UG0890
User Guide
PolarFire SoC FPGA Power-Up and Resets

Microsemi Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Learn more at www.microsemi.com.

50200890. 1.0 12/19

Contents

Figures

Tables

1 Acronyms

The following acronyms are used in this document.

Table 1 • **List of Acronyms**

Acronym	Expanded
AMBA	ARM Advanced Microcontroller Bus Architecture
eNVM	embedded Non-Volatile Memory
MSS	Microprocessor Subsystem
POR	Power on Reset
SCB	System Controller Bridge
sNVM	Secure Non-volatile Memory
HSIO	High-speed IO
GPIO	General Purpose IO
PLL	Phase-locked loop
DLL	Delay-locked loop
FIC	Fabric Interface Controller
PCIe	Peripheral Component Interconnect Express

2 Power-Up

PolarFire SoC FPGAs use advanced power-up circuitry to ensure reliable power-up. When the device is powered on, the Power-on Reset (POR) circuitry and the System Controller ensure a systematic POR. System Controller is responsible for device boot and design initialization.

The entire process of powering up the device includes the following sequential steps:

- Power-On
- Device Boot
- Design and Memory Initialization
- MSS Pre-Boot
- MSS User-Boot

Note: For information about MSS Pre Boot and MSS User Boot, see *UG0881: PolarFire SoC FPGA Booting And Configuration User Guide*.

2.1 Power-On

When the device is power cycled, the POR circuitry detects voltage ramp-up on the V_{DD18}, V_{DD25}, and V_{DD} power supply rails using voltage detectors. For a list of power supplies, see Appendix: Power Supplies, page 12. The System Controller remains in the reset state until the required voltage threshold levels are reached. The System Controller is responsible enabling, or turning on the FPGA fabric and related I/Os.

The voltage detectors in the PolarFire SoC devices are calibrated with a high-level of accuracy to ensure reliable monitoring of minimum threshold levels. For power-supply threshold voltage levels to release POR, see the “Power-on Reset Voltages” section in *PolarFire SoC FPGA Advance Datasheet*. The device boot starts after a delay after the voltage supply rails reach their respective threshold voltage levels.

In PolarFire SoC devices, there are separate voltage detectors to monitor I/O bank supplies. During POR, the serial transceivers and the fabric are powered down, and HSIO/GPIO banks are tri-stated.

For more information on power supply sequencing requirements and recommendations, see the “**Core Power Supply Operations”** section**,** in *UG0901: PolarFire SoC Board Design Guidelines User Guide*.

2.2 Device Boot

After POR circuitry releases the System Controller from reset, the device boot-up procedure is executed by the System Controller. The System Controller always executes the same device boot-up sequence irrespective of the user design.

The following events occur during device boot-up:

- sNVM is powered up and enabled for normal operation.
- Transceiver I/Os are tri-stated.
- Termination can be enabled with programming bits to enable early PCI Express Receiver Detection.
- Fabric is powered up and enabled.
- HSIO and GPIO banks are configured based on the user configuration in Libero SoC.
- MSS is powered down and MSSIOs are tri-stated.

