Libero® SoC v2021.2

PolarFire® System Services Simulation User Guide

Introduction

System services are system controller actions initiated by asynchronous events from the user design via the system controller's design service interface. System services are not accessible when the system controller suspend mode is enabled. Microchip provides PolarFire System Services SgCore to execute the system services on the PolarFire devices. The PolarFire System Services SgCore provides an ARM® Advanced Microcontroller Bus Architecture (AMBA™) Advanced Peripheral Bus (APB) interface for controlling the registers implemented in it. Microchip provides PolarFire System Services SgCore firmware driver with a set of functions for controlling the PolarFire System Services SgCore from a general purpose processor.

Note:
For more information on the PolarFire system services, see UG0848 PolarFire System Services User Guide.

System services are invoked by writing a 16-bit system service descriptor to the system service interface, which triggers a service request to the system controller. The lower seven bits of the descriptor specify the service to be performed and the upper nine bits are used to provide additional information such as the address of a location in the 2 Kbytes mailbox RAM. The mailbox address specifies the location of a service-specific data structure, which is used for any additional input parameters and any outputs from the service. Mailbox addresses are specified using a word offset (0-511).

The following table lists the system service request descriptor bits.

Table 1. System Service Request Descriptor

System Service Descriptor Bit Field	Value	Description
15:7	MBOXADDR[10:2]	Specifies the address in mailbox RAM to access minimum four bytes of memory.
6:0	SERVICEID	Service command for system controller to execute the request.

This document is intended to provide you details on how to run simulation on the PolarFire system services. For more information on PolarFire® system services, see UG0753: PolarFire FPGA Security User Guide.

© 2021 Microchip Technology Inc.
and its subsidiaries

Table of Contents

© 2021 Microchip Technology Inc.
and its subsidiaries

1. Device and Design Information Services

These services return information about the device and current user design. The requested information is copied to a location in the mailbox RAM whose address is included in the service descriptor. The size of the returned data is service dependent. Overflows result in the return data wrapping around the start of the mailbox. These services are available on all devices.

For more information about the return status of device and design information services, see 5. System Service Return Status Codes.

1.1 Serial Number Service

Serial Number Service fetches the 128-bit Device Serial Number (DSN). The DSN is a 128-bit quantity unique to every device, set during manufacturing. It comprises of two parts—the Factory Serial Number (FSN) and the Serial Number Modifier (SNM). The first part of the device serial number is the 64-bit FSN that uniquely identifies a device. The DSN is zeroized if the unrecoverable zeroization action is performed on the device.

You can pass the desired DSN either by passing the value using the `vsim` command or by port mapping during the instantiation.

```
vsim -L PolarFire -L presynth -L CORESYSSERVICES_PF_LIB -gSRLNUM=128'h12345678 -t 1ps -
gSIM_PA5M300T=0 presynth.ss_tb
```

Table 1-1. Serial Number Service Request

System Service Descriptor Bit Field	Value	Description
15:7	MBOXADDR[10:2]	Mailbox address. See Table 1-2
6:0	00H	Serial number service command

The following table lists the Serial Number Service mailbox format.

Table 1-2. Serial Number Service Mailbox Format

Offset	Length (bytes)	Parameter	Direction	Description
0	16	DSN	Output	Device Serial Number

For more information about Device Serial Number, see UG0753: PolarFire FPGA Security User Guide.

Simulation Log

```
SysServices: Device Serial Number service request received at time             11450000 ps.
# Serial Number 0x00000000000000000000000123456789 is being written to Mailbox Address 0x000.
```

1.2 USERCODE Service

USERCODE service fetches the 32-bit USERCODE/Silicon signature. You can pass the desired USERCODE either by passing the value using the `vsim` command or by port mapping during instantiation.

```
vsim -L PolarFire -L presynth -L CORESYSSERVICES_PF_LIB -gUSRCD =32'h1234 -t 1ps -
gSIM_PA5M300T=0 presynth.ss_tb
```

Table 1-3. USERCODE Service Request

System Service Descriptor Bit Field	Value	Description
15:7	MBOXADDR[10:2]	Mailbox address. See Table 1-4.
6:0	01H	USERCODE service command

The following table lists the USERCODE Service mailbox format.

Table 1-4. USERCODE Service Mailbox Format

Offset	Length (bytes)	Parameter	Direction	Description
0	4	USERCODE	Output	Device USERCODE

Simulation Log

```
SysServices: User Code service request received at time             18050000 ps.
#               User Code 0x00000000 is being written to Mailbox Address 0x000
```

1.3 Design Information Service

Design Information service fetches design information including 256-bit user defined design ID, 16-bit design version, and 16-bit design back-level protection value. You can pass the desired DSGNINFO either by passing the value using the `vsim` command or by port mapping during instantiation.

```
vsim -L PolarFire -L presynth -L CORESYSSERVICES_PF_LIB -gDSGNINFO =288'h1234 -t 1ps -
gSIM_PA5M300T=0 presynth.ss_tb
```

Table 1-5. Design Information Service Request

System Service Descriptor Bit Field	Value	Description
15:7	MBOXADDR[10:2]	Mailbox address. See Table 1-6.
6:0	02H	Design Information service command

The following table lists the Design Information Service mailbox format.

Table 1-6. Design Information Service Mailbox Format

Offset	Length (bytes)	Parameter	Direction	Description
0	32	DESIGNID	Output	256-bit user-defined design ID
32	2	DESIGNVER	Output	16-bit design version (DCV)
34	2	BACKLEVEL	Output	16-bit design back-level (DCB)

Simulation Log

```
Services: Design INFO service request received at time             73970000 ps.
#               Design Id 0x000000000000000000000000000000000000000000000000000000000001234
is being written to Mailbox Address 0x000.
#               Design Version 0x0000 is being written to Mailbox Address 0x00000020.
#               Back Level 0x0000 is being written to Mailbox Address 0x00000022.
#                 76210             Device Version Services Completed
```

1.4 Device Certificate Service

Device Certificate service fetches the device's Supply Chain Assurance Certificate from pNVM.
Note: The certificate data is stored as a 1024-bit entity but the actual certificate size might be smaller. Any excess bytes must be discarded.

The device validates the certificate by checking its signature using the Microchip Public Key (MCPK). In addition,

- The certificate DSN is checked against the DSN.
- The certificate public key is checked by recalculating the value using factory key and comparing against the certificate.

Note: In the event of an error, the certificate content is still returned for inspection.

You can pass the desired CERT either by passing the value using the `vsim` command or by port mapping during instantiation.

```
vsim -L PolarFire -L presynth -L CORESYSSERVICES_PF_LIB -gCERT=288'h1234 -t 1ps -
gSIM_PA5M300T=0 presynth.ss_tb
```

Table 1-7. Design Certificate Service Request

System Service Descriptor Bit Field	Value	Description
15:7	MBOXADDR[10:2]	Mailbox address. See Table 1-8.
6:0	03H	Design certificate service command.

