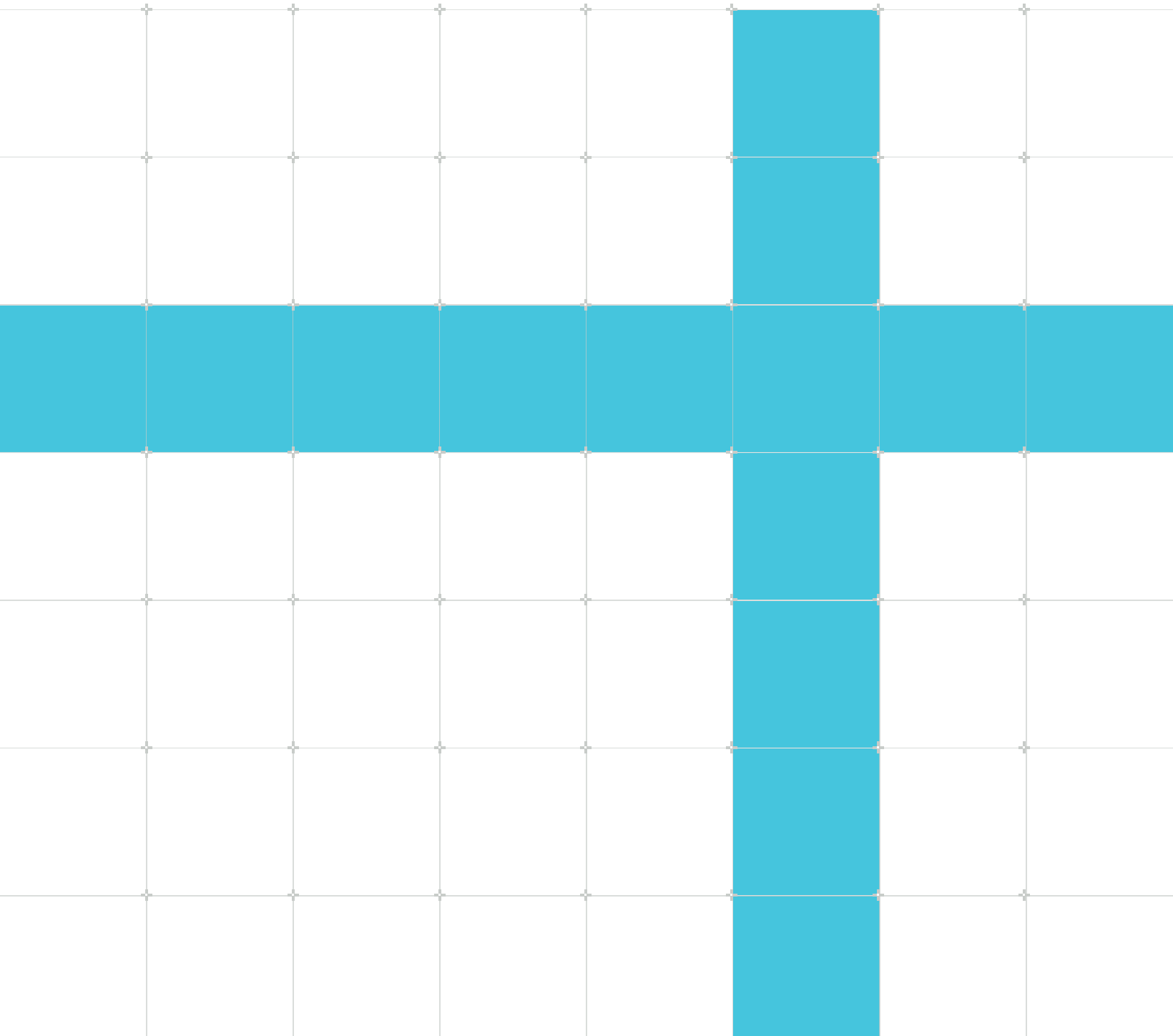
Arm® CoreLink™ GFC-200 Generic Flash Controller

Revision: r0p0

Technical Reference Manual

Issue 01
101484_0000_01_en



Arm® CoreLink™ GFC-200 Generic Flash Controller

Technical Reference Manual

Release Information

Document history

Issue	Date	Confidentiality	Change
0000-00	19 February 2019	Non-Confidential	First early access release for r0p0.
0000-01	28 February 2022	Non-Confidential	First release for r0p0 REL.

Proprietary Notice

This document consists solely of commercial items. You shall be responsible for ensuring that any use, duplication or disclosure of this document complies fully with any relevant export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word "partner" in reference to Arm's customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of the Agreement shall prevail.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349)

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Unrestricted Access is an Arm internal classification.

Product Status

The information in this document is Final, that is for a developed product.

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on the product, create a ticket on https://support.developer.arm.com.

To provide feedback on the document, fill the following survey: https://developer.arm.com/documentation-feedback-survey.

Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

Previous issues of this document included language that can be offensive. We have replaced this language. See B Revisions on page 90.

To report offensive language in this document, email terms@arm.com.

Contents

1 Introduction

1.1 Product revision status

The r*x*p*y* identifier indicates the revision status of the product described in this manual, for example, r1p2, where:

r*x* Identifies the major revision of the product, for example, r1.
p*y* Identifies the minor revision or modification status of the product, for example, p2.

1.2 Intended audience

This book is for system designers, system integrators, and programmers who are designing or programming a *System-on-Chip* (SoC) that uses the GFC-200 Generic Flash Controller.

1.3 Conventions

The following subsections describe conventions used in Arm documents.

Glossary

The Arm® Glossary is a list of terms used in Arm documentation, together with definitions for those terms. The Arm Glossary does not contain terms that are industry standard unless the Arm meaning differs from the generally accepted meaning.

See the Arm Glossary for more information: developer.arm.com/glossary.

Typographic conventions

Convention	Use
italic	Citations.
bold	Interface elements, such as menu names. Signal names. Terms in descriptive lists, where appropriate.
`monospace`	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
`monospace bold`	Language keywords when used outside example code.
`monospace underline`	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.

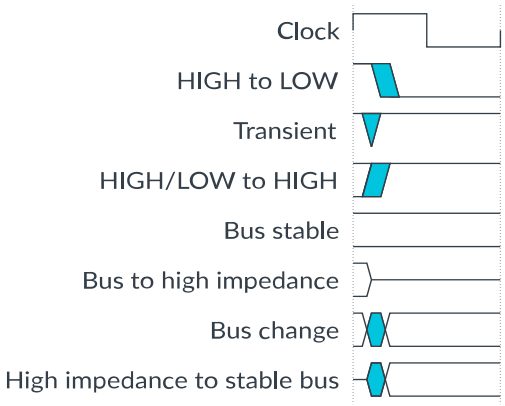
Convention	Use
<and>	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: `MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2>`
SMALL CAPITALS	Terms that have specific technical meanings as defined in the *Arm® Glossary*. For example, **IMPLEMENTATION DEFINED**, **IMPLEMENTATION SPECIFIC**, **UNKNOWN**, and **UNPREDICTABLE**.
Caution	Recommendations. Not following these recommendations might lead to system failure or damage.
Warning	Requirements for the system. Not following these requirements might result in system failure or damage.
Danger	Requirements for the system. Not following these requirements will result in system failure or damage.
Note	An important piece of information that needs your attention.
Tip	A useful tip that might make it easier, better or faster to perform a task.
Remember	A reminder of something important that relates to the information you are reading.

Timing diagrams

The following figure explains the components used in timing diagrams. Variations, when they occur, have clear labels. You must not assume any timing information that is not explicit in the diagrams.

Shaded bus and signal areas are undefined, so the bus or signal can assume any value within the shaded area at that time. The actual level is unimportant and does not affect normal operation.

Figure 1-1: Key to timing diagram conventions



Signals

The signal conventions are:

Signal level

The level of an asserted signal depends on whether the signal is active-HIGH or active-LOW. Asserted means:

- HIGH for active-HIGH signals.
- LOW for active-LOW signals.

Lowercase n

At the start or end of a signal name, n denotes an active-LOW signal.

1.4 Additional reading

This document contains information that is specific to this product. See the following documents for other relevant information:

Table 1-2: Arm publications

Document name	Document ID	Licensee only
AMBA® APB Protocol Specification Version 2.0	IHI 0024	No
AMBA® 3 AHB-Lite Protocol Specification v1.0	IHI 0033A	No
AMBA® Low Power Interface Specification, Arm® Q-Channel and P-Channel Interfaces	IHI 0068	No
Arm® CoreLink™ GFC-200 Generic Flash Controller Configuration and Integration Manual	101485	Yes
AMBA® Generic Flash Bus Protocol Specification	IHI 0083	Yes

Table 1-3: Other publications

Document ID	Organization	Document name
JEP106	JEDEC	*Standard Manufacturer's Identification Code*

Arm tests its PDFs only in Adobe Acrobat and Acrobat Reader. Arm cannot guarantee the quality of its documents when used with any other PDF reader.

Adobe PDF reader products can be downloaded at http://www.adobe.com

2 Overview

Read this for an introduction to the GFC-200 Generic Flash Controller and its features.

2.1 About the GFC-200

The GFC-200 Generic Flash Controller comprises the generic part of a Flash controller in a *System-on-Chip* (SoC). The GFC-200 enables an embedded Flash macro to be integrated easily into any system.

An eFlash macro enables a Flash controller to access eFlash memory. The eFlash macros produced by different foundries and processes can have different interfaces, timings, signal names, protocols, and features that are determined by the foundry processes that produced the eFlash memory.