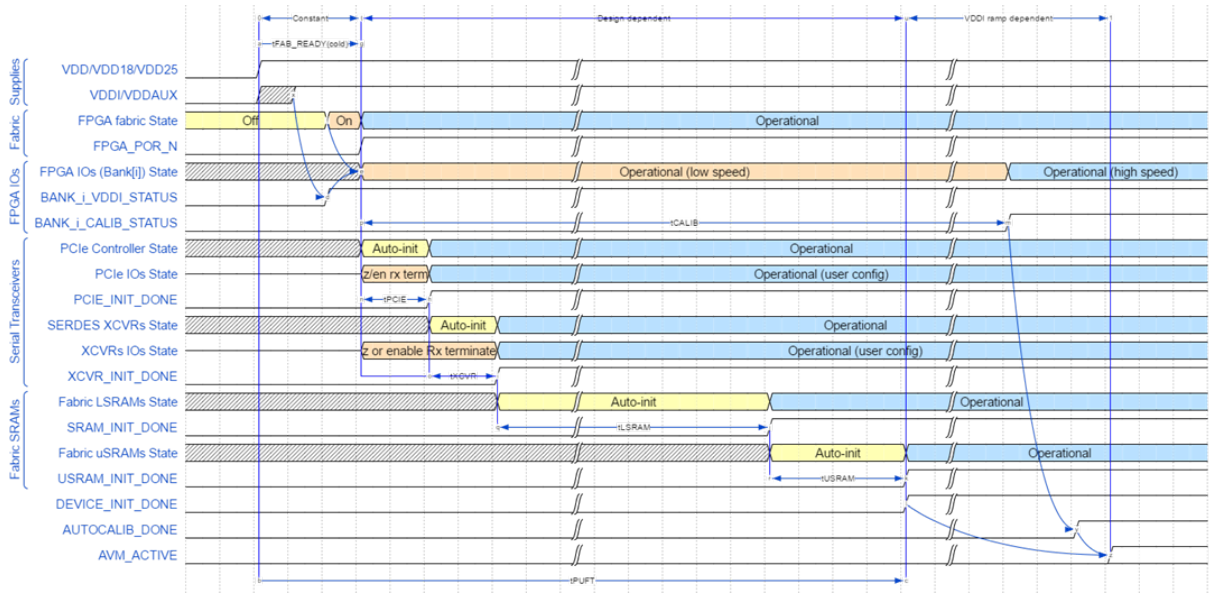
2.3 Design and Memory Initialization

The state of the RAM blocks (LSRAMs and µSRAMs) is indeterminate after power-up. In PolarFire SoC devices, the RAM blocks can be initialized with known values, if desired. PCIe and XCVR blocks used in the design, are initialized with the user configuration data at power-up. The System Controller performs the design and memory initialization during the power up sequence. The memory initialization data can be stored in µPROM, sNVM, or an external SPI flash. The storage location of the initialization data is selected during the Libero design flow. The initialization data can be encrypted for storing in external SPI Flash.

The following figure shows the sequence in which the fabric, PCIe, Transceiver, LSRAMs, and µSRAMs are automatically initialized. The sequence is customized depending on the resources instantiated in the user design. For example, the PCIE_INIT_DONE will not assert if the user design does not contain PCIe. As a result, the sequence skips the PCIe initialization and moves to the next step.

The user can monitor the design initialization status using the PolarFire SoC Initialization Monitor.

Figure 1 • **Power-up To Functional Time**



Note: PolarFire SoC devices have built-in tamper detection features to monitor voltage supplies and flags to detect minimum or maximum threshold values. These flags are valid only after design initialization, and not during POR.

2.3.1 How To Set Up Design and Memory Initialization

To be updated

2.4 HSIO/GPIO Bank Initialization

Unused GPIO and HSIO banks can be left powered down or powered up. During the device power-up, power up the used GPIO and HSIO banks simultaneously along with all the other power supplies. All of the banks are initialized automatically with flash configuration bits when fabric is powered up.

The time at which I/Os are functional depends on a combination of the following:

- Device boot
- Ramp-up time of the power applied to the I/O banks
- Calibration time of the high-speed I/Os (For example, DDR interfaces)

For low-speed operations below 400 MHz, I/Os are functional after the power applied to I/O banks exceeds the threshold levels. If the user requires the I/Os to be usable immediately upon completion of device boot, the I/O power ramp time must beset to recommended minimum value. The user can also apply slow or delayed I/O ramp times on I/O banks to delay the time until which the I/Os are usable. If the user is applying slow ramp rates to I/O banks, the user logic in the fabric must monitor the state of the I/O banks to know when they are usable as low-speed I/Os.

If the I/Os need to be operated for high-speed (for example, DDR interfaces), another step of I/O calibration is necessary.

The high-speed I/O calibration process occurs automatically. The user's design in the FPGA fabric, must not use these IOs until IO calibration has completed. The status of the I/O calibration and bank power supply can be monitored using the status signals of the PolarFire SoC Initialization Monitor IP.

The PolarFire SoC Initialization Monitor asserts BANK_#_CALIB_STATUS and BANK_#_VDDI_STATUS signals to the fabric. BANK_#_CALIB_STATUS can be used by the user logic to determine if the calibration completes for each I/O bank. BANK_#_VDDI_STATUS signal can be used to monitor VDDI supply on specific I/O banks.