The following table lists the Design Certificate Service mailbox format.

Table 1-8. Design Certificate Service Mailbox Format

Offset	Length (bytes)	Parameter	Direction	Description
0	1024	CERTIFICATE	Output	Device certificate

Simulation Log

```
SysServices: Device Certificate service request received at time             13370000 ps.
# Design Certificate
0xxxxxxxxxxxxxxxx000000000000000000000000000000000000000000000000000000000000000000001234 is
being written to Mailbox Address 0x000.
# 69950 Device Certificate Service Completed
```

1.5 Read Digests Service

Read Digests service returns the stored digests for the device. You can pass the desired RDDGST either by passing the value using the `vsim` command or by port mapping during instantiation.

```
vsim -L PolarFire -L presynth -L CORESYSSERVICES_PF_LIB -gRDDGS=3328'h1234 -t 1ps -
gSIM_PA5M300T=0 presynth.ss_tb
```

Table 1-9. Read Digest Service Request

System Service Descriptor Bit Field	Value	Description
15:7	MBOXADDR[10:2]	Mailbox address. See Table 1-10.
6:0	04H	Read digests service command.

The following table lists the Read Digest Service mailbox format.

© 2021 Microchip Technology Inc.
and its subsidiaries

Table 1-10. Read Digests Service Mailbox Format

Offset	Length (bytes)	Parameter	Direction	Description
0	416	DIGESTS	Output	Digest array

Table 1-11. Returned Digests Format

Offset (byte)	Size (bytes)	Value	Description
0	32	FD	Fabric digest
32	32	CCDIGEST	Fabric configuration data digest
64	32	SNVMDIGEST	sNVM ROM pages digest
96	32	ULDIGEST	User security segment digest
128	32	UKDIGEST0	Digest of user key segment containing SRAM-PUF data
160	32	UKDIGEST1	Digest of user key segment containing UEK (User EC key)
192	32	UKDIGEST2	Digest of user key segment containing UPK1
224	32	UKDIGEST3	Digest of user key segment containing UEK1
256	32	UKDIGEST4	Digest of user key segment containing DPK
288	32	UKDIGEST5	Digest of user key segment containing UPK2
320	32	UKDIGEST6	Digest of user key segment containing UEK2
352	32	UPDIGEST	Digest of permanent lock security segments
384	32	FDIGEST	Digest of factory lock segment, factory key segment in pNVM, and System Controller ROM

Simulation Log

```
SysServices: Read Digest service request received at time             79010000 ps.
# Fabric Digest 0x000000000000000000000000000000000000000000000000000000000000000d is being
written to Mailbox Address 0x000.
# UFS CC Segment Digest 0x000000000000000000000000000000000000000000000000000000000000000c is
being written to Mailbox Address 0x00000020.
# SNVM Digest 0x000000000000000000000000000000000000000000000000000000000000000b is being
written to Mailbox Address 0x00000040.
# UFS UL Segment 0x000000000000000000000000000000000000000000000000000000000000000a is being
written to Mailbox Address 0x00000060.
# User Key Digest 0 0x0000000000000000000000000000000000000000000000000000000000000009 is
being written to Mailbox Address 0x00000080.
# User Key Digest 1 0x0000000000000000000000000000000000000000000000000000000000000008 is
being written to Mailbox Address 0x000000a0.
# User Key Digest 2 0x0000000000000000000000000000000000000000000000000000000000000007 is
being written to Mailbox Address 0x000000c0.
# User Key Digest 3 0x0000000000000000000000000000000000000000000000000000000000000006 is
being written to Mailbox Address 0x000000e0.
# User Key Digest 4 0x0000000000000000000000000000000000000000000000000000000000000005 is
being written to Mailbox Address 0x00000100.
# User Key Digest 5 0x0000000000000000000000000000000000000000000000000000000000000004 is
being written to Mailbox Address 0x00000120.
# User Key Digest 6 0x0000000000000000000000000000000000000000000000000000000000000003 is
being written to Mailbox Address 0x00000140.
# UFS UPERM Segment Digest 0x0000000000000000000000000000000000000000000000000000000000000002
is being written to Mailbox Address 0x00000160.
# Factory Digest 0x0000000000000000000000000000000000000000000000000000000000000001 is being
written to Mailbox Address 0x00000180.
# 103150 Read Digests System Services Completed
```

© 2021 Microchip Technology Inc.
and its subsidiaries

1.6 Query Security Service

Query Security service fetches non-volatile states of user security locks. The following table lists the description of returned LOCKS array.

You can pass the desired QRYSEC either by passing the value using the `vsim` command or by port mapping during instantiation.

```
vsim -L PolarFire -L presynth -L CORESYSSERVICES_PF_LIB -gQRYSEC=72'h1234 -t 1ps -
gSIM_PA5M300T=0 presynth.ss_tb
```

Table 1-12. Query Security Service Request

System Service Descriptor Bit Field	Value	Description
15:7	MBOXADDR[10:2]	Mailbox address. See Table 1-13.
6:0	05H	Query security service command

The following table lists the Query Security Service mailbox format.

Table 1-13. Query Security Service Mailbox Format

Offset	Length (bytes)	Parameter	Direction	Description
0	33	LOCKS	Output	Lock array

Table 1-14. Returned LOCKS Array

Byte	Bit	Lock	Description
0	0	UL_DEBUG	Debug instructions disable
0	1	UL_SNVM_DEBUG	sNVM debug disable
0	2	UL_LIVEPROBE	Live probes disable
0	3	UL_UJTAG	User JTAG interface disable
0	4	UL_JTAG_BS	JTAG boundary scan disable
0	5	UL_TVS_MONITOR	External access to System TVS monitor disable
0	6	UL_JTAG_MONITOR	JTAG fabric monitor enable
0	7	UL_JTAG	JTAG TAP disable
1	0	UL_PLAINTEXT	Plaintext passcode unlock disable
1	1	UL_FAB_PROTECT	Fabric erase/write disable
1	2	UL_EXT_DIGEST	External digest check disable
1	3	UL_VERSION	Replay protection enable
1	4	UL_FACT_UNLOCK	Factory test disable
1	5	UL_IAP	IAP disable
1	6	UL_EXT_ZEROIZE	External zeroization disable
1	7	UL_SPI_SLAVE	SPI port disable
2	0	UL_USL	UFS UL segment protect
2	1	UL_BS_AUTHENTICATE	External bit stream authentication disable