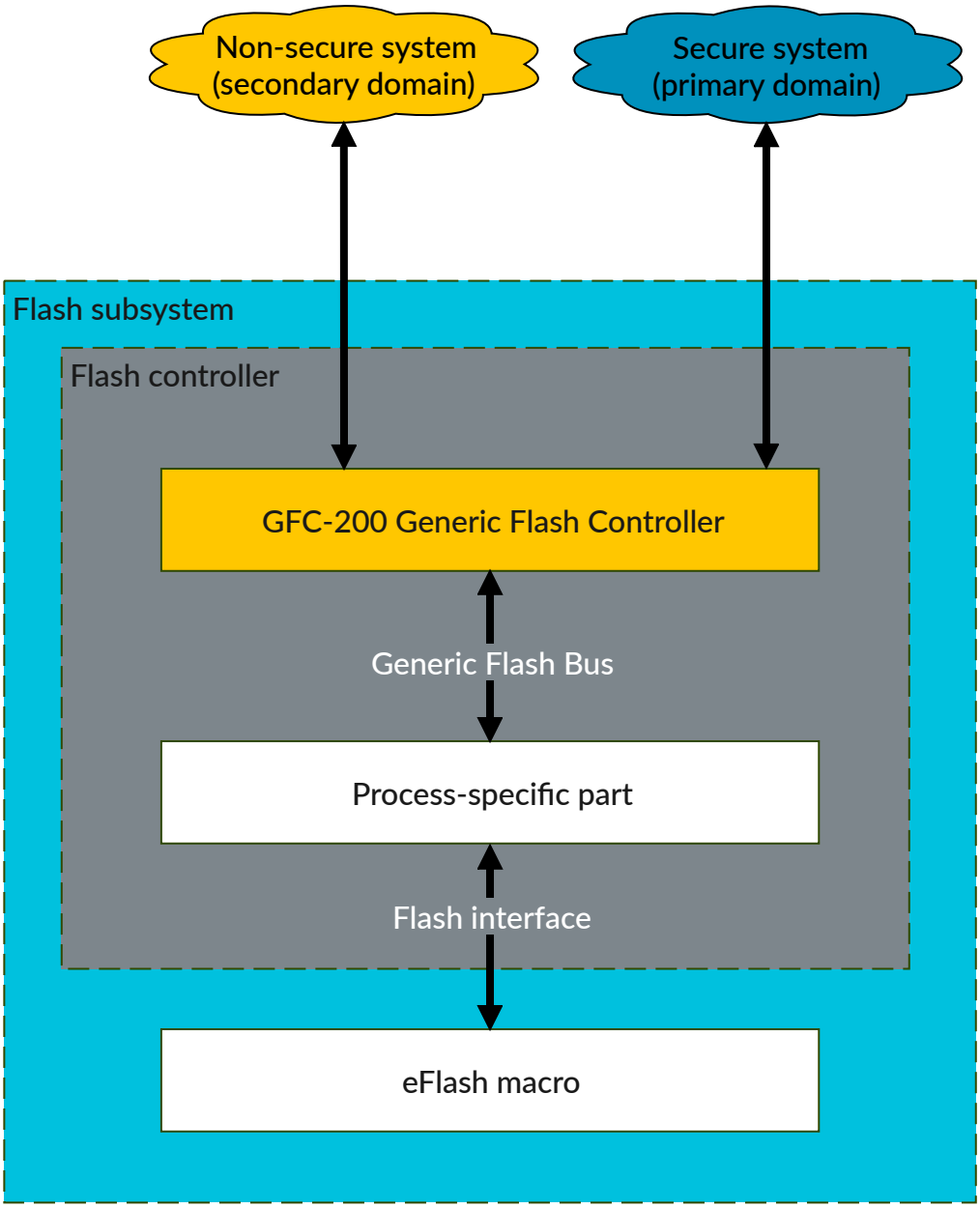
The GFC-200 provides functions that relate only to services for the system side of the Flash controller. The GFC-200 cannot communicate directly with the eFlash macro. Therefore, the GFC-200 must be integrated with a process-specific part that connects to, and communicates with, the eFlash macro.

The process-specific part of the Flash controller is part of the Flash subsystem in your SoC. It communicates directly with the eFlash macro through a Flash interface.

The GFC-200 supports accesses from two managers that can operate in separate domains such as a Non-secure domain and a Secure domain. Communication between the system and eFlash memory is through a *Generic Flash Bus* (GFB) supplied with GFC-200.

The following figure shows how the GFC-200 is used in a Flash controller implementation.

Figure 2-1: GFC-200 in a Flash controller implementation



2.2 Compliance

The GFC-200 interfaces are compliant with Arm specifications and protocols.

The GFC-200 is compliant with the:

- AMBA® GFB protocol. See the *AMBA® Generic Flash Bus Protocol Specification*.

- AMBA® 3 AHB-Lite protocol. See the *AMBA® 3 AHB-Lite Protocol Specification v1.0*.
- AMBA® 4 APB protocol. See the *AMBA® APB Protocol Specification Version 2.0*.
- AMBA® Low Power Interface specification. See the *AMBA® Low Power Interface Specification, Arm® Q-Channel and P-Channel Interfaces*.

2.3 Features

The GFC-200 provides several interfaces and features.

Flash memory partitioning:

- Ability to divide the available Flash memory space into several partitions and perform access control on a per partition basis
- Dynamically configurable access rights to partitions
- A configuration parameter controls the size of the partitions

AMBA® AHB-Lite interface:

- Read-only access to the embedded Flash
- Configurable data width
- Burst support
- Low latency

Primary APB completer interface:

- Write and erase access to the embedded Flash
- Debug read access to the embedded Flash
- Control port for GFC-200 and the eFlash macro
- Interrupt capability for long running commands
- Access to internal registers and the control registers in the process-specific part

Secondary APB completer interface:

- Write and erase access to the embedded Flash
- Debug read access to the embedded Flash
- Control port for GFC-200
- Interrupt capability for long running commands
- Access to internal registers

APB register requester interface:

- Enables access to the registers in the process-specific part

Q-Channel interface:

- Control port for system power
- Control port for the system clock

P-Channel controller interface:

- Control port for power to the process-specific part

Generic Flash Bus (GFB):

- Enables GFC-200 accesses to embedded Flash
- Simple command-based protocol
- Synchronous with the AHB clock
- Simplifies communication between GFC-200 and the attached process-specific part

2.4 Configurable options

The GFC-200 provides design-time configuration options.

At design-time, you can configure:

- The width of the AHB read data bus, by using the `HRDATA_WIDTH` parameter.
- The width of the GFB read data bus, by using the `FRDATA_WIDTH` parameter.
- The width of the GFB write data bus, by using the `FWDATA_WIDTH` parameter.
- The size of the partitions in Flash memory, by using the `PARTITION_SIZE` parameter.

2.5 Test features

GFC-200 provides components that comprise a Flash model, that simulates the behavior of the process-specific part and the attached embedded Flash.

The GFC-200 deliverables include an *Out-of-Box* (OoB) execution testbench. You can use the execution testbench to check that the delivered RTL is complete and that it can perform transactions towards the Flash model as expected.

2.6 Product documentation

Documentation that is provided with this product includes a *Technical Reference Manual* (TRM) and a *Configuration and Integration Manual* (CIM), together with architecture and protocol information.

For relevant protocol and architectural information that relates to this product, see 1.4 Additional reading on page 10".

The GFC-200 documentation is as follows:

Technical Reference Manual

The TRM describes the functionality and the effects of functional options on the behavior of GFC-200. It is required at all stages of the design flow. The choices that are made in the design flow can mean that some behaviors that the TRM describes are not relevant. If you are programming GFC-200, contact:

- The implementer to determine:
  - The build configuration of the implementation.
  - What integration, if any, was performed before implementing GFC-200.
- The integrator to determine the signal configuration of the device that you use.

The TRM complements architecture and protocol specifications and relevant external standards. It does not duplicate information from these sources.

Configuration and Integration Manual

The CIM describes:

- The available build configuration options.
- How to configure the RTL with the build configuration options.
- How to integrate GFC-200 into an SoC.
- How to implement GFC-200 into your design.
- The processes to validate the configured design.

The Arm product deliverables include reference scripts and information about using them to implement your design.

The CIM is a confidential book that is only available to licensees.

2.7 Product revisions

This section describes the differences in functionality between product revisions:

r0p0 First release.