The DRI clock (DRI_CLK) must be gated off until the assertion of DEVICE_INIT_DONE, which asserts at the end of the complete device initialization.

2.5 Transceiver Initialization

Transceiver power-up depends on $V_{DDA,}$ VDDA25, and $V_{DD\_XCVR\_CLK}$. $V_{DD\_XCVR\_CLK}$ is applicable if an external reference clock is used for transceivers. For a list of power supplies, see Appendix: Power Supplies, page 12. Glitches can occur in the reference clocks and the data bits during power-up.

The transceiver is initialized by the flash configuration bits and the design initialization client.

When XCVR_INIT_DONE/DEVICE_INIT_DONE from PF_INIT_MONITOR goes high, the transceiver is completely configured. The user logic using the XCVR clock must be held in reset until the XCVR_INIT_DONE signal is asserted.

The transceiver data pins are in hot-plug mode at power-up. Programming bits can be used to enable TX and/or RX termination early to enable fast Receiver Detection in standards such as PCI Express.

2.6 User PLLs and DLLs Initialization

Both PLLs and DLLs are initialized automatically with flash configuration bits when fabric is powered up.

2.7 PCIe Initialization

To achieve the PCIe initialization requirement, the physical layer is configured using flash configuration bits. The remainder of the configuration is done during design initialization with the user data stored in the non-volatile memory.

For more information about PCIe initialization process, see *UG0685: PolarFire FPGA PCI Express User Guide*.

2.8 State of Blocks During Power-Up

The following table shows the state of different blocks during device power-up.

Table 2 • **Default State During Device Power-Up**

Block	POR	Device Boot	Design and Memory Initialization State
System Controller	Held in reset	Executing boot-up sequence	Performs design and memory initialization
sNVM	Held in reset	Power up sequence, then functional	Functional
FPGA fabric array	Powered down	Power up sequence, then functional	Functional
LSRAM	Powered down	Powered up, uninitialized	Initialized with user data if configured
µSRAM	Powered down	Powered up, uninitialized	Initialized with user data if configured
µPROM	Powered down	Powered up	Functional
Math block	Powered down	Powered up	Functional
Transceiver and TX PLLs	Powered down	Powered up but not functional Termination Optionally Enabled	Initialized with user data and functional
GPIO/HSIO - Low Speed (if power is applied)	Input buffers are disabled and output buffers are tri-stated. GPIO buffers are in hot-plug mode.	Powered up but not usable GPIO buffers are in hot-plug mode. HSIO buffers do not support hot-plug capability	Functional if I/O and I/O Auxiliary Supplies supply exceeds threshold
GPIO/HSIO - High-speed (if power is applied)	Input buffers are disabled and output buffers are tri-stated GPIO buffers are in hot-plug mode.	Powered up but not usable GPIO buffers are in hot-plug mode. HSIO buffers do not support hot-plug capability	Functional at high-speed after the completion of I/O calibration if I/O and I/O Auxiliary Supplies are applied
PCIe	Powered down	Powered up but not functional	Initialized with user data and functional
Transceiver I/O	Tri-stated and hot-plug mode	Tri-stated in hot-plug mode Termination optionally enabled	Termination Enabled, operational
MSSIOs	Tri-stated	Tri-stated	Tri-stated
MSS	Powered down	Powered down	Powered down

Note: For more information about cold boot and warm boot power-up to functional time, see the “Power-Up to Functional Timing” section in *PolarFire SoC FPGA Advance Datasheet*.

3 Resets

The fabric flip-flops powers-up in an intermediate state. The reset logic should be included in the design for proper functioning.

3.1 User Resets

3.1.1 Device Reset Pad (DEVRST_N)

DEVRST_N or device reset, is powered through the dedicated I/O bank. The DEVRST_N assertion results in full re-initialization of the device, including the loading of user configuration data to PCIe, transceivers, and the re-initialization of MSS, fabric LSRAMs and µSRAMs.

For designing a robust system, users may use the dedicated DEVRST_N pin or a general purpose reset signal using any GPIO/HSIO as a global system reset. For the following cases, the users must use the DEVRST_N as a warm reset for the device:

- User design modifies auto-initialized fabric RAMs or PCIe configuration during operation.
- User design is using PCIe, transceivers or UserCrypto.