...........continued

Byte	Bit	Lock	Description
2	2	UL_BS_PROGRAM	External bit stream program mode disable
2	3	UL_BS_VERIFY	External bit stream verify mode disable
2	4	UL_BITS_KEYMD[0]	Bitstream key mode disable
2	5	UL_BITS_KEYMD[1]	Bitstream key mode disable
2	6	UL_BITS_KEYMD[2]	Bitstream key mode disable
2	7	UL_BITS_KEYMD[3]	Bitstream key mode disable
3	0	UL_BITS_KEYMD[4]	Bitstream key mode disable
3	1	UL_BITS_KEYMD[5]	Bitstream key mode disable
3	2	UL_BITS_KEYMD[6]	Bitstream key mode disable
3	3	UL_BITS_KEYMD[7]	Bitstream key mode disable
3	4	UL_BITS_KEYMD[8]	Bitstream key mode disable
3	5	UL_BITS_KEYMD[9]	Bitstream key mode disable
3	6	UL_BITS_KEYMD[10]	Bitstream key mode disable
3	7	UL_BITS_KEYMD[11]	Bitstream key mode disable
4	0	UL_BITS_KEYMD[12]	Bitstream key mode disable
4	1	UL_BITS_KEYMD[13]	Bitstream key mode disable
4	2	UL_BITS_KEYMD[14]	Bitstream key mode disable
4	3	UL_BITS_KEYMD[15]	Bitstream key mode disable
4	4	UL_KEYMD[0]	Global key mode disable
4	5	UL_KEYMD[1]	Global key mode disable
4	6	UL_KEYMD[2]	Global key mode disable
4	7	UL_KEYMD[3]	Global key mode disable
5	0	UL_KEYMD[4]	Global key mode disable
5	1	UL_KEYMD[5]	Global key mode disable
5	2	UL_KEYMD[6]	Global key mode disable
5	3	UL_KEYMD[7]	Global key mode disable
5	4	UL_KEYMD[8]	Global key mode disable
5	5	UL_KEYMD[9]	Global key mode disable
5	6	UL_KEYMD[10]	Global key mode disable
5	7	UL_KEYMD[11]	Global key mode disable
6	0	UL_KEYMD[12]	Global key mode disable
6	1	UL_KEYMD[13]	Global key mode disable
6	2	UL_KEYMD[14]	Global key mode disable
6	3	UL_KEYMD[15]	Global key mode disable
6	4	UL_SNVM_PROTECT	sNVM bit stream write protection enable

...........continued

Byte	Bit	Lock	Description
6	5	UL_EXT_CHALLENGE	CHALLENGE instruction disable
6	6	UL_UEK_PROTECT	UEK overwrite protection
6	7	UL_HWM	High Water Mark Reset disable
7	0	UL_ENVM_PROTECT	Disable bit stream programming of eNVM
7	1	UL_USER_KEY	User Key1 write-protect
7	2	UL_USER_KEY2	User Key2 write-protect
7	3	UP_FACTORY	Permanent factory test disable
7	4	UP_DEBUG	Permanent debug disable
7	5	UP_FABRIC	Permanent fabric write-protect
7	6	UP_UPK1	Permanent disable of UPK1
7	7	UP_UPK2	Permanent disable of UPK2
8	0	UP_DPK	Permanent disable of DPK
8	1	UP_PROTECT	Write disable for UPERM segment

Simulation Log

```
SysServices: Query Security service request received at time             77650000 ps.
#    Lock Array 0x0000000000000001234 is being written to Mailbox Address 0x000.
```

1.7 Read Debug Information Service

Read Debug Information service fetches debug information on programming, user initialization, device programming cycle count, and In-application programming (IAP) actions. The device programming cycle count increases for device PROGRAM and ERASE actions.

You can pass the desired RDDBGINF either by passing the value using the `vsim` command or by port mapping during instantiation.

```
vsim -L PolarFire -L presynth -L CORESYSSERVICES_PF_LIB -gRDDBGINF=672'h1234 -t 1ps -
gSIM_PA5M300T=0 presynth.ss_tb
```

Table 1-15. Read Debug Information Service Request

System Service Descriptor Bit Field	Value	Description
15:7	MBOXADDR[10:2]	Mailbox address. See Table 1-16.
6:0	06H	Read debug service command.

The following table lists the debug information reported by the Read Debug Information service.

Table 1-16. Debug Information

Size (Bytes)	Byte Offset	Parameter	Description
32	0	Reserved	Reserved
4	32	TOOL_INFO	Reflects the tool specific data passed in during programming. IAP sets this field to 0.

© 2021 Microchip Technology Inc. and its subsidiaries

...........continued

Size (Bytes)	Byte Offset	Parameter	Description
1	36	TOOL_TYPE	Tool types used to program device: 1: JTAG 2: IAP 3: SPI_SLAVE
4	37	Reserved	Reserved
1	41	FRAME_ERRORCODE	An error has occurred during bit stream frame processing and the error is identified by the FRAME_ERRORCODE field. See Table 1-17.
6	42	Reserved	Reserved
1	48	UIC_STATUS	Device and design initialization Status. 0: Successful completion. Others: Device and design initialization failed.
11	49	Reserved	Reserved
4	60	CYCLECOUNT	Programming cycle count.
1	64	IAP ERRORCODE	IAP Error Information. Returns ERRORCODE 21-27, see Table 1-17.
7	65	Reserved	Reserved
4	72	IAP Location	SPI address that was last run in IAP
4	76	SYSCTRL_STATUS	System Controller reset status
4	80	RESET_REASON	Reason for last device reset

The following table lists the error codes and their description.

Table 1-17. ERRORCODE

ERRORCODE	Description	Additional Notes
0	No error	—
1	Bitstream authentication failed	Invalid bit stream or wrong key used.
2	Unexpected data received	Additional data is received after end of bit stream component.
3	Invalid/corrupt encryption key	The requested key mode is disabled or the key could not be read/reconstructed.
4	Invalid component header	Invalid bit stream
5	Back level not satisfied	Bitstream version is older than that of the current back level value set in the device.
6	Illegal bit stream key mode	Bitstream key mode is not initialized or bit stream key mode is disabled by user security.
7	DSN binding mismatch	Bitstream is rejected because DSN in the bit stream does not match with the DSN present in the device. A bit stream can be bound to device's unique DSN such that only a specific device can be programmed with that bit stream.
8	Illegal component sequence	Incorrect bit stream