3 Functional Description

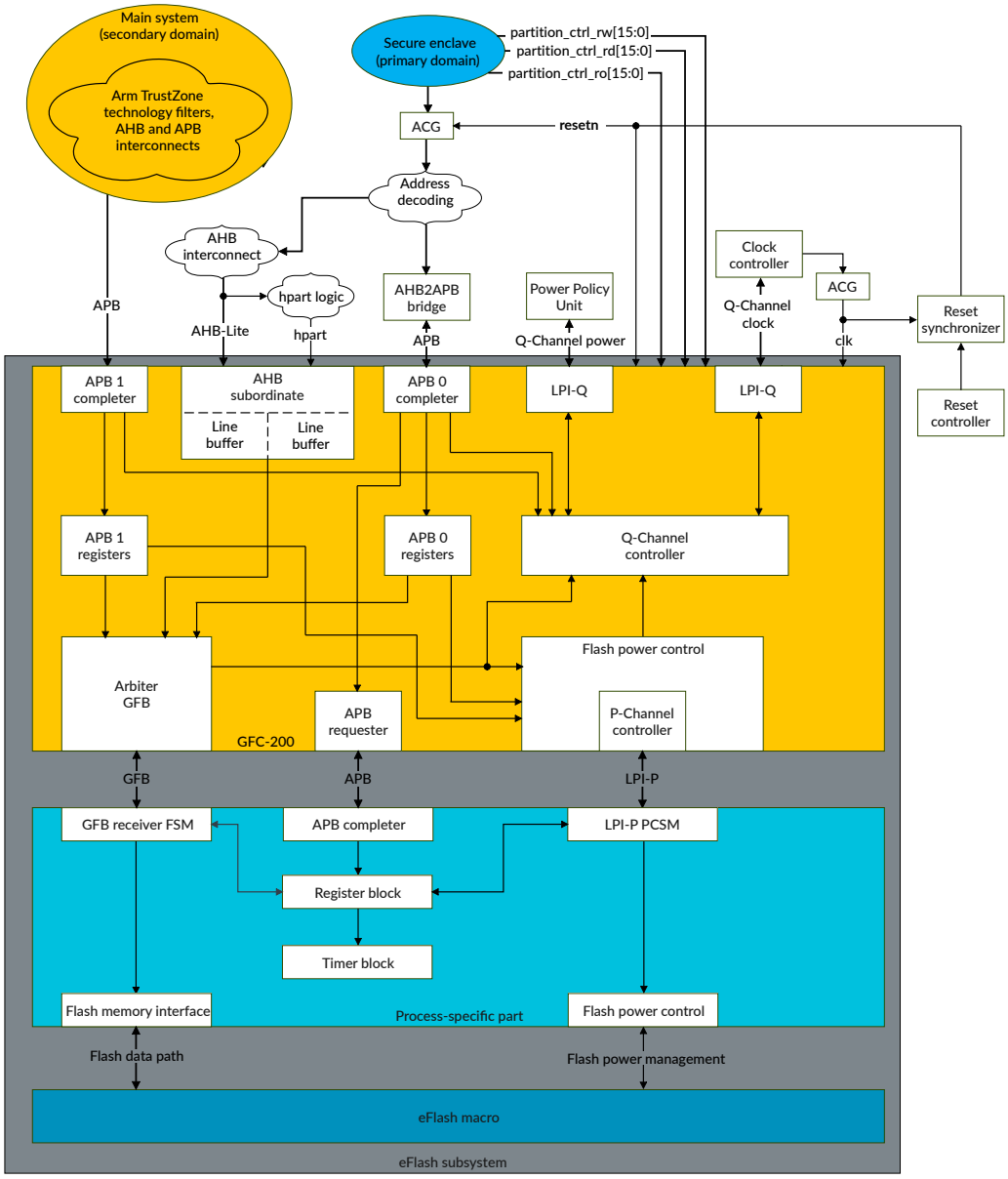
Read this for a description of the GFC-200 Generic Flash Controller functions.

3.1 Internal structure

GFC-200 comprises several submodules.

The following figure shows the internal high-level structure of an example system that integrates GFC-200 with a process-specific part and embedded Flash.

Figure 3-1: GFC-200 internal structure



3.2 Interfaces

GFC-200 has several interfaces that enable it to communicate with the system and the process-specific part.

3.2.1 AHB-Lite subordinate interface

The AHB-Lite subordinate interface is a read-only port through which the system can make read-only accesses to the embedded Flash. System accesses through the AHB-Lite subordinate interface are then transferred over the *Generic Flash Bus* (GFB).

Data width

The *HRDATA_WIDTH* configuration parameter sets the width of the data bus. Read accesses that are greater than *HRDATA_WIDTH* are ignored and generate an error response. Write accesses of any size are ignored and generate an error response.


Note

If the GFB read data bus width is smaller than the AHB data bus width, then **hsize** must not exceed the GFB read data bus width otherwise the GFC-200 generates an error response. The *FRDATA_WIDTH* configuration parameter sets the width of the GFB read data bus.

The address of the access does not have to be aligned to the data width, because it is forwarded to the GFB directly without any restriction. The *AMBA® Generic Flash Bus Protocol Specification* defines that, for wider data widths, the lower address bits are ignored. Although the AHB specification defines that the address must be aligned for all bursts, and single bursts, GFC-200 does not generate an error if an access is not aligned.

Line buffers

If the configured GFB data width is greater than the AHB data width, then each domain has a dedicated line buffer in the AHB interface logic.

Each line buffer can store an entire Flash read data word. If a new AHB read request targets an address for which the data is available in the line buffer, then the GFC-200 fulfills the request without initiating any GFB transactions. The line buffers reduce the number of Flash accesses to the minimum.

AHB sideband signal for access rights checking

The **hpart** input is a sideband signal on the AHB interface to identify the manager that initiates an AHB transaction. External logic must drive this signal for each AHB transfer during the address phase, depending on which domain initiates the transfer. For AHB transfers, the GFC-200 uses **hpart** to perform access permission checking and line buffer selection. Therefore, the system must drive **hpart** to the correct logic value and it must remain stable during the entire address phase, and during waited transfers.

Address width

The address width is fixed to allow for access to a 4MB memory area.

If the process-specific Flash controller allows less than 4MB or the Flash macro is smaller than 4MB, then aliasing might occur in the process-specific part. Therefore, the process-specific controller must provide protection against aliasing, so that it can give an error response for any out-of-range addresses.

See 4.2.1 AHB subordinate interface memory map on page 32 for a description of the AHB-Lite subordinate interface memory map.

Burst transfers

Burst transfers are supported to allow large blocks of data to be read from memory. GFC-200 supports all burst types that are specified in the *AMBA® 3 AHB-Lite Protocol Specification v1.0*. Burst transfers ensure that read accesses from the memory have priority above any other commands, and are executed at the same speed as commands are executed in the process-specific part.

Locked transfers

A system AHB manager can use locked transfers to control the GFC-200 arbitration scheme. Access is granted to the AHB-Lite subordinate port only, and all APB accesses are blocked until the entire locked transfer finishes. This behavior ensures that accesses through the AHB-Lite subordinate port have deterministic response times.

Delayed response

The response time through the AHB-Lite subordinate interface is delayed by asserting the **hreadyout** signal LOW. The following conditions assert **hreadyout**:

- A delayed GFB transfer response, because the embedded Flash is slow to respond.
- The GFB arbiter block selects a different requestor to access the memory.
- An AHB manager initiates a transfer while a Q-Channel interface is in the Q_STOPPED state.
- The GFB has a transfer in progress.

A Flash write or erase operation can require millions of clock cycles that can significantly block the AHB interconnect.

Error response

The following conditions can generate an error response:

- An AHB manager sends a write access to the GFC-200.
- The **hsize** data width exceeds *HRDATA_WIDTH*, when *HRDATA_WIDTH* ≤ *FRDATA_WIDTH*.
- The **hsize** data width exceeds *FRDATA_WIDTH*, when *FRDATA_WIDTH* ≤ *HRDATA_WIDTH*.
- A GFB transfer generates an error response.
- The primary AHB manager (**hpart** == 0) initiates an access to partition that belongs to the secondary AHB manager (**hpart** == 1), when the **partition_ctrl_rd** signal does not grant read access to the non-owner of that partition.
- The secondary AHB manager (**hpart** == 1) initiates an access to partition that belongs to the primary AHB manager (**hpart** == 0), when the **partition_ctrl_rd** signal does not grant read access to the non-owner of that partition.
- If the secondary AHB manager (**hpart** == 1) initiates an access, while the GFC-200 is in partition configuration mode.

Related information

AHB-Lite subordinate interface signals on page 82
Partition configuration interface on page 25

3.2.2 Primary APB completer interface

The primary APB completer interface enables read, write, and erase access to the embedded Flash, if that partition has the appropriate R/W permissions set. It also acts as a control port for GFC-200 and the Flash macro.

The primary APB interface also provides direct access to a register interface that is external to GFC-200. If the process-specific part contains programmable registers, then software can use the primary APB interface to initialize its parameters such as the Flash interface access times.


Note

The secondary APB interface does not have access to the external registers.

Address width

The address width is fixed at 13 bits, to allow for 2 × 4KB address spaces. One 4KB region is for internal registers, and the other 4KB region is for external registers that might reside in the process-specific part. The MSB, **paddr_s[12]**, selects either internal or external accesses.

See 4.2.2 APB memory maps on page 38 for a description of the APB completer interface memory map.

Strobe signals

The strobe signals are checked for writes to ensure that all bits are set to 1 to indicate 32-bit word accesses. Otherwise the write is ignored and has no effect on the registers. The strobe signals are forwarded to the downstream APB requester, because the process-specific part might support byte accesses.

Delayed response

Accesses to the internal register bank are serviced without delay. The design of the process-specific register bank determines how much delay can be expected for accesses that target its interface.

Error response

APB accesses to the internal register bank always give an OKAY response. For reads of reserved addresses, the GFC-200 sets **prdata_s0** LOW. The GFC-200 ignores writes to a reserved address.

The response behavior to external register accesses depends on the implementation of the attached process-specific register bank. Therefore, any errors that the APB requester interface receives, are forwarded through the APB completer interface to the access initiator.

PWAKEUP signal

The primary APB completer interface has a **pwakeup** signal. When the GFC-200 detects that the **pwakeup_s0** input goes HIGH, it sets the GFC-200 **qactive** outputs HIGH to generate a wake request to the system Q-Channel controllers.

Related information

APB completer interface signals on page 83

3.2.3 Secondary APB completer interface

The secondary APB completer interface enables read, write, and erase access to the embedded Flash, if that partition has the appropriate R/W permissions set. It also acts as a control port for GFC-200 and the Flash macro. The secondary APB interface does not have access to the external registers.

Address width

The address width is fixed at 12 bits, which provides a 4KB address region for the GFC-200 registers.

See 4.2.2 APB memory maps on page 38 for a description of the APB completer interface memory map.

Strobe signals

For writes, the GFC-200 requires that the strobe signal bits are all set to 1, to indicate a full 32-bit write. If any **pstrb_s1** bit is LOW, the GFC-200 ignores the write so it has no effect on the registers.

Delayed response

Accesses to the internal register bank are serviced without delay.

Error response

APB accesses to the internal register bank always give an OKAY response. For reads of reserved addresses, the GFC-200 sets **prdata_s0** LOW. The GFC-200 ignores writes to a reserved address.