For all other use cases, it is recommended to use a general purpose reset signal using any GPIO/HSIO I/O because they take much shorter time for design to come out of reset.

If the dedicated DEVRST_N is not used for warm resets, the DEVRST_N pin must be configured using one of the following methods:

- Drive the signal with a POR chip or an external device and keep the DEVRST_N asserted till the system/clocks are stable and the chip is properly powered up.
- Connect DEVRST_N to VDDI3 through a 1 kΩ ohm resistor per pin without sharing with any other pins.
    - In this case, the user needs to ensure that all clocks going to the device are stable before the user design is released from power-on reset. The details of the minimum time taken for the fabric design to be activated after power-on is specified in the Power-Up To Functional section of *PolarFire SoC FPGA Advance Datasheet*.

3.1.2 Device Reset from Fabric

The tr_reset_device signal from the fabric may be pulsed high by user logic in order to initiate a full device reset and re-boot. The tr_reset_device signal is a pre-configured response to a detected tamper event in the fabric. It can also be used as a device reset from user logic in the fabric. In this case, the signal should be pulsed low for a short duration. For information about pulse duration, *PolarFire SoC FPGA Advance Datasheet*.

Assertion of DEVRST_N or device reset initiated from the fabric triggers the System Controller to power down the device in the following sequence:

1. The reset signal propagates as a non-maskable interrupt to the System Controller, which first disables all I/Os.
2. Starts the Watchdog timer to schedule an device reset
3. The fabric gets powered down.

Resets are issued to all peripherals, such as MSS, Fabric, transceivers, PCIe, PLLs, and DLLs. For information about peripherals, see *UG0886: PolarFire SoC FPGA Peripherals User Guide*.