...........continued

ERRORCODE	Description	Additional Notes
9	Insufficient device capabilities	Bitstream is rejected because the capabilities specified in the bit stream do not match the target device's capabilities.
10	Incorrect DEVICEID	Bitstream is rejected because an attempt by the DEVICEID specified in the bit stream does not match the part identification field (for example, MPF300, MPF500 and so on) of the target device.
11	Unsupported bit stream protocol version (bit stream regeneration required)	Bitstream is rejected because of an attempt made by the old version of a device to decode a bit stream created in new format or by the new version of a device to decode a bit stream created in old format.
12	Verify not permitted on this bit stream	Verify programming action is disabled in the bit stream.
13	Invalid Device Certificate	Device certificate is invalid or not present.
14	Invalid DIB	Device Integrity Bits (DIB) are invalid.
21	Device not in SPI Master Mode	Error might occur only when the bit stream is executed through IAP mode. The System Controller SPI controller is not configured in the master mode.
22	No valid images found	Error might occur when bit stream is executed through Auto Update mode. Occurs when no valid image pointers are found.
23	No valid images found	Error might occur when bit stream is executed through IAP mode via Index mode. Occurs when No valid image pointers are found.
24	Programmed design version is the same as the Auto Update image found	Error might occur when bit stream is executed through Auto Update mode.
25	Reserved	Reserved
26	Selected image was invalid and no recovery was performed due to valid design in device.	Error might occur only when bit stream is executed through Auto Update or IAP mode. Error can also occur due to BACKLEVEL protection.
27	Selected and Recovery image failed to program	Error might occur only when bit stream is executed through Auto Update or IAP mode.
127	Abort	Non-bit stream instruction is executed during bit stream loading.
128	NVMVERIFY	Fabric or security segment verification failed.
129	PROTECTED	Device security is prevented modification of nonvolatile memory.
130	NOTENA	Programming mode not enabled.
131	PNVMVERIFY	pNVM verify operation failed.
132	SYSTEM	System hardware error (PUF or DRBG).
133	BADCOMPONENT	An internal error was is detected in a bit stream component payload.
134	HVPROGERR	Failure in programming subsystem.
135	HVSTATE	Error in the programming subsystem.

© 2021 Microchip Technology Inc.
and its subsidiaries

Simulation Log

```
SysServices: Read Debug Info service request received at time            104590000 ps.
# Read Debug Info
0x000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000001234 is being written
to Mailbox Address 0x000.
# 109470 Read Debug Info System Services Completed
```

2. Design Programming Services

An IAP image contains the image descriptor, bit stream, and optional design initialization data. The design programming services are used to authenticate entire IAP image, bit stream portion, or program the device. For more information about the return status of Design Programming Services, see 5. System Service Return Status Codes.

2.1 Execute UIC Script

Execute UIC service allows you to invoke a UIC script stored in any of the available nonvolatile memory sources.

A SPI Flash memory address can be specified instead of the image index within the SPI directory, as specified in the following table.

Table 2-1. Execute UIC Script Service Request

System Service Descriptor Bit Field	Value	Description
15:7	MBOXADDR[10:2]	For the mailbox format, see Table 2-2.
6:0	24H	Execute UIC script service command

The following table lists the Execute UIC script service mailbox format.

Table 2-2. Execute UIC Script Service Mailbox Format

Offset	Length (bytes)	Parameter	Direction	Description
0	4	ADDR	Input	Peripheral address
4	1	SRC	Input	Source peripheral type

Simulation Log
In a simulation model, only the execute UIC service receive request is displayed.

```
SysServices: Execute UIC Script service request received at time  15510000 ps
```

2.2 UIC Bitstream Authentication Service

UIC Bitstream Authentication serive can be used to authenticate the UIC Bitstream located in the SPI through a system service routine.

Table 2-3. UIC Bitstream Authentication Service Request

System Service Descriptor Bit Field	Value	Description
15:7	MBOXADDR[10:2]	Mailbox address. See Table 2-4.
6:0	25H	UIC bit stream authentication service command

The following table lists the UIC bit stream authentication service mailbox format.

Table 2-4. UIC Bit Stream Authentication Service Mailbox Format

Offset	Length (bytes)	Parameter	Direction	Description
0	4	ADDR	Input	SPI address

© 2021 Microchip Technology Inc.
and its subsidiaries

Simulation Log
In a simulation model, only the UIC bit stream authentication service receive request is displayed.

```
SysServices: UIC Bitstream Authentication service request received at time  13670000 ps
```

2.3 Bitstream Authentication Service

Prior to using the IAP service, it might be required to first validate the new bit stream before committing the device to reprogramming, thus avoiding the need to invoke recovery procedures if the bit stream is invalid.

The bit stream authentication service analyzes a bit stream image stored in SPI Flash and checks for all conditions, which might result in an authentication error. While the authentication is in progress, the user design continues to operate normally, but without access to SPI Flash and system services until the authentication process is complete.

The `spi_flash_address` parameter passed to this service specifies the address within SPI Flash where the bit stream is stored.

If the authentication service is called while a new bit stream is being loaded through the JTAG interface, the system service takes precedence and the JTAG interface is stalled during the authentication and will ultimately fail.

Table 2-5. Bitstream Authentication Service Request

System Service Descriptor Bit Field	Value	Description
15:7	MBOXADDR[10:2]	Mailbox address. See Table 2-6.
6:0	23H	Bitstream authentication service command.

The following table lists the Bitstream Authentication service mailbox format.

Table 2-6. Bitstream Authentication Service Mailbox Format

Offset	Length (bytes)	Parameter	Direction	Description
0	4	SPIADDR	Input	Address of the bitstream in SPI Flash. If an external SPI Flash device does not support 32-bit addresses, SPIADDR[31:24] is ignored.

Simulation Log
In a simulation model, only the Bitstream Authentication service receive request is displayed.

```
SysServices: Bitstream Authentication service request received at time  11890000 ps.
```

2.4 IAP Image Authentication Service

IAP Image Authentication service allows you to validate an IAP image stored in SPI Flash. The service authenticates the entire IAP image containing the image descriptor, the referenced bit stream, and optional initialization data. If the image is authenticated successfully, the image is ensured to be valid when used by an IAP programming service.

The SPI_IDX parameter passed to this service identifies the index in the SPI directory to be used. To support recovery, SPI_IDX = 1 must be an empty slot and the recovery image must be located in SPI_IDX = 0. As, SPI_IDX = 1 must be an empty slot, it must not be passed into the system service. The following table lists the fields contained in an IAP image authentication service request.

Table 2-7. IAP Image Authentication Service Request

System Service Descriptor Bit Field	Value	Description
15	—	Reserved.
14:7	SPI_IDX[7:0]	Identifies the image index in the SPI directory for image authentication.
6:0	22H	IAP Image Authenticate service command.

Simulation Log
In a simulation model, only the IAP image authentication service receive request is displayed.

```
SysServices: IAP Image Authentication service request received at time  17090000 ps.
```

© 2021 Microchip Technology Inc.
and its subsidiaries

3. Data Security Services

The data security services are used to authenticate the device, generate unique random number, and store the encrypted data. For more information about the return status of Data Security Services, see 5. System Service Return Status Codes.

3.1 Digital Signature Service

The digital signature service takes a user-supplied SHA-384 hash and signs it with the device's 384-bit private Factory EC key (FEK), which is the private half of the key pair whose public key (DCPK) is certified by Microchip in the device's X.509-compliant supply chain assurance certificate. The resulting P-384 ECDSA signature can either be formatted using ASN.1 DER or simply returned in a raw format compatible with the user cryptoprocessor. As, ECDSA requires the use of a nonce, the service returns a different result each time, even if the hash input is the same.

