PG28. This is because the VDAC's existing PRS consumer inputs are repurposed, such that…

- CH0 becomes the sine wave generator reset input (equivalent to SINEMODESTOP in VDACn_CMD) when SINEMODEPRS = 1 and…
- CH1 becomes the output buffer enable (equivalent to CH0DIS and CH1DIS in VDAC_CMD) when OUTENPRS = 1.

Likewise, for the reasons above, the functionality of the PRS_CONSUMER_VDAC0_ASYNCTRIGCHn registers also remains the same.

5.1.4 `em_system.c` and `em_system.h`

So that `SYSTEM_GetFamily()` can return correct values for EFM32PG28, EFR32FG28, and EFR32ZG28, entries have been added to the `SYSTEM_PartFamily_TypeDef` enumeration in `em_system.h`:

```
#if defined(_SILICON_LABS_32B_SERIES_2_CONFIG_8)
  /**< EFR32 Flex Gecko Series 2 Config 8 Value Device Family */
  systemPartFamilyFlex28 = DEVINFO_PART_FAMILY_FG | (28 << _DEVINFO_PART_FAMILYNUM_SHIFT),
  /**< EFR32 Zen Gecko Series 2 Config 8 Value Device Family */
  systemPartFamilyZen28 = DEVINFO_PART_FAMILY_ZG | (28 <<  _DEVINFO_PART_FAMILYNUM_SHIFT),
  /**< EFR32 Pearl Gecko Series 2 Config 8 Value Device Family */
  systemPartFamilyPearl28 = DEVINFO_PART_FAMILY_PG | (28 <<  _DEVINFO_PART_FAMILYNUM_SHIFT),
#endif
```

Note that this information is just an encoding of the FAMILY and FAMILYNUM fields stored in the DEVINFO_PART register and can be read directly from the DEVINFO region of flash, if desired.

Firmware should use `SYSTEM_ChipRevisionGet()` to determine the specific chip revision level — such as might such as might be needed to determine whether or not to execute a particular erratum workaround — because it has been updated to handle the changes discussed in 4.6 SYSCFG Revision Information Encoding.

5.1.5 `em_vdac.c` and `em_vdac.h`

The addition of PRS controls to the VDAC sine wave generator on EFR32xG28 and EFM32PG28 is accommodated by the `VDAC_Init_TypeDef` in `em_vdac.h` and the `VDAC_Init()` function in `em_vdac.c`. The initialization structure's new PRS settings are shown in bold below:

```
typedef struct {
  uint32_t                   warmupTime;
  bool                       dbgHalt;
  bool                       onDemandClk;
  bool                       dmaWakeUp;
  bool                       biasKeepWarm;
  VDAC_Refresh_TypeDef       refresh;
  VDAC_TimerOverflow_TypeDef timerOverflow;
  uint32_t                   prescaler;
  VDAC_Ref_TypeDef           reference;
  bool                       ch0ResetPre;
  bool                       sineReset;
  bool                       sineEnable;
  bool                       diff;
  bool                       sineModePrsEnable;
  bool                       prsOutEnable;
} VDAC_Init_TypeDef;
```

As would be expected for backwards compatibility, these new settings are disabled in the `VDAC_INIT_DEFAULT` initializer:

```
#define VDAC_INIT_DEFAULT                                                                \
  {                                                                                      \
    _VDAC_CFG_WARMUPTIME_DEFAULT, /* Number of prescaled DAC_CLK for Vdac to warmup. */  \
    false,                        /* Continue while debugging. */                      \
    true,                         /* On demand clock. */                               \
    false,                        /* DMA wake up. */                                   \
    false,                        /* Bias keep warm. */                                \
    vdacRefresh8,                 /* Refresh every 8th cycle. */                       \
    vdacCycles2,                  /* Internal overflow every 8th cycle. */             \
    0,                            /* No prescaling. */                                 \
    vdacRef1V25,                  /* 1.25 V internal low noise reference. */           \
    false,                        /* Do not reset prescaler on CH 0 start. */          \
    false,                        /* Sine wave is stopped at the sample its currently outputting. */ \
    false,                        /* Disable sine mode. */                             \
    false,                        /* Differential mode. */                             \
    false,                        /* PRS controlled sinemode. */                       \
    false,                        /* PRS controlled output enable. */                  \
  }
```

No changes have been made to the calling convention of `VDAC_Init()` to accommodate the sine wave generator PRS control functionality nor have other emlib VDAC APIs changed. Barring any modifications related to device pin out, such as moving from a smaller xG23 package to a larger xG28 package, existing VDAC firmware can be migrated as-is if these new features are not needed.

5.2 NVM3

The functionality of NVM3 is unchanged on EFR32xG28 and EFM32PG28. Consequently, there is no need to change the number of flash pages allocated to NVM3 in any given application. In cases where application functionality is added when upgrading to xG28, the larger flash permits the allocation of extra flash pages to NVM3. This adjustment is made in the NVM3 Default Instance component of the firmware's Simplicity Studio project.

6. Revision History

Revision 0.4

August, 2023

- Added note in 3. External Hardware Migration regarding additional DVDD pin on EFM32PG28 QFN68 package.

Revision 0.3

March, 2023

- Updated 1. Introduction wording.
- Updated 1.1 Pins and External Hardware to reflect EFM32PG23 and EFM32PG28 pin incomabitibility.
- Removed EFM32PG23/EFM32PG28 Pin Compatibility Overview table, as these devices are not pin compatible.
- Fixed typos in 3. External Hardware Migration.
- Fixed typo in 4. Peripheral Compatibility.
- Fixed typo in 4.1 Analog Comparator (ACMP).
- Fixed typo in 4.2 Digital-to-Analog Converter (VDAC) PRS Sine Wave Control.
- Fixed typo in 4.3 HFCLKOUT.
- Fixed typo in 4.4 Liquid Crystal Display (LCD) Expansion.
- Fixed Note 1 in Table 4.1 Table 4.1 LCD Comparison on page 7.
- Added Note 1 in Table 4.2 Table 4.2 EFR32xG28 and EFM32PG28 vs. EFR32xG23 and EFM32PG23 Flash on page 8.
- Fixed missing information in Table 4.3 Table 4.3 EFR32xG28 and EFM32PG28 vs. EFR32xG23 and EFM32PG23 RAM on page 8.
- Fixed typo in 5. Software Compatibility.
- Added EFM32PG28 to 5.1.4 `em_system.c` and `em_system.h`.

Revision 0.2

March, 2023

- Revised the title to add EFM32PG23 and EFM32PG28 support.
- Added EFM32PG23 and EFM32PG28 support.
- Added EFM32PG23/EFM32PG28 Pin Compatiblity Overview table.
- Renamed tables Table 4.2 EFR32xG28 and EFM32PG28 vs. EFR32xG23 and EFM32PG23 Flash on page 8 andTable 4.3 EFR32xG28 and EFM32PG28 vs. EFR32xG23 and EFM32PG23 RAM on page 8.

Revision 0.1

December, 2022

- Initial revision.

Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support & Community
www.silabs.com/community

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

www.silabs.com