PWAKEUP signal

The secondary APB completer interface has a **pwakeup** signal. When the GFC-200 detects that the **pwakeup_s1** input goes HIGH, it sets the GFC-200 **qactive** outputs HIGH to generate a wake request to the system Q-Channel controllers.

Related information

APB completer interface signals on page 83

3.2.4 APB requester interface

The APB requester interface is a control port for any registers that are connected externally that control the behavior of the process-specific part. This interface forwards the transfers that the primary APB completer interface receives, where **paddr_s0[12]** is HIGH.

Address width

The address width is 12 bits, and allows access to a 4KB address space.

Strobe signals

The write strobe signals can perform byte writes to external registers. Each strobe signal corresponds to a byte within the 32-bit write data.

Delayed response

The GFC-200 drives transfers through the APB requester interface that are based on the incoming accesses from the primary APB completer interface. Any standard APB completer devices can be attached to the APB requester interface. Therefore, how the completer delays an access is implementation-dependent.

Error response

Error responses that the GFC-200 APB requester interface receives, are forwarded to the GFC-200 primary APB completer interface. The conditions that determine when the attached APB completer responds with errors is implementation-dependent.

Related information

APB requester interface signals on page 85

3.2.5 GFB interface

The *Generic Flash Bus* (GFB) interface enables the GFC-200 to access embedded Flash.

Commands are received from the system through the AHB-Lite and APB completer interfaces, and converted internally to transfers over the GFB. A process-specific part that connects to the GFB is expected to serve as a receiver while GFC-200 is the manager.

Command bus

The GFC-200 supports all the commands that the *AMBA® Generic Flash Bus Protocol Specification* specifies.

Address width

The GFC-200 allows access to a 4MB memory region.

Data width

The configuration parameters `FRDATA_WIDTH` and `FWDATA_WIDTH` set the width of the read data bus and the write data bus, respectively.

The GFC-200 interprets **faddr** as little endian, so if *FWDATA_WIDTH* < *FRDATA_WIDTH* then the **faddr[3:2]** bits select the location of the write data within the wider GFB data bus. For example, if:

***FWDATA_WIDTH*=32, *FRDATA_WIDTH*=64**

faddr[2] selects the location of the 32-bit write data in the 64-bit data bus.

***FWDATA_WIDTH*=32, *FRDATA_WIDTH*=128**

faddr[3:2] selects the location of the 32-bit write data in the 128-bit data bus.

***FWDATA_WIDTH*=64, *FRDATA_WIDTH*=128**

faddr[3] selects the location of the 64-bit write data in the 128-bit data bus.

Delayed response

The GFC-200 is the GFB manager and drives commands over the GFB. A GFB receiver is allowed to delay its response to the commands. The process-specific part can take several cycles to respond to READ commands. However, WRITE and ERASE commands can take many more cycles to execute.

Error response

For GFB transfers that fail, the process-specific part can generate a 2-cycle error response. If the process-specific part generates a 2-cycle error response, the GFC-200 either:

- Generates an AHB ERROR response, when the transaction originates from an AHB manager.
- Sets STATUS.CMD_FAIL = 1, when the transaction originates from an APB requester.

If a transfer fails, then the effect on the Flash macro contents is non-deterministic because it depends on the implementation of the Flash macro and the process-specific part.

Aborting commands

The APB requesters can abort commands that they initiate by writing to the CTRL.ABORT register bit. When ABORT == 1, the GFC-200 sets the **fabort** signal HIGH. The method that the process-specific part uses to support the abort function depends on its implementation.


Note

- Commands from the AHB-Lite subordinate interface cannot be aborted.
- For more information about GFB transactions, see the *AMBA® Generic Flash Bus Protocol Specification*.

Related information

Control register, CTRL on page 47
GFB manager interface signals on page 86

3.2.6 Q-Channel interface for clock

The Q-Channel interface for clock is a control port for managing the system clock.

GFC-200 can accept or deny a request from the clock controller to turn off all operations that are using the clock.

Activity indication

GFC-200 uses the **qactive_clk** signal to indicate when any ongoing activity requires the clock. When GFC-200 asserts **qactive_clk**, the clock controller must keep the clock enabled.

Accepting a request

When GFC-200 has no activity, it accepts the quiescence request for the clock, and enters the Q_STOPPED state.

Denying a request

When the clock controller requests the clock to be disabled, if there is any ongoing activity in the GFC-200 then it denies the request.

Related information

Q-Channel interface signals on page 88

3.2.7 Q-Channel interface for power

The Q-Channel interface for power is a control port for managing the system power.

The GFC-200 can accept or deny a request from an external *Power Policy Unit* (PPU) to turn off all operations that are using power.

Activity indication

The GFC-200 uses the **qactive_pwr** signal to indicate when any ongoing activity requires power. When GFC-200 asserts **qactive_pwr**, the PPU must keep the power enabled.

Accepting a request

When the GFC-200 or the embedded Flash is inactive, the GFC-200 accepts the quiescence request for power and enters the Q_STOPPED state. The GFC-200 requests the process-specific part PCSM to enter the powerdown or full-retention state. If the PCSM accepts the request, then the GFC-200 accepts the Q-Channel request.

Denying a request

When the PPU requests power to be disabled, if there is any ongoing activity in the GFC-200 then it denies the request.

Related information

Q-Channel interface signals on page 88

3.2.8 P-Channel controller interface

The P-Channel controller interface is a control port for managing the power of the attached process-specific part.

The system-level Power Policy Unit can request changes to the GFC-200 power state. If the GFC-200 receives a powerdown request, then it uses the P-Channel controller interface to forward the request to the process-specific part.

You can also program the GFC-200 with a minimum Flash power state that the GFC-200 can transition to when it is idle. If the GFC-200 is idle, it uses the P-Channel controller interface to forward the request to the process-specific part. See 3.7 Flash power control on page 30 for more information about the power states.

When GFC-200 leaves reset and is not in quiescent state, it enables power to the embedded Flash automatically.

State encoding

The GFC-200 supports multiple power states and it uses the **pstate[4:0]** signal to encode the various Flash macro power states. See 3.7 Flash power control on page 30 for more information about the power states.

Responding to a request

The GFC-200 initiates all activity and does not expect the process-specific part to deny a request. The time that the process-specific part takes to accept a request depends on its implementation.

Related information

P-Channel controller interface signals on page 88

3.2.9 Partition configuration interface

This interface controls whether the primary domain or the secondary domain is the owner of a Flash partition. It also controls whether the domain has read/write or read-only access to a partition.

The GFC-200 splits the Flash address space into 16 partitions. The *PARTITION_SIZE* configuration parameter sets the size of all partitions.

An external Secure enclave assigns the access rights to each partition, for the primary and secondary domains, by setting the state of the following signals:

partition_ctrl_rw[15:0]

Each bit sets the domain owner of the corresponding partition. By default, the owner has read/write access to that partition.

For example, if:

- **partition_ctrl_rw[n]** is LOW, then the owner of the n^{th} partition is the primary domain.
- **partition_ctrl_rw[n]** is HIGH, then the owner of the n^{th} partition is the secondary domain.

partition_ctrl_ro[15:0]

Each bit controls whether the domain owner has read/write or read-only access to the corresponding partition.

For example, if:

- **partition_ctrl_ro[n]** is LOW, then the owner of the n^{th} partition has read/write access to that partition.
- **partition_ctrl_ro[n]** is HIGH, then the owner of the n^{th} partition has read-only access to that partition.

partitition_ctrl_rd[15:0]

Each bit controls whether the domain owner grants read access to the other domain.

For example, if:

- **partition_ctrl_rd[n]** is LOW, then the primary domain can read the n^{th} partition.
- **partition_ctrl_rd[n]** is HIGH, then the secondary domain can read the n^{th} partition.


Note

- If **partition_ctrl_rd[n]** is equal to **partition_ctrl_rw[n]**, then the partition is accessible only for the domain owner and is inaccessible to the other domain.
- If **partition_ctrl_rd[n]** is not equal to **partition_ctrl_rw[n]**, then the partition is accessible for the domain owner and is read-only to the other domain.

The GFC-200 samples the partition configuration interface signals:

- As it exits reset.
- While it is in partition configuration mode. See 4.4.16 Partition configuration mode request register, PART_CONFIG_MODE_REQ on page 57 for more information.

If 16 × *PARTITION_SIZE* is less than the address space of the Flash, the highest control inputs, **partition_ctrl_r<x>[15]**, set the access rights of the remaining part of the Flash.

If 16 × *PARTITION_SIZE* is greater than the address space of the Flash, the remaining upper control bits, **partition_ctrl_r<x>[15:n]** have no effect on the actual Flash memory space, but they do affect MASS ERASE commands.

MASS ERASE in configuration mode

The GFC-200 has an **config_mode_me_en** input signal. When the GFC-200 is in configuration mode, the primary domain can use **config_mode_me_en** to initiate a MASS ERASE operation, regardless of the partition assignment.

To enter configuration mode, set PART_CONFIG_MODE_REQ to 1 in the PARTITION_CONFIG_MODE_REQ register.

This feature is useful in the following use cases:

- If the primary domain is a Secure enclave, then the enclave can erase the Flash contents when it detects an attack.
- If the partition control inputs are tied to fixed values, a production tester can erase Flash memory after production testing finishes.

3.2.10 System interface

The GFC-200 system interface comprises two interrupt output signals and several Flash control signals.

Interrupt request

The GFC-200 has two interrupt output signals, one for the primary domain and one for the secondary domain. The GFC-200 generates an interrupt request that indicates when an important event in GFC-200 occurs. The interrupts are active-HIGH. The interrupts are cleared by accessing registers in GFC-200, and acknowledging the reason for the interrupt, see 4.4.4 Interrupt status clear register, IRQ_STATUS_CLR on page 44.

Flash power ready

When the P-Channel controller sets the embedded Flash power to ON, GFC-200 asserts the **flash_pwr_rdy** signal. This signal is sent to the process-specific part GFB receiver so that it can initiate any startup sequence that requires the embedded Flash to be fully functional.