The system controller cryptoprocessor does not directly support generating a nonce with the required numerical range required for ECDSA. It is therefore possible that the generated nonce is rejected, in which case a new nonce is automatically generated until a good value is found. This makes the execution time of this service non-deterministic, however, the probability of an out-of-range nonce being initially generated is extremely low and the probability of a second bad nonce is infinitesimal.

```
SIGNATURE = ECDSA (FEK, HASH)
```

If the Raw format is selected, the `SIGNATURE` field contains two unsigned little-endian 12-word (48 byte) values compatible with the user cryptoprocessor.

If the DER format is selected, the `SIGNATURE` field is returned in a minimal length DER encoding using a maximum of 104 bytes. If the encoded signature is less than 104 bytes, the output is padded with zeroes. The extra bytes, if any, must be deleted by you.

Table 3-1. Digital Signature Service Request

System Service Descriptor Bit Field	Value	Description
15:7	MBOXADDR[10:2]	Mailbox address. See Table 3-2.
6:0	19H	Digital signature Raw format service command
	1AH	Digital Signature DER format service command

The following table lists the Digital Signature Service mailbox format.

Table 3-2. Digital Signature Service Mailbox Format

Offset	Length (bytes)	Parameter	Direction	Description
0	48	HASH	Input	SHA384 hash to be signed
48	96 (Raw)	SIGNATURE	Output	ECDSA signature (r, s)
	104 (DER)			

Simulation Log

In simulation model, data moving occurs with the proper display messages, but no processing of the data is done.

```
SysServices: Digital Signature RAW service request received at time  12430000 ps.
# Hash located at Mailbox Address 0x000 is being signed and stored at Mailbox  Address
0x00000030
# 17970  Digital Signature RAW System Services Completed
```

3.2 Secure NVM (sNVM) Services

Secure NVM (sNVM) occupies the Sector 1 of the pNVM. Each page of the pNVM in this region constitutes one sNVM module. Three pages of the pNVM sector are reserved for administrative purposes, leaving 221 pages available for sNVM modules. sNVM modules can be marked as ROM during bitstream programming. These modules cannot be modified by the secure NVM functions, but they can be read.

sNVM data can be stored in any of the following formats:

- Non-authenticated plaintext
- Authenticated plaintext
- Authenticated ciphertext

When the data is authenticated, 236 bytes of storage per page is available. When the data is not authenticated, 252 bytes can be stored. Non-authenticated plaintext provides the fastest access time, and authenticated ciphertext provides the highest level of security. When authentication is used, a user-provided 96-bit key USK is used to enhance security.

sNVM can be marked as ROM during bitstream programming. In this case, sNVM cannot be modified by the Secure NVM services, but it can be read.

In a simulation model, only the non-authenticated plain text formats are supported. Current, only 2048 bytes of data to wite/read in/from sNVM is supported.
Note: Simulation model does not provide the page access for sNVM reads and writes.

3.2.1 Secure NVM Write Service

The Secure NVM write service provides write access to pages in the sNVM region of the pNVM. Data can be stored as encrypted and authenticated ciphertext, authenticated plaintext, and non-authenticated plaintext.

For authenticated plaintext and authenticated ciphertext, a 512-bit sNVM Master Key (SMK) is the primary key used, with 256-bits allocated for authentication and 256 bits for encryption. SMK is common for all sNVM pages. In addition, a 96-bit User-Supplied Key (USK), is used to protect each sNVM page independently. USK is not stored on the device but must be presented to sNVM read system service to correctly retrieve the data.

For crypto-enabled options, the System Controller uses AES-256 in Synthetic Initialization Vector (SIV) mode, which supports authenticated encryption. In SIV mode, the IV used for the encryption function is computed from the data, preventing IV misuse, and doubles as the authentication tag. The computed 128-bit IV is stored in the same page as the user data, reducing the available space for user data by 16 bytes compared to the non-authenticated plaintext-only option.

Besides, the user-supplied plaintext data, PolarFire SoC FPGAs also submit additional metadata for authentication that effectively provides a “tweak” to the encryption and authentication functions. Some of the data included are the page address and the page write-counter. It means that the ciphertext and the authentication tag are different even if the same data is written to two different sNVM pages, or even if the same data is written to the same page again (as, the page-write counter advances).

USK is used as another element in the “tweak”. Without the same 96-bit, USK is used during the write command, the read command fails authentication (and could not possibly decrypt correctly, either). You can choose to set this key differently for each page, for groups of pages, or the same for all pages—either as a secret key for added security, or to a invalid value such as all zeroes if this feature is not needed.

sNVM modules marked as ROM cannot be overwritten by this service. The service cannot be used to create ROM modules (write-protected pages). ROM is declared when a bitstream is generated, and a page's ROM status can only be changed with a new bitstream, and not at run-time.

Table 3-3. Secure NVM Write Request

System Service Descriptor Bit Field	Value	Description
15:7	MBOXADDR[10:2]	Mailbox address. See Table 3-4 and Table 3-5.

...........continued		
System Service Descriptor Bit Field	**Value**	**Description**
6:0	10H	Non-authenticated plaintext service command
	11H	Authenticated plaintext service command
	12H	Authenticated ciphertext service command

The following table lists the Secure NVM Write Service Mailbox Format for Non-authenticated plaintext (10H).

Table 3-4. Secure NVM Write Service Mailbox Format (10H)

Offset	Length (bytes)	Parameter	Direction	Description
0	1	SNVMADDR	Input	sNVM address
1	3	RESERVED		Reserved
4	252	PT	Input	Data to write to sNVM

The following table lists the Secure NVM Write Service Mailbox Format for authenticated plaintext (11H) and Authenticated ciphertext (12H).

Table 3-5. Secure NVM Write Service Mailbox Format (11H, 12H)

Offset	Length (bytes)	Parameter	Direction	Description
0	1	SNVMADDR	Input	sNVM address
1	3	RESERVED		Reserved
4	236	PT	Input	Data to write to sNVM
240	12	USK	Input	User Secret Key

Simulation Log
In a simulation model, the data is written starting from the offset+4.

```
SysServices: Non-Authenticated PlainText SNVM Write service request received at time
126410000 ps.
# Data located at Mailbox Address 0x00000004 is being written to the SNVM at Address 0x000
# 126550   SEC NVM WRITE Completed
```

3.2.2 Secure NVM Read Service

The Secure NVM read service provides access to the data stored by the Secure NVM Write service or data programmed via a bitstream. If the data is programmed using authentication, USK used at the time of programming must also be provided.