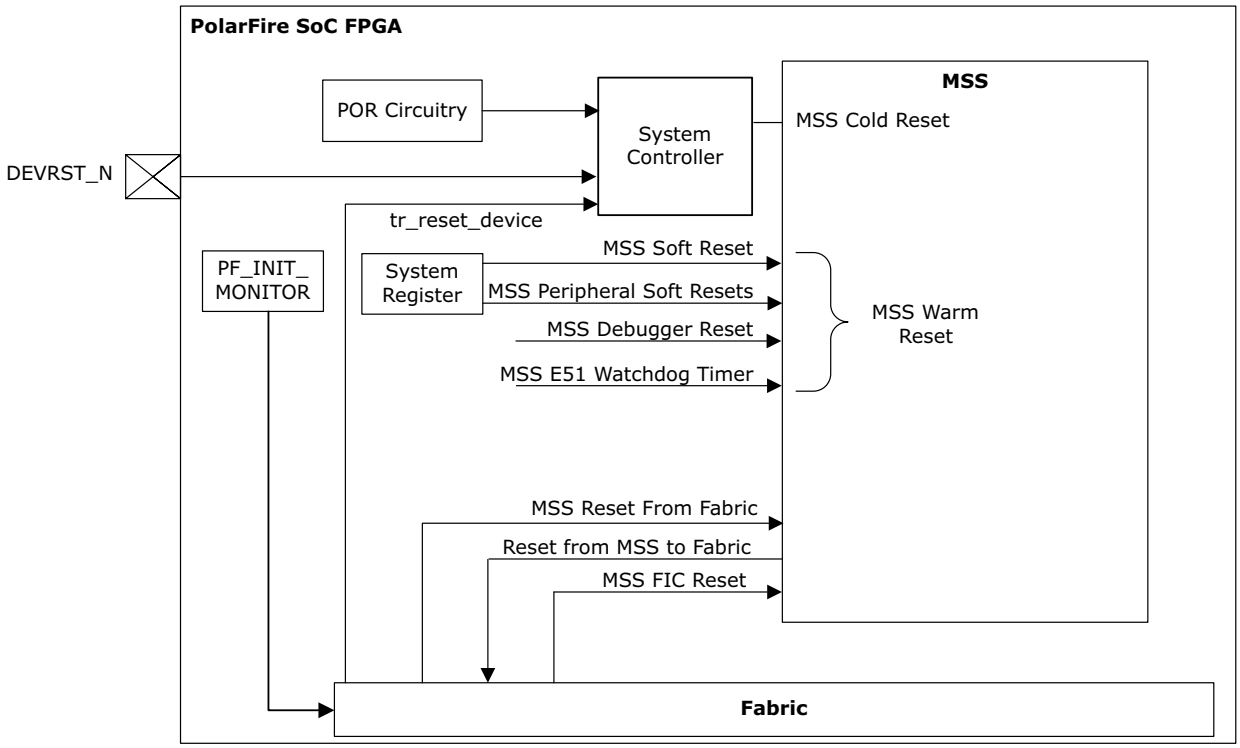
3.2 MSS Resets

After device power-up, the PolarFire SoC System Controller manages the device initialization. Following are the PolarFire SoC resets:

- Microprocessor subsystem (MSS) Cold Reset
- MSS Warm Reset
- MSS FIC Reset
- MSS Peripheral Soft Resets
- User Reset

For information about MSS, see *UG0880: PolarFire SoC MSS User Guide*.

Figure 2 • **Block Diagram of Resets**



3.2.1 MSS Cold Reset

MSS cold reset is initiated by the Power on Reset (POR) circuitry when the device is powered on. MSS cold reset results in resetting of all the functionality within the MSS except the eNVM. eNVM can be reset using SOFT_RESET_CR register.

3.2.2 MSS Warm Reset

Any of the four MSS warm reset provides a method to reset the entire MSS and all the peripherals. This results in the asynchronous resetting of all functionality within the MSS (except the MSSIO configuration, IOMUXes and potentially MSS GPIO peripherals, if configured to be reset by fabric). The MSS internally remains in reset until the warm reset source is removed. When the warm reset signal is removed, an interrupt is generated to the System Controller, to indicate the MSS warm reset release event. After this, the System Controller firmware sequences the MSS back out of reset.

Following are the sources for initiating warm resets of the MSS.

3.2.2.1 MSS Soft Reset

An MSS_RESET_CR soft reset register can be written with a specific value via the Application code in order to fully reset the MSS.

Table 3 • **MSS_RESET_CR**

	Register Name	Type	Default Value	Field Description
31:16	Reserved	RO	0x0000	Reserved
15:0	RESET_VALUE	RW	0x0000	When written, 16'hDEAD causes a full MSS reset. The reset clears this register. The register may be written to any value but only a value off 16'hDEAD causes the reset to happen.

3.2.2.2 MSS Debugger Reset

An MSS debugger reset register bit can be set via the software debugger in order to fully reset the MSS.

3.2.2.3 MSS Reset from Fabric

User logic in the fabric asserts a reset signal to asynchronously reset the MSS. See *PolarFire SoC FPGA Advance Datasheet* for more information.

3.2.2.4 MSS E51 Processor Watchdog Timeout Reset

The MSS can be configured such that the E51 processor's watchdog timer causes a reset of the MSS when the timer runs out.

MSS Reset Reasons

MSS can be reset in various ways as explained in the preceding sections. The user can access the 32-bit register RESET_SR to know which reset caused the MSS to be reset. The following table shows the reason for resetting the MSS.

Table 4 • **Reset Reasons**

Reason	Reset Reason Bit	Asserted By	Notes
SCB_PERIPH_RESET	0	SCB	This is the power on reset. This fully resets the MSS including eNVM trim values. Additional bits in the SOFT-RESET register also allow the SCB registers to be reset.
SCB_MSS_RESET	1	SCB, CPU, MSS	This resets the MSS including the Core Complex, Peripherals and all AXI infrastructure. It does not reset the eNVM trim values and SCB registers
SCB_CPU_RESET	2	SCB, CPU, MSS	This resets the Core Complex only. This reset should be used in most cases as the MSS will require reseting at the same time to clear outstanding AXI transactions and so on
DEBUGER_RESET	3	Debugger	This is asserted by the Core Complex debugger and has the same effect as the SCB_MSS_RESET