All the other interfaces operate without any restrictions. The transfers from the AHB-Lite subordinate interface are forwarded to the GFB, but transfers are blocked if the process-specific part sets **fready** LOW while it completes the initialization tasks.

Flash power OPMODE

The GFC-200 supports two operational modes, OPMODE_0 and OPMODE_1. The **pstate[4]** signal, on the P-Channel controller interface, selects the OPMODE. The GFC-200 uses the **flash_pwr_opmode** signal to send the current operating mode to the process-specific part.

The process-specific GFB receiver can use **flash_pwr_opmode** so that it applies the correct sequences or timings that the Flash requires for each operating mode. An alternative operating mode might lower the supply voltage to the Flash, which can reduce the power consumption but with an impact on performance.

See 3.7 Flash power control on page 30 for more information about Flash power modes and OPMODEs.

Related information

System interface signals on page 82

3.3 Clocking

GFC-200 has a single clock domain, that is used for clocking all internal registers.

Synchronous interfaces

The GFC-200 AHB-Lite subordinate, APB, and GFB interfaces are expected to run on the same clock.

Asynchronous interfaces

The GFC-200 Q-Channel and P-Channel interfaces might be clocked asynchronously. Therefore, these interfaces have internal synchronizers.

3.4 Resets

GFC-200 has a single reset input. GFC-200 expects that this reset is synchronized externally.

The reset input signal must be synchronous with the clock. GFC-200 expects the reset to be asserted asynchronously and deasserted synchronously.

3.5 Interrupt sources

The GFC-200 has two active-HIGH, level-based interrupt-generating outputs, **irq0** and **irq1**. The **irq0** interrupt is for the primary domain and the **irq1** interrupt is for the secondary domain. There are multiple sources that can trigger an interrupt output event.

The following sources can trigger an interrupt output event:

Command accept (CMD_ACCEPT_IRQ)

The GFC-200 sets CMD_ACCEPT_IRQ to 1 when the arbiter accepts a write command to the CTRL register. After clearing the interrupt, you can write another command to the CTRL register where it enters a pending state, which enables back-to-back transfers to be executed. This behavior enables the use of the ROW WRITE command on the GFB. See 4.6 Preloading transfers on page 76 for more information.

If back-to-back transfers occur and the command accept and command success (or command fail) interrupts are enabled, then both interrupt sources are set to 1.

The IRQ_ENABLE_SET.CMD_ACCEPT_IRQ_EN_SET bit controls whether the relevant **irq** signal goes HIGH when CMD_ACCEPT_IRQ is set to 1.

Command success (CMD_SUCCESS_IRQ)

The GFC-200 sets CMD_SUCCESS_IRQ to 1 when the Flash successfully executes a command. The software driver can use this interrupt to know when the command completes.

When CMD_SUCCESS_IRQ == 1, the GFC-200 ignores writes to the CTRL register. Therefore, software must clear the set CMD_SUCCESS_IRQ bit by setting IRQ_STATUS_CLR.CMD_SUCCESS_IRQ_STS_CLR =1, before it can write another command to the CTRL register.

The IRQ_ENABLE_SET.CMD_SUCCESS_IRQ_EN_SET bit controls whether the relevant **irq** signal goes HIGH when CMD_SUCCESS_IRQ is set to 1.

Command fail (CMD_FAIL_IRQ)

The GFC-200 sets CMD_FAIL_IRQ to 1 when the Flash fails to successfully execute a command. The software driver can use this interrupt to know when the command completes.

When CMD_FAIL_IRQ == 1, the GFC-200 ignores writes to the CTRL register. Therefore, software must clear the set CMD_FAIL_IRQ bit by setting IRQ_STATUS_CLR.CMD_FAIL_IRQ_STS_CLR =1, before it can write another command to the CTRL register.

The IRQ_ENABLE_SET.CMD_FAIL_IRQ_EN_SET bit controls whether the relevant **irq** signal goes HIGH when CMD_FAIL_IRQ is set to 1.

Command reject (CMD_REJECT_IRQ)

The GFC-200 sets CMD_REJECT_IRQ to 1, when software attempts to write to the CTRL, ADDR, or DATA0 registers either:

- While a command is pending in the CTRL register.
- When any GFB command-related interrupts are pending.

The IRQ_ENABLE_SET.CMD_REJECT_IRQ_EN_SET bit controls whether the relevant **irq** signal goes HIGH when CMD_REJECT_IRQ is set to 1.

Read overflow (READ_OVERFLOW_IRQ)

The GFC-200 sets READ_OVERFLOW_IRQ to 1, when it cannot update the IRQ_MASKED_STATUS register with result of a READ command, so the command enters the finished state. As the GFC-200 is unaware whether software retrieved the previous read data, then it discards the newly received data and sets READ_OVERFLOW_IRQ.

The IRQ_ENABLE_SET.READ_OVERFLOW_IRQ_EN_SET bit controls whether the relevant **irq** signal goes HIGH when READ_OVERFLOW_IRQ is set to 1.

Power state change

The GFC-200 sets PWR_STATE_CHANGE_IRQ to 1, when the power state of the Flash memory changes.

The IRQ_ENABLE_SET.PWR_STATE_CHANGE_IRQ_EN_SET bit controls whether the relevant **irq** signal goes HIGH when PWR_STATE_CHANGE_IRQ is set to 1.

Entered partitioning configuration mode

The GFC-200 sets PART_CONFIG_MODE_IRQ to 1, to notify the primary domain when it enters configuration mode.

The IRQ_ENABLE_SET.PART_CONFIG_MODE_IRQ_EN_SET bit controls whether the **irq0** signal goes HIGH when PART_CONFIG_MODE_IRQ is set to 1.

Partition access violation

The GFC-200 sets ACC_VIOLATION_IRQ to 1, to notify the primary domain when an access violation occurs. For example, an access violation occurs when a domain attempts to access a partition that does not belong to it.

The IRQ_ENABLE_SET.ACC_VIOLATION_IRQ_EN_SET bit controls whether the **irq0** signal goes HIGH when ACC_VIOLATION_IRQ is set to 1.

3.6 Generic Flash Bus arbiter

The GFB arbiter selects requests from the AHB-Lite subordinate interface or the APB completer interfaces, and it forwards them to the Generic Flash Bus. A simple grant-request mechanism selects the appropriate interface.

Arbitration scheme

The GFB arbiter uses a round-robin arbitration scheme between the AHB-Lite subordinate interface and the two APB completer interfaces. When the AHB-Lite subordinate interface sends a burst or locked transfer, the AHB-Lite subordinate interface has priority over the APB completer interfaces until the transfer finishes.

The incoming APB and AHB requests cannot be always guaranteed to follow the arbitration scheme, for example, due to the occurrence of LPI requests, or the GFB interface is busy when incoming requests arrive. In these cases, outgoing transfers might occur in a different order to the incoming requests.

Access rights checking

After arbitration, the GFC-200 performs access control on Flash requests based on the partition access rights configuration. Only those accesses that target a permitted partition are forwarded to the GFB.

3.7 Flash power control

To control the transitions of the Flash power state, the GFC-200 includes a *Power Policy Unit* (PPU). The PPU uses the P-Channel controller interface to send power transition requests to the process-specific part.

PPU states

The **PSTATE** signal, the Q-Channel requests from the system, and the POWER_STATE_REQ registers control the chosen PPU power mode.

The following table shows the mapping of the **PSTATE** values to the PPU power modes and the Flash power modes.

Table 3-1: PSTATE to PPU and Flash power modes

PSTATE[4:0]	PPU power mode	Flash power mode
0b1_1000	ON (OPMODE_1)	All powerup (low-voltage read mode)
0b0_1000	ON (OPMODE_0)	All powerup
0b1_0111	FUNC_RET (OPMODE_1)	Sleep (low-voltage read mode)
0b0_0111	FUNC_RET (OPMODE_0)	Sleep
0b1_0101	FULL_RET (OPMODE_1)	Powerdown (low-voltage read mode)
0b0_0101	FULL_RET (OPMODE_0)	Powerdown
0b1_0000	OFF (OPMODE_1)	All power off
0b0_0000	OFF (OPMODE_0)	

4 Programmers Model

Read this for a description of the memory map and registers, and for information about programming the device.

4.1 About this programmers model

The following information applies to the GFC-200 registers:

- The base address is not fixed, and can be different for any particular system implementation. The offset of each register from the base address is fixed.
- Do not attempt to access reserved or unused address locations. Attempting to access these locations can result in unpredictable behavior.
- Unless otherwise stated in the accompanying text:
  - Do not modify undefined register bits.
  - Ignore undefined register bits on reads.
  - All register bits are reset to the reset value specified in the 4.3 Register summary on page 39.
- Access type is described as follows:

RW	Read and write.
RO	Read only.
WO	Write only.

4.2 Memory maps

The GFC-200 has several memory maps that are attached to different interfaces.