Table 3-6. Secure NVM Write Request

System Service Descriptor Bit Field	Value	Description
15:7	MBOXADDR[10:2]	Mailbox address. See Table 3-7.
6:0	18H	Secure NVM Read service command

The following table lists the Secure NVM Read Service Mailbox Format (18H).

© 2021 Microchip Technology Inc.
and its subsidiaries

Table 3-7. Secure NVM Read Service Mailbox Format (18H)

Offset	Length (bytes)	Parameter	Direction	Description
0	1	SNVMADDR	Input	sNVM address
1	3	RESERVED		Reserved
4	12	USK	Input	User Secret Key (ignored if page is plaintext)
16	4	ADMIN	Output	Page admin data
20	236 or 252	PT	Output	Data read from sNVM. 236 bytes of data per page is available when the data is authenticated. 252 bytes of data per page is available when the data is not authenticated.

Simulation Log
In a simulation model, the data is read starting from the offset+14.

```
SysServices: SNVM Read service request received at time  128310000 ps.
# Data located at SNVM Address 0x000 is being written to the Mailbox at Address 0x00000014
```

3.3 PUF Emulation Service

The PUF emulation service provides a mechanism for authenticating a device or for generating pseudo-random bit strings that can be used for many different purposes. The service accepts a 128-bit challenge and an 8-bit optype, and returns a 256-bit response unique to the given challenge, optype, and device.

```
RESPONSE = KeyTree(PEK, OPTYPE, CHALLENGE)
```

Where:

- PEK is the factory-defined PUF emulation key.
- KeyTree is a function that uses the 8-bit OPTYPE concatenated with the 128-bit CHALLENGE to navigate a binary key tree with the 256-bit secret PEK at its root.
- The leaf of the tree that is computed as a result of the 136 internal hashing operations (one for each level in the binary tree), is a 256 bit secret.
- The root key, PEK, the result, RESPONSE, and the intermediate results are protected against side-channel attacks due to the nature of the protocol. The SHA algorithm implemented in the System Controller's cryptoprocessor also has strong DPA countermeasures.
- The OPTYPE and CHALLENGE are not protected against side-channel leakage. The OPTYPE allows you to conceptualize that there are 256 different 128-bit key trees, each with 2^{128} possible output responses, which can be put to different uses without much danger of collision.

The function emulates a strong PUF, which means that it takes a cryptographically large challenge space and computes a pseudo-random repeatable output response from it, but in this implementation, it does not use unclonable physical properties developed during the manufacturing of the device for the challenge-response calculation, instead using classical cryptographic algorithms; thus the “emulation” disclaimer. The root key PEK is, however, protected as an encrypted/authenticated PUF key code, so the unclonable physical properties of the PolarFire SoC device do enter into the reconstruction of the PUF secret and decryption of the key code to unwrap PEK for use in this function.

There are many uses in cryptography for such a per-device unique, pseudo-random function. One use is to identify a particular chip by first recording (possibly several) challenge-response pairs, then later seeing if the target chip provides the same response as expected for one of the recorded challenges-response pairs. Another application derives many keys from one.

© 2021 Microchip Technology Inc.
and its subsidiaries

Table 3-8. PUF Emulation Service Request

System Service Descriptor Bit Field	Value	Description
15:7	MBOXADDR[10:2]	Mailbox address. See Table 3-9.
6:0	20H	PUF Emulation service command

The following table lists the PUF Emulation Service Mailbox Format.

Table 3-9. PUF Emulation Service Mailbox Format

Offset	Length (bytes)	Parameter	Direction	Description
0	1	OPTYPE	Input	Operational type
1	3	RESERVED	—	Reserved
4	16	CHALLENGE	Input	Challenge input
20	32	RESPONSE	Output	Response output

Simulation Log
In a simulation model, the first four bytes are not supported. Data is written at offset+4 address.

```
SysServices: PUF Emulation service request received at time 11950000 ps.
# Challenge Input is stored at Mailbox address 0x00000004 and the Response will be written to
the Mailbox at address 0x00000014
```

3.4 Nonce Service

The nonce service generates a 256-bit random number derived from the start-up states of a dedicated SRAM. The nonce service provides you with the ability to strengthen the NRBG of the User Cryptoprocessor random bit generator by providing an alternate entropy source to use as additional seed data in its DRBG functions.

NONCE = KeyTree256(PUK, 0, PUFSEED)

Where, PUFSEED is a 256-bit conditioned true random output of the SRAM-PUF. PUK is a 256-bit device-generated nonce set in the factory.

To generate maximum entropy and forward and backward resistance, the SRAM-PUF is automatically power-cycled before generating the seed.

Table 3-10. Nonce Service Request

System Service Descriptor Bit Field	Value	Description
15:7	MBOXADDR[10:2]	Mailbox address. See Table 3-11.
6:0	21H	Nonce service command

The following table lists the Nonce Service Mailbox Format.

Table 3-11. Nonce Service Mailbox Format

Offset	Length (bytes)	Parameter	Direction	Description
0	32	NONCE	Output	Generated nonce

© 2021 Microchip Technology Inc.
and its subsidiaries

Simulation Log

In a simulation model, a random number is stored in the mailbox offset address and might not match the silicon.

```
SysServices:  Received NONCE service request
# SysServices:  Generated Nonce :
00000000000000000000000000000000000000000000000000000000fbb52986 is read from Mailbox Offset
Address :0
#
#13470  Nonce Services Completed
```

4. Fabric Services

Fabric services are used to calculate digests of nonvolatile memories and program the device. For more information about the return status of fabric services, see 5. System Service Return Status Codes.

4.1 Digest Check Service

Digest Check service recalculates digests of selected nonvolatile memories and compares against stored values. The OPTIONS parameter passed in the digest check service indicates the area for which the digest check must be performed.

Table 4-1. Digest Check Service Request

System Service Descriptor Bit Field	Value	Description
15:7	MBOXADDR[10:2]	Mailbox address. See Table 4-2.
6:0	47H	Digest Check service command

The following table lists the Digest Check Service mailbox format.

Table 4-2. Digest Check Service Mailbox Format

Offset	Length (bytes)	Parameter	Direction	Description
0	4	OPTIONS	Input	Digest options. See Table 4-3.
4	4	DIGESTERR	Output	See Table 4-4.

Table 4-3. OPTIONS

OPTIONS	Name	Description
0	CHECK FABRIC	Enables fabric digest.
1	CC	Enables digest of fabric configuration data.
2	sNVM	Enables digest of sNVM pages marked as ROM.
3	UL	Enables digest of user security segment.
4	UKDIGEST0	Enables digest of user key segment containing SRAM-PUF data.
5	UKDIGEST1	Enables digest of user key segment containing UEK (User EC key).
6	UKDIGEST2	Enables digest of user key segment containing UPK1.
7	UKDIGEST3	Enables digest of user key segment containing UEK1.
8	UKDIGEST4	Enables digest of user key segment containing DPK.
9	UKDIGEST5	Enables digest of user key segment containing UPK2.
10	UKDIGEST6	Enables digest of user key segment containing UEK2.
11	UPERM	Enables digest of permanent lock security segments.
12	SYS	Enables digest of factory lock segment, factory key segment in pNVM, and System Controller ROM.