Table 4 • **Reset Reasons**

Reason	Reset Reason Bit	Asserted By	Notes
FABRIC_RESET	4	Fabric	This is asserted by the fabric and has the same effect as the SCB_MSS_RESET. This reset is disabled by a system register bit at reset and does not function until enabled.
WDOG_RESET	5	Watchdog	This indicates that the watchdog reset has activated.
GPIO_RESET	6	Fabric	This indicates that the fabric GPIO reset was asserted. It resets the GPIO blocks if the GPIO's are configured to be reset by this signal. It does not reset the MSS
SCB_BUS_RESET	7	Fabric	Indicates that SCB bus reset occurred
CPU_SOFT_RESET	8	CPU	Indicates that the CPU reset the MSS using the soft reset register
Reserved	31:9		Reserved

3.2.3 MSS FIC Reset

Each of the main ARM Advanced Microcontroller Bus Architecture (AMBA) buses between the MSS and the FPGA fabric has an associated reset, which resets that bus interface. In this way, user logic in the fabric may hold the MSS from initiating any activity on these interfaces until its own logic is ready to accept it. For information about FIC, *UG0880: PolarFire SoC MSS User Guide*.

3.2.4 MSS Peripheral Soft Resets

Each MSS peripheral has a soft reset register (SOFT_RESET_CR) bit associated with it in the MSS system registers and this bit must be written to "1" and then "0" to allow the peripheral to be used. When the MSS is reset, all these resets are asserted.

Table 5 • **MSS Peripheral Soft Resets**

ADDR	Register	Field	Bit	Type	Reset value
		ENVM	0	RW	0x0
		MAC0	1	RW	0x1
		MAC1	2	RW	0x1
		MMC	3	RW	0x1
		TIMER	4	RW	0x1
		MMUART0	5	RW	0x1
		MMUART1	6	RW	0x1
		MMUART2	7	RW	0x1
		MMUART3	8	RW	0x1
		MMUART4	9	RW	0x1
		SPI0	10	RW	0x1
		SPI1	11	RW	0x1
		I2C0	12	RW	0x1
		I2C1	13	RW	0x1
		CAN0	14	RW	0x1
x88	SOFT_RESET_CR	CAN1	15	RW	0x1
		USB	16	RW	0x1
		FPGA	17	RW	0x1
		RTC	18	RW	0x1
		QSPI	19	RW	0x1
		MSS_GPIO0	20	RW	0x1
		MSS_GPIO1	21	RW	0x1
		MSS_GPIO2	22	RW	0x1
		DDRC	23	RW	0x1
		FIC0	24	RW	0x1
		FIC1	25	RW	0x1
		FIC2	26	RW	0x1
		FIC3	27	RW	0x1
		ATHENA	28	RW	0x1
		CFM	29	RW	0x1
		Reserved	31:30		

Following is the exception for MSS peripheral soft resets:

MSS GPIO Soft Reset: Each of the three MSS GPIO blocks can be configured to be reset by MSS warm reset or by the MSS GPIO reset signal from the fabric (if the device is programmed).

If configured to use the MSS warm reset (the default configuration), then they are also reset by MSS GPIO soft reset registers in the MSS system registers.

If configured to use the GPIO fabric reset, the MSS GPIO registers state are unaffected by writes to the MSS GPIO soft reset registers. However, these MSS GPIO registers are reset during the handling of the MSS warm reset event by the System Controller firmware.

3.2.5 MSS eNVM Reset

Reset of the eNVM is handled by the System Controller.

3.2.6 Resets from MSS to Fabric

There is a status signal from the MSS to the fabric, which indicates the reset status of MSS. This signal can be used by the fabric logic to hold the data transfers between MSS and Fabric.

3.3 User Reset Generation Scheme

To be updated

4 Appendix: Power Supplies

The following table lists the power supplies.

Table 6 • **Power Supplies in PolarFire Devices**

Power Supply	Description
V_{DD}	For fabric core and transceiver/PCIe blocks.
V_{DD18}	For fabric programming and RC oscillators.
V_{DD25}	For corner phase-locked loop (PLLs) and on-chip non-volatile memory (sNVM).
V_{DDIx}	For I/O banks.
V_{DDAUXx}	For GPIO and HSIO banks.
V_{DDA}	For transceiver.
V_{DDA25}	For transceiver PLLs.
$V_{DD\_XCVR\_CLK}$	For transceiver reference clock input buffers.

5 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most publication.

5.1 Revision 1.0

The first publication of this document.