The memory map locations are:

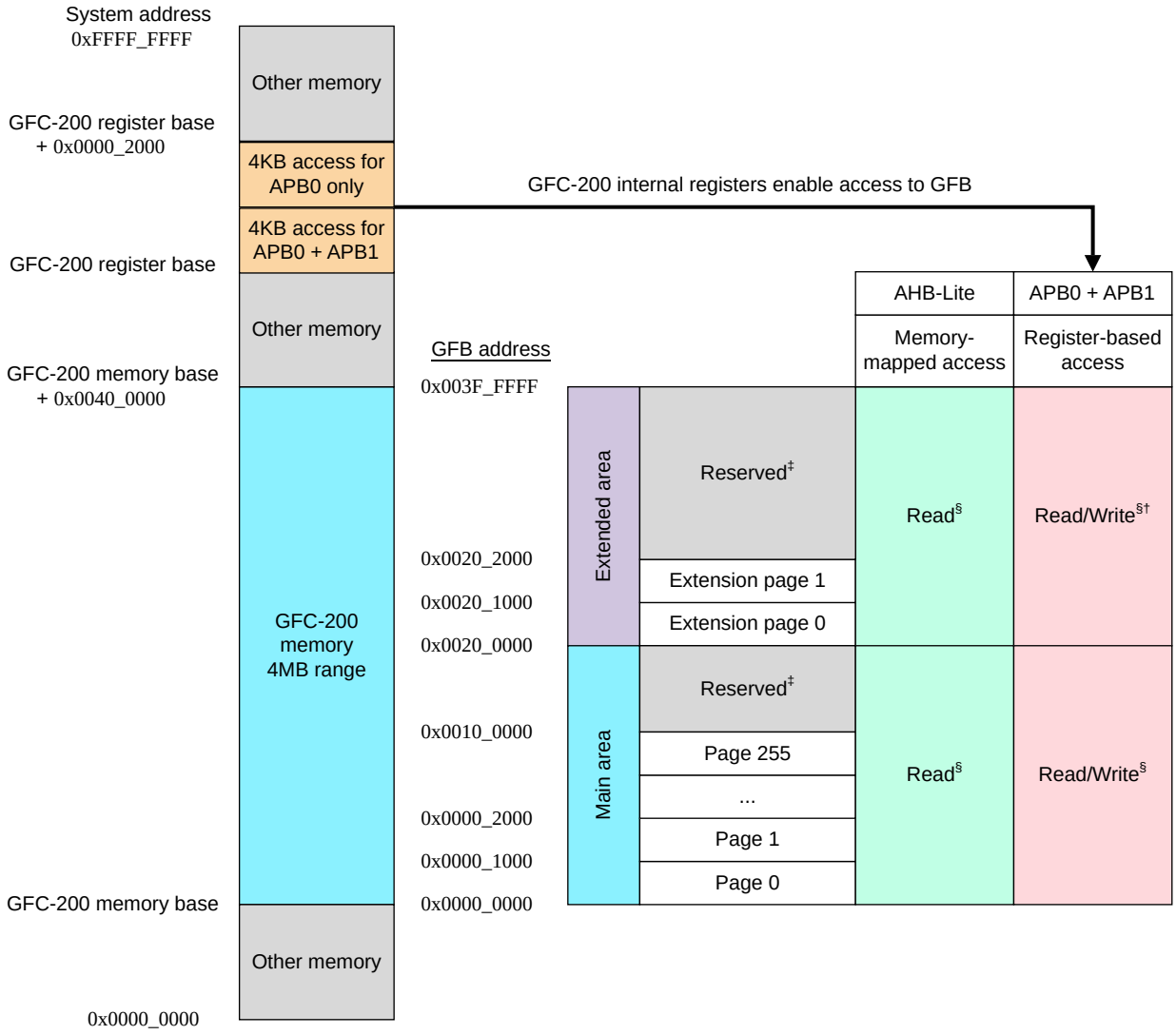
- AHB-Lite subordinate interface, 4MB.
- GFB, 4MB.
- APB primary completer interface, 8KB.
- APB secondary completer interface, 4KB.
- APB requester interface, 4KB.

4.2.1 AHB subordinate interface memory map

The AHB subordinate interface can access 4MB of memory space, which enables the contents of a 4MB, or smaller, embedded Flash to be read.

The following figure is an example that uses 1MB of large Flash with 4KB page size and two extension pages. The example shows how the embedded Flash space is accessible to the AHB interface and the APB interfaces.

Figure 4-1: AHB subordinate interface memory map



In the figure:

- † Indicates that access rights depend on the properties of extension pages that the process-specific part handles. The GFC-200 supports both read and write, or erase.
- § Access rights also depend on the partition control signals.

- ‡ For the reserved regions, the process-specific part must prevent aliasing, and must not map the same addresses multiple times within the 4MB address space of the Flash area.
  For any out-of-range addresses, Arm expects the process-specific part to respond with error.

The example memory map has 256 pages in the main memory area. GFC-200 maps 4MB of address space from the system memory to the GFB address range. The first 1MB is mapped to the embedded Flash main area directly, the second 1MB is reserved. The extended memory area has 8KB of space that is mapped to the extended memory area from the third 1MB. The last 1MB is reserved.

4.2.1.1 Accessing partitions

The partition control input signals control the accessibility of each partition in the memory map. These quasi-static inputs control whether the primary domain or secondary domain is the owner of a partition, and also the access rights that the domains are granted.

Therefore, the partition control input signals affect the accessibility of the memory map. See 3.2.9 Partition configuration interface on page 25 for more information about partitioning.

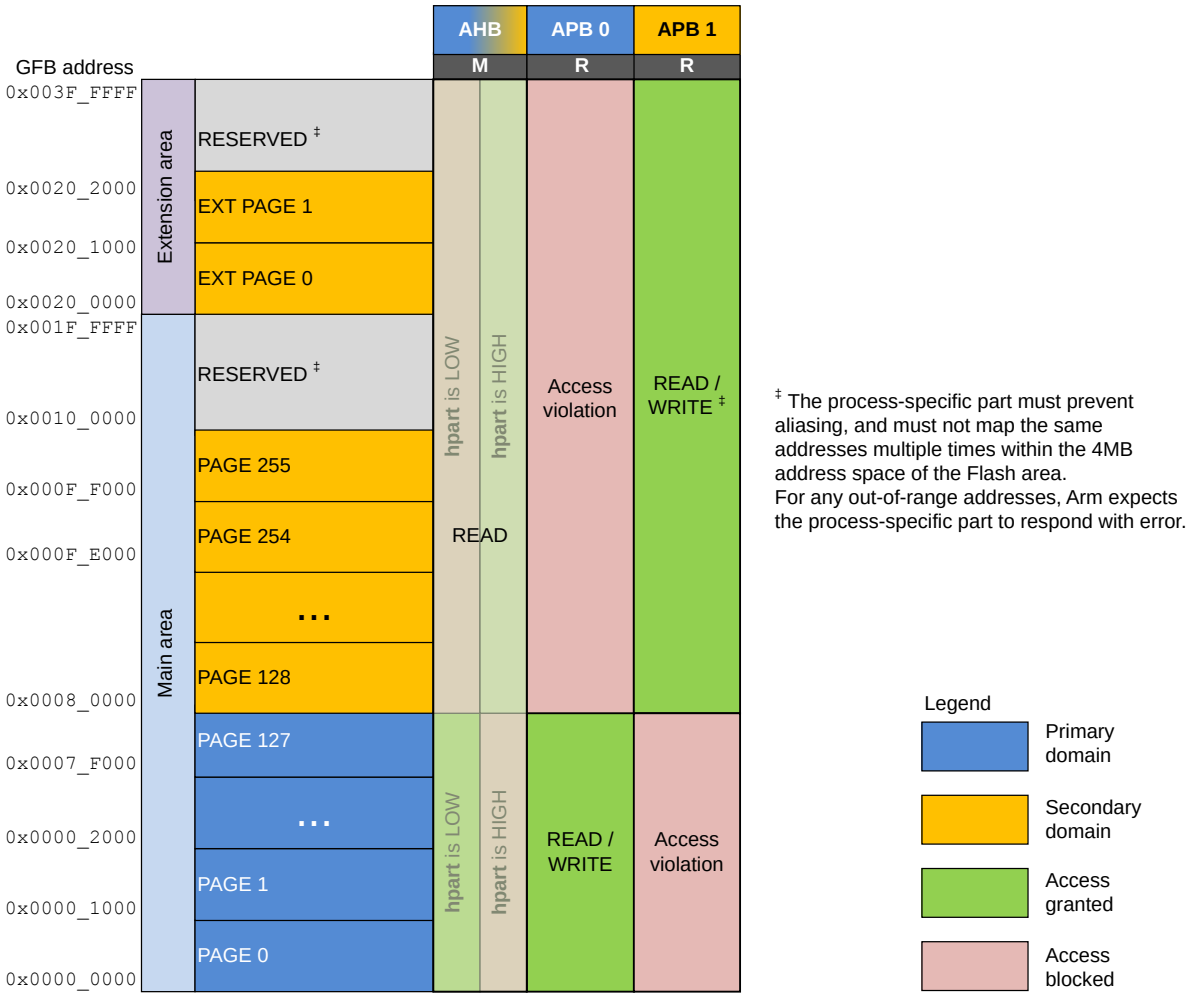
The following examples show some partition assignment configurations:

- Isolated partition assignment on page 34.
- Non-contiguous partition assignments on page 36.
- Partition owner with no modification rights on page 37.

Example 4-1: Isolated partition assignment

The following figure shows an example isolated partition assignment where the first 128 pages are assigned as primary domain partitions, and the remaining memory region belongs to the secondary domain.

Figure 4-2: Contiguous partition assignment



The primary domain can access the first 128 pages only, using either AHB (**hpart** set LOW during address phase) and APB 0.

The secondary domain can access pages 128-255, using either AHB (**hpart** set HIGH during address phase) and APB 1.