If CHECK FABRIC is 1, the FPGA fabric is placed in suspend state and I/Os behave in the same way as programming mode. Upon completion of the fabric digest, the suspend state is automatically exited. LSRAMs do not retain the user data after performing digest check on FPGA fabric. The status of the fabric digest check must be monitored by MSS. After checking the status of the fabric digest check, MSS needs to issue a design reset or device reset depending on the design requirements. Use RESET_DEVICE tamper response signal for device reset.

If CHECK FABRIC is 0, the fabric continues to operate as normal during the requested digest calculations.

If a digest mismatch occurs, DIGESTERR indicates the selected digests are in error as listed in Table 4-4. A failure of any digest results in the DIGEST tamper flag being triggered. The DIGESTERR indicates zero when it is successful.

Table 4-4. DIGESTERR

DIGESTERR Bit Field	Name	Description
0	FABRICERR	Fabric digest error (0 if CHECK FABRIC is 0)
1	CCERR	Fabric configuration digest error
2	SNVMERR	sNVM (ROM pages) digest error (0 if CHECKSNVM is 0)
3	ULERR	User security segment digest error
4	UK0ERR	Digest error in user security segment containing SRAM-PUF data
5	UK0ERR	Digest error in user security segment containing UEK (User EC key)
6	UK2ERR	Digest error in user security segment containing UPK1
7	UK3ERR	Digest error in user security segment containing UEK1
8	UK4ERR	Digest error in user security segment containing DPK
9	UK5ERR	Digest error in user security segment containing UPK2
10	UK6ERR	Digest error in user security segment containing UEK2
11	UPERR	Digest error in permanent security lock segments
12	SYSERR	Digest error in factory key segment, factory lock segment, or System Controller ROM.

Simulation Log
In a simulation model, only the digest check service receive request is displayed.

```
SysServices:  Received Digest options at Mailbox Offset address : 0
# SysServices:  Executed Digest Check service
```

4.2 In-Application Programming (IAP)/Auto Update Service

IAP reprograms the device with a specific programming image. In IAP, regardless of the image version, the device chooses the programming image based on either the image index or the SPI image address. The MSS specifies the programming image and initiates reprogramming of the device using the IAP system service.

The user application initiates an IAP system service request using the core system services IP. The system service specifies whether the image is used for verification or programming. The System Controller automatically reads the bitstream from the SPI Flash to verify or program the device contents.

Verify Operation
The verify operation compares the specified programming image contents with the device contents. The following table lists the fields in an IAP system service request using the image index.

© 2021 Microchip Technology Inc.
and its subsidiaries

Table 4-5. IAP Verify Request by Image Index

System Service Descriptor Bit Field	Value	Description
15	—	Reserved.
14:7	SPI_IDX[7:0]	Identifies the image index in the SPI directory for IAP operation.
6:0	44H	IAP verify operation.

An SPI Flash memory address can be specified instead of the image index within the SPI directory, as shown in the following table.

Table 4-6. IAP Verify Request by Image Address

System Service Descriptor Bit Field	Value	Description
15:7	MBOXADDR[10:2]	Mailbox address. See Table 4-9.
6:0	45H	IAP verify operation.

Upon successful IAP verification, the status code 0 is generated. If the IAP verification fails, an 8-bit error code is generated.

Note: 4.1 Digest Check Service is recommended to verify the integrity of the device contents instead of IAP verify operation.

Program Operation

The program operation updates the device contents using a specified programming image. The IAP program operation does not authenticate the image before executing the program. The image can be authenticated using the IAP image authentication system service.

The user application cannot obtain the status code in the following scenarios:

- If IAP is successful, the device is automatically restarted to initialize a new design.
- If IAP fails, the IAP recovery procedure attempts to program the device with image 0.

Note: IAP recovery considers image 0 when the pointer to image 1 in the SPI directory is null.

The following table lists the fields in an IAP system service request using the image index.

Table 4-7. IAP Program Request by Image Index

System Service Descriptor Bit Field	Value	Description
15	—	Reserved.
14:7	SPI_IDX[7:0]	Identifies the image index in the SPI directory for IAP operation.
6:0	42H	IAP program operation.

An SPI Flash memory address can be specified instead of the image index within the SPI directory, as specified in the following table.

Table 4-8. IAP Request by Image Address

System Service Descriptor Bit Field	Value	Description
15:7	MBOXADDR[10:2]	For the mailbox format, see Table 4-9.

...........continued		
System Service Descriptor Bit Field	Value	Description
6:0	43H	IAP program operation.

The following table lists the IAP mailbox format.

Table 4-9. IAP Mailbox Format

Offset	Length (bytes)	Parameter	Direction	Description
0	4	SPIADDR	Input	Programming image address in SPI Flash memory. If the attached SPI Flash device does not support 32-bit addresses, SPIADDR[31:24] is ignored.

Auto Update

In this service, the newest image of the first two images in the SPI directory is chosen to be programmed.

Table 4-10. Digest Check Service Request

System Service Descriptor Bit Field	Value	Description
15:7	Reserved	Reserved
6:0	46H	Auto Update service command

The user application cannot obtain the status code in the following scenarios:

- If the auto update program is successful, the device is automatically restarted to initialize a new version of the design.
- If the auto update program fails, the auto update recovery procedure attempts to program the device with the valid image again.
- If the device remains blank at the end of auto update, there is no indication through I/O and user intervention is required.

Simulation Log

```
SysServices:  Received IAP Program by Index service request
# SysServices:  Executed IAP Program by Index service
SysServices  :  Received IAP Program by SPIADDR service request
# SysServices:  Executed IAP Program by SPIADDR service

SysServices:  Received IAP Program by Auto Update service request
# SysServices:  Executed IAP Program by Auto Update service

SysServices:  Received IAP Program by SPIADDR service request
# SysServices:  Executed IAP Program by SPIADDR service
```

5. System Service Return Status Codes

The following table lists all the system services with their command values and return status.

Table 5-1. PolarFire FPGA System Services Status Code

Category	System Service Name	Command Value in Hexadecimal	Response Status
Device and Design Information Services	Serial Number Service	0x0	0: Success
	USERCODE Service	0x1	0: Success
	Design Information Service	0x2	0: Success
	Design Certificate Service	0x3	0: Success - Certificate is valid and consistent with device. 1: Device mismatch - Public key or FSN does not match with a device. 2: Signature invalid - Certificate signature is invalid. 3: System error - PUF or storage failure.
	Read Digests Service	0x4	0: Success
	Query Security Service	0x5	0: Success
	Read Debug Information Service	0x6	0: Success
Design Programming Services	Execute UIC Script	0x24	—
	UIC Bitstream Authentication Service	0x25	—
	Bitstream Authentication Service	0x23	—
	IAP Image Authentication Service	0x22	—
Data Security Services	Digital Signature Service	0x19 RAW format 0x1A DER format	0: Success. 1: FEK failure - Error retrieving FEK. 2: DRBG error - Failed to generate nonce. 3: ECDSA error - ECDSA failed.