In this example, the partition size is configured to be 256KB. Partitions are isolated between the managers and none of the partitions are set to read only. For this example, the partition control input signals are:

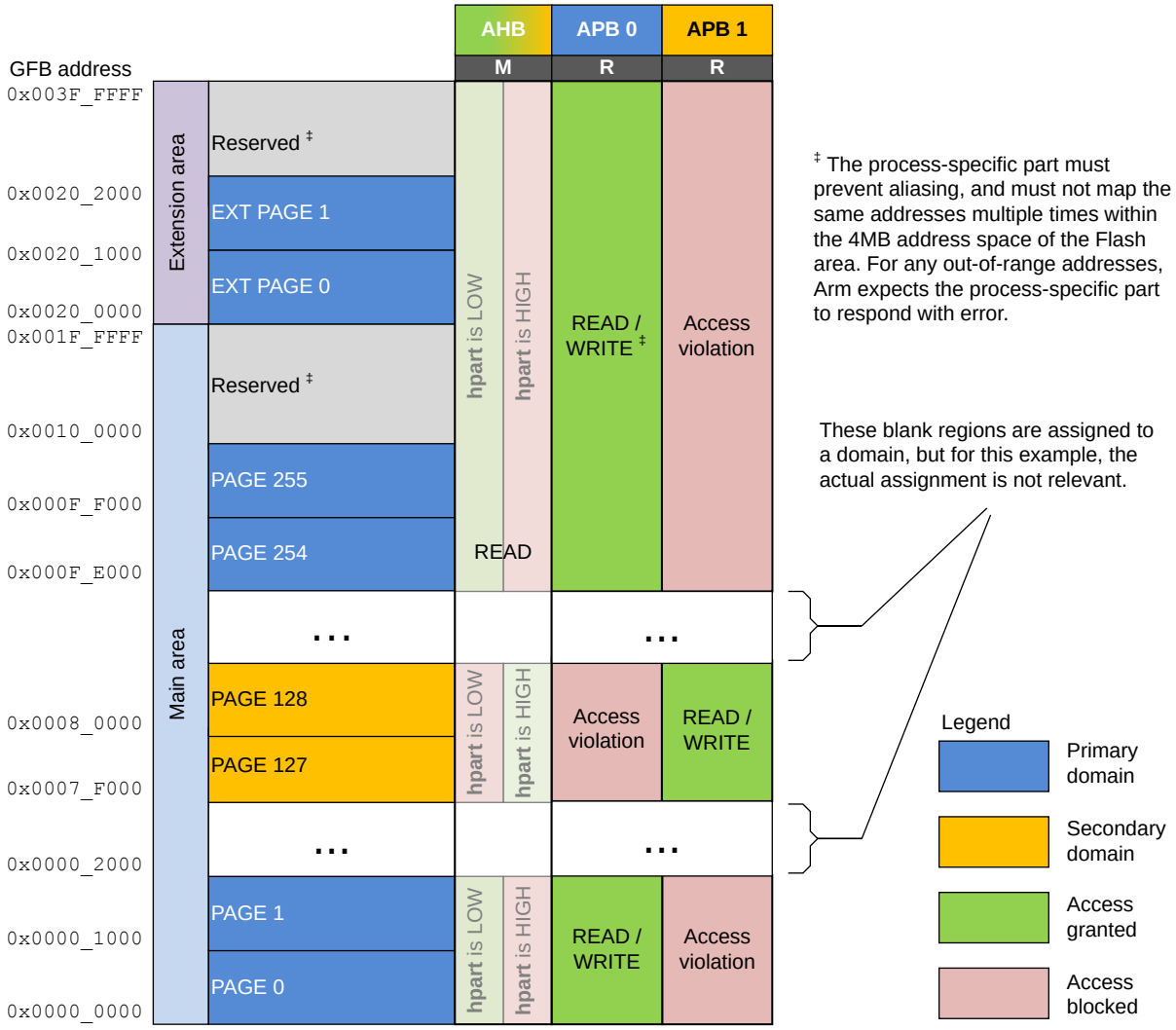
- **partition_ctrl_rw[15:0]** == `0xFFFC`
- **partition_ctrl_rd[15:0]** == **partition_ctr_rw[15:0]**

- **partition_ctrl_ro[15:0]** == 0x0000

Example 4-2: Non-contiguous partition assignments

The following figure shows an example partition assignment, where the partitions for the primary domain are non-contiguous. Other configurations are also possible because the GFC-200 allows you to assign any partition to either domain.

Figure 4-3: Non-contiguous partition assignment



In this example, the partition size is configured to be 256KB. Partitions are isolated between the managers and none of the partitions are set to read only. For this example, the partition control input signals are:

- **partition_ctrl_rw[15:0]** == 0x0006
- **partition_ctrl_rd[15:0]** == **partition_ctr_rw[15:0]**

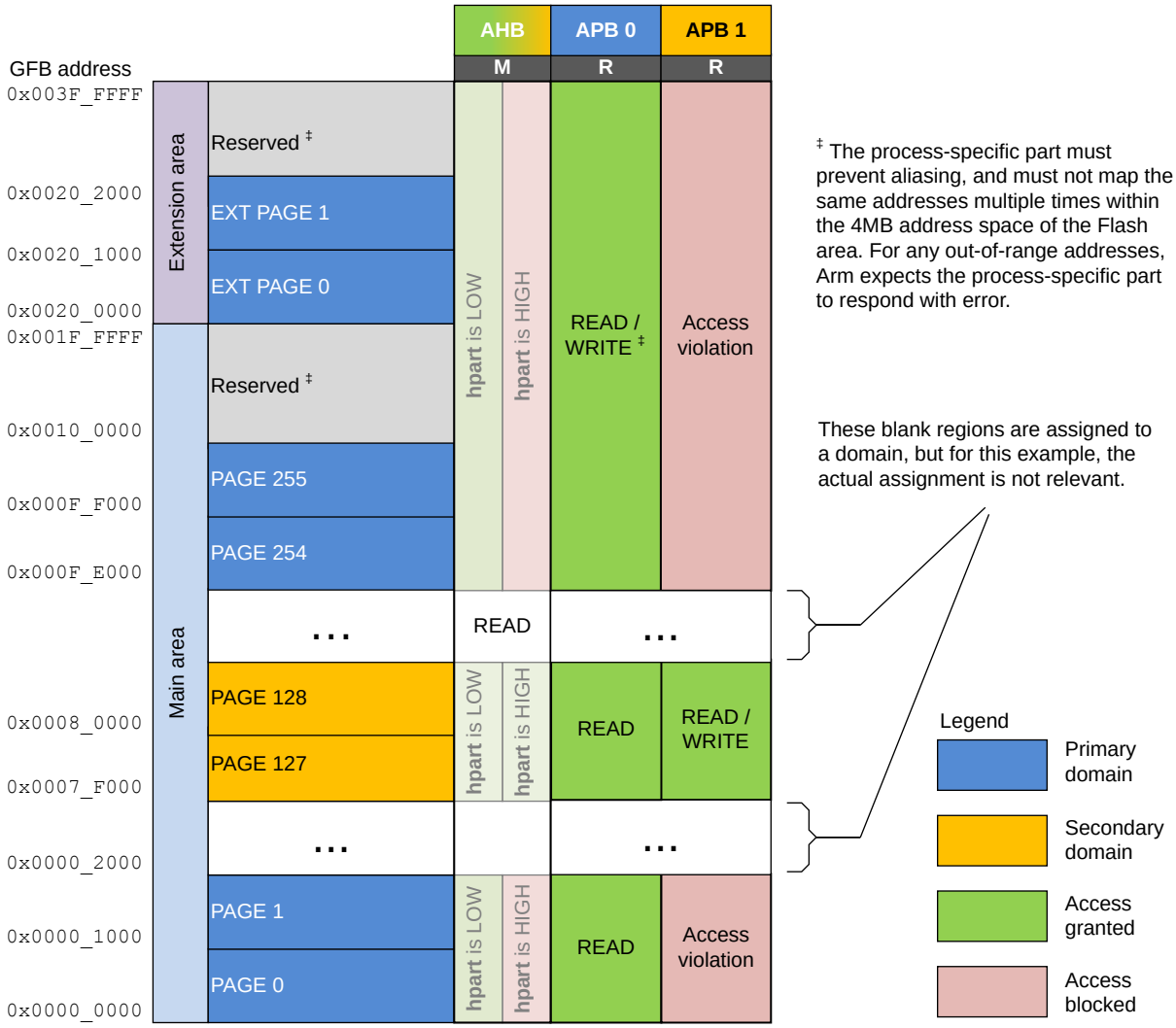
- **partition_ctrl_ro[15:0]** == `0x0000`

Example 4-3: Partition owner with no modification rights

The following figure shows an example partition assignment, where:

- The partition owner does not have modification rights to the partition.
- The partition owner grants read access to the other domain.

Figure 4-4: Partition owner with no modification rights



In this example, the secondary domain has full access to its partition (orange color). The primary domain is granted read access to this partition. For this example, the partition control input signals are:

- **partition_ctrl_rw[n]** == `0b1`
- **partition_ctrl_rd[n]** == `0b0`
- **partition_ctrl_ro[n]** == `0b0`

Also in this example, the primary domain owns the first partition, but it is restricted to read-only access. The secondary domain has no access to the first partition. For this example, the partition control input signals are:

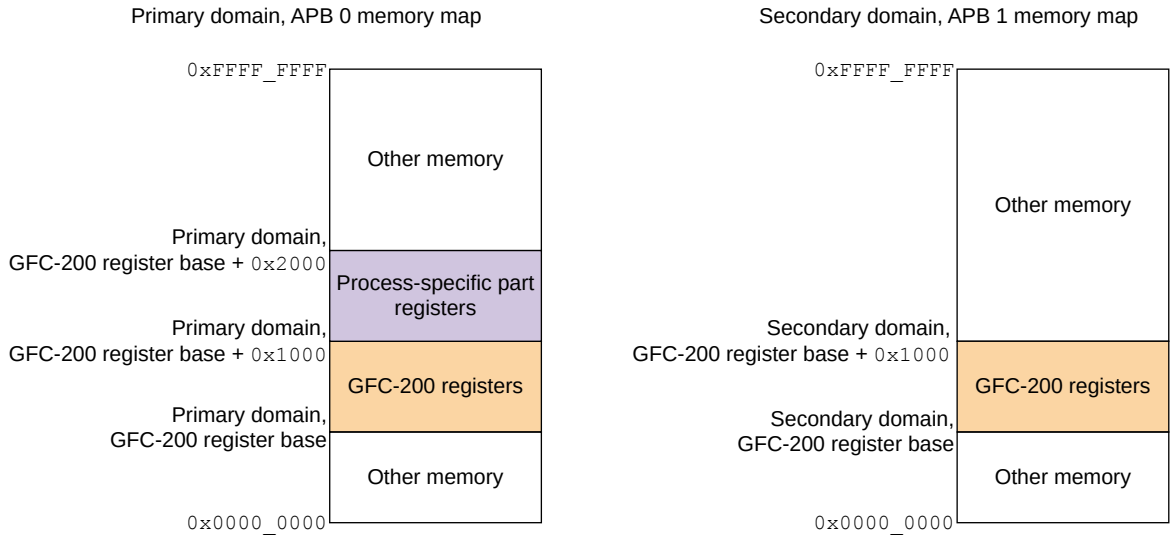
- **partition_ctrl_rw[0]** == 0b0
- **partition_ctrl_rd[0]** == 0b0
- **partition_ctrl_ro[0]** == 0b1

4.2.2 APB memory maps

The system can use either the primary or secondary APB completer interfaces to access a 4KB register space of GFC-200. The system can also use the primary completer interface to access the 4KB register space of the process-specific part.