© 2021 Microchip Technology Inc.
and its subsidiaries

...........continued

Category	System Service Name	Command Value in Hexadecimal	Response Status
Secure NVM (SNVM) Functions	Secure NVM Write Service	0x10 Non-authenticated plain text format 0x11H Authenticated plain text format 0x12 Authenticated ciper text format	0: Success. 1: Invalid SNVMADDR - Illegal page address. 2: Write failure - PNVM program/ verify failed. 3: System error - PUF or storage failure. 4: Write Not Permitted - ROMFLAG is set.
	Secure NVM Read Service	0x18	0: Success. 1: Invalid SNVMADDR - Illegal page address. 2: Authentication failure - Page blank, storage corrupt or incorrect USK. 3: System error - PUF or storage failure.
	PUF Emulation Service	0x20	—
	Nonce Service	0x21	—
Fabric Services	Flash Freeze Service	0x40 FlashFreeze 0x41 Flash Freeze with time-out	—
	Digest Check Service	0x47	—
	In-application Programming Service	0x42 IAP program by index 0x44 IAP verify by index 0x43 IAP program by SPIADDR 0x45 IAP verify by SPIADDR	—
	Auto Update Service	0x46	—

6. Error Response on System Services

In a simulation model, you can provide the error/success status within a text file (`.txt`) using the `vsim` command. For example, use the following example command to pass the `ERROR_RESPONSE.txt` file to simulation model.

```
vsim -L PolarFire -L presynth -L CORESYSSERVICES_PF_LIB  -
gSYS_SERVICES_RESPONSE_FILE=ERROR_RESPONSE.txt -t 1ps presynth.ss_tb
```

The following is the format of error status to be written in text file.

```
[14:8] – SERVICEID
[7:0] – Error Status
```

The following is a snippet from a sample error status text file.

```
0024      // Serial Number Service = 00H, Response = 24H
0105      // USERCODE Service = 01H Response = 24H
0202      // Design Info Service = 02H Response = 24H
0303      // Device Certificate Service = 03H Response = 03H (System error : PUF or storage
failure )
0424      // Read Digests Service = 04H Response = 24H
0524      // Query Security Service  = 05H Response = 24H
0624      // Read Debug Info Service = 06H Response = 24H
0724      // eNVM Parameters Info Service = 07H Response = 24H
240C      // Execute UIC Script Service = 24H Response = 0CH (Script Timeout Error)
250C      // UIC Bitstream Authentication Service = 25H, Response = 0CH (Script Timeout Error)
2302      // Bitstream Authentication Service = 23H, Response = 02H(Unexpected data received)
2202      // IAP Image Authentication Service  = 22H, Response = 02H(Unexpected data received)
1901      // Digital Signature Service(Raw Format)  = 19H, Response = 01H(FEK Failure)
1A02      // Digital Signature Service(DER Format)  = 1AH, Response = 02H(DRBG Error)
1002      // Secure NVM Write Service(Non-authenticated plaintext) = 10H, Response =
02H(Write failure)
1103      // Secure NVM Write Service(Authenticated plaintext) = 11H, Response = 03H(System
error)
1204      // Secure NVM Write Service(Authenticated ciphertext) = 12H, Response = 04H(Write
Not Permitted)
1803      // Secure NVM Read  Service = 18H, Response = 03H(System error)
2001      // PUF Emulation Service  = 20H, Response = 01H(Internal error)
2101      // Nonce Service   = 21H, Response = 01H(Error fetching PUK)
4005      // FlashFreeze Service   = 40H, Response = 05H(Exit initiated by IO SCB interrupt)
4106      // FlashFreeze with timeout Service   = 41H, Response = 06H(Exit initiated by mesh
error)
4701      // Digest Check Service   = 47H, Response = 01H(FABRICERR)
4202      // IAP Program by Index Service   = 42H, Response = 02H(Unexpected data received)
4403      // IAP Verify by Index Service   = 44H, Response = 03H(Invalid/corrupt encryption
key)
4304      // IAP Program by SPIADDR Service   = 43H, Response = 04H(Invalid component header)
4505      // IAP Verify by SPIADDR Service   = 45H, Response = 05H(Back level not satisfied)
4606      // IAP Verify by SPIADDR Service   = 46H, Response = 06H(Illegal bitstream mode)
```

7. Revision History

Revision	Date	Description
A	08/2021	Initial Revision.

8. Microchip FPGA Technical Support

Microchip FPGA Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, and worldwide sales offices. This section provides information about contacting Microchip FPGA Products Group and using these support services.

8.1 Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

- From North America, call **800.262.1060**
- From the rest of the world, call **650.318.4460**
- Fax, from anywhere in the world, **650.318.8044**

8.2 Customer Technical Support

Microchip FPGA Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microchip FPGA Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues, and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

You can communicate your technical questions through our Web portal and receive answers back by email, fax, or phone. Also, if you have design problems, you can upload your design files to receive assistance. We constantly monitor the cases created from the web portal throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

Technical support can be reached at soc.microsemi.com/Portal/Default.aspx.

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), log in at soc.microsemi.com/portal/DPortal.aspx, go to the **My Cases** tab, and select **Yes** in the ITAR drop-down list when creating a new case. For a complete list of ITAR-regulated Microchip FPGAs, visit the ITAR web page.

You can track technical cases online by going to My Cases.

8.3 Website

You can browse a variety of technical and non-technical information on the Microchip FPGA Products Group home page, at www.microsemi.com/soc.

8.4 Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support at (https://soc.microsemi.com/Portal/Default.aspx) or contact a local sales office.

Visit About Us for sales office listings and corporate contacts.

© 2021 Microchip Technology Inc.
and its subsidiaries

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods being used in attempts to breach the code protection features of the Microchip devices. We believe that these methods require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Attempts to breach these code protection features, most likely, cannot be accomplished without violating Microchip's intellectual property rights.
- Microchip is willing to work with any customer who is concerned about the integrity of its code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable." Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication is provided for the sole purpose of designing with and using Microchip products. Information regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications.

THIS INFORMATION IS PROVIDED BY MICROCHIP “AS IS”. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, FlashTec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, Inter-Chip Connectivity, JitterBlocker, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SMART-I.S., storClad, SQI, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, TSHARC, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2021, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-8618-3

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4485-5910 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820