The following figure shows the memory map for both APB completer interfaces.

Figure 4-5: APB memory map



Each APB interface has a separate 4KB register space that enables an APB domain requester to program the GFC-200 functionality. Some of the GFC-200 registers are shared between the APB interfaces and some registers have identical functionality but are unique to that APB interface. Some registers are available for the primary domain APB interface only.

For the primary domain, the GFC-200 and the process-specific part are visible as two separate peripherals in the system. The benefit of accessing the process-specific part by using the GFC-200 APB 0 interface, is that the GFC-200 is aware of any quiescence requests and can block access to the process-specific part.

The secondary domain cannot access the process-specific part registers but it can access the GFC-200 registers.

4.3 Register summary

The GFC-200 provides a separate 4KB register space for each of the APB interfaces. Some of the GFC-200 registers are shared between the interfaces and some registers have identical functionality but are unique to that APB interface. Some registers are available for the primary domain APB interface only.

The following table shows the registers in offset order from the base memory address. All registers are 32 bits wide.

Table 4-1: Register summary

Offset	Name	Type	Reset	Availability	Description
0x000	IRQ_ENABLE_SET	RW	0x0	A banked register that is unique to each APB interface.	4.4.1 Interrupt enable set register, IRQ_ENABLE_SET on page 40
0x004	IRQ_ENABLE_CLR	RW	0x0		4.4.2 Interrupt enable clear register, IRQ_ENABLE_CLR on page 42
0x008	IRQ_STATUS_SET	RW	0x0		4.4.3 Interrupt status set register, IRQ_STATUS_SET on page 43
0x00C	IRQ_STATUS_CLR	RW	0x0		4.4.4 Interrupt status clear register, IRQ_STATUS_CLR on page 44
0x010	IRQ_MASKED_STATUS	RO	0x0		4.4.5 Interrupt masked status register, IRQ_MASKED_STATUS on page 46
0x014	CTRL	RW	0x0		4.4.6 Control register, CTRL on page 47
0x018	STATUS	RO	0x0		4.4.7 Status register, STATUS on page 49
0x01C	ADDR	RW	0x0		4.4.8 Address register, ADDR on page 50
0x020	DATA0	RW	0x0		4.4.9 Data 0 register, DATA0 on page 51
0x024	DATA1	RO	0x0		4.4.10 Data 1 register, DATA1 on page 52
0x028	DATA2	RO	0x0		4.4.11 Data 2 register, DATA2 on page 53
0x02C	DATA3	RO	0x0		4.4.12 Data 3 register, DATA3 on page 54
0x030	PART_CTRL_RW_STATUS	RO	_[1]	A shared register for both APB interfaces.	4.4.13 Partition ownership status register, PART_CTRL_RW_STATUS on page 54
0x034	PART_CTRL_RO_STATUS	RO	_[1]		4.4.14 Partition control read-only status register, PART_CTRL_RO_STATUS on page 55
0x038	PART_CTRL_RD_STATUS	RO	_[1]		4.4.15 Partition control read status register, PART_CTRL_RD_STATUS on page 56
0x040	PART_CONFIG_MODE_REQ	RW	0x0	This register is only accessible to the primary APB interface.	4.4.16 Partition configuration mode request register, PART_CONFIG_MODE_REQ on page 57
0x044	PART_CONFIG_MODE_STATUS	RO	0x0		4.4.17 Partition configuration mode status register, PART_CONFIG_MODE_STATUS on page 59
0x048	ACCESS_ERR_RESP_CTRL	RW	0x0		4.4.18 Access violation response register, ACCESS_ERR_RESP_CTRL on page 60

[1] A signal or configuration parameters set the reset value of this register. See the register description for more information.

Offset	Name	Type	Reset	Availability	Description
0x04C	ACCESS_ERR_INFO	RO	0x0		4.4.19 Access violation response register, ACCESS_ERR_INFO on page 61
0x050	POWER_STATE	RO	0x8	A shared register for both APB interfaces.	4.4.20 Power state status register, POWER_STATE on page 62
0x054	POWER_STATE_REQ	RW	0x8	A banked register that is unique to each APB interface.	4.4.21 Power state request register, POWER_STATE_REQ on page 63
0x060	HWPARAMS	RO	_[1]	A shared register for both APB interfaces.	4.4.22 Hardware parameters register, HWPARAMS on page 64
0xFD0	PIDR4	RO	0x04		4.4.23 Peripheral ID register 4, PIDR4 on page 66
0xFE0	PIDR0	RO	0x33		4.4.24 Peripheral ID register 0, PIDR0 on page 67
0xFE4	PIDR1	RO	0xB8		4.4.25 Peripheral ID register 1, PIDR1 on page 67
0xFE8	PIDR2	RO	0x0B		4.4.26 Peripheral ID register 2, PIDR2 on page 68
0xFEC	PIDR3	RO	0x00		4.4.27 Peripheral ID register 3, PIDR3 on page 69
0xFF0	CIDR0	RO	0x0D		4.4.28 Component ID register 0, CIDR0 on page 70
0xFF4	CIDR1	RO	0xF0		4.4.29 Component ID register 1, CIDR1 on page 71
0xFF8	CIDR2	RO	0x05		4.4.30 Component ID register 2, CIDR2 on page 72
0xFFC	CIDR3	RO	0xB1		4.4.31 Component ID register 3, CIDR3 on page 73

4.4 Register descriptions

This section describes the GFC-200 registers.

4.3 Register summary on page 39 provides cross references to individual registers.

4.4.1 Interrupt enable set register, IRQ_ENABLE_SET

The IRQ_ENABLE_SET register controls which events can generate an interrupt to the APB requester.

The IRQ_ENABLE_SET register characteristics are:

Usage constraints

Setting a bit to zero has no effect.

Bits[7:6] functionality is accessible to the primary requester only.

Configurations

Available in all configurations. Each APB interface has its own instance of an IRQ_ENABLE_SET register.

Attributes

Offset

0x000

Type

Read/write

Reset

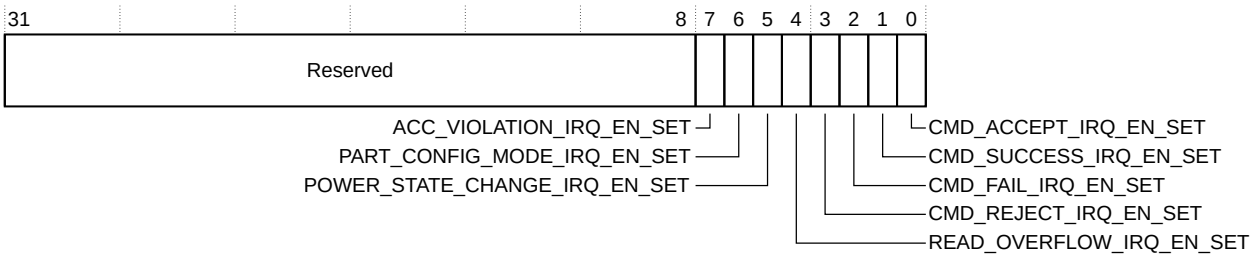
0x0

Width

32

The following figure shows the bit assignments.

Figure 4-6: IRQ_ENABLE_SET register bit assignments



The following table shows the register bit assignments.

Table 4-2: IRQ_ENABLE_SET

Bits	Name	Function
[31:8]	-	Reserved.
[7]	ACC_VIOLATION_IRQ_EN_SET	Set to 1 to enable the ACC_VIOLATION_IRQ bit to generate interrupts on the **irq0** signal. For the secondary requester, this bit is reserved and behaves RAZ/WI.
[6]	PART_CONFIG_MODE_IRQ_EN_SET	Set to 1 to enable the PART_CONFIG_MODE_IRQ bit to generate interrupts on the **irq0** signal. For the secondary requester, this bit is reserved and behaves RAZ/WI.
[5]	PWR_STATE_CHANGE_IRQ_EN_SET	Set to 1 to enable the PWR_STATE_CHANGE_IRQ bit to generate interrupts on the interrupt signal, **irq0** or **irq1**, that belongs to the APB requester.
[4]	READ_OVERFLOW_IRQ_EN_SET	Set to 1 to enable the READ_OVERFLOW_IRQ bit to generate interrupts on the interrupt signal, **irq0** or **irq1**, that belongs to the APB requester.
[3]	CMD_REJECT_IRQ_EN_SET	Set to 1 to enable the CMD_REJECT_IRQ bit to generate interrupts on the interrupt signal, **irq0** or **irq1**, that belongs to the APB requester.
[2]	CMD_FAIL_IRQ_EN_SET	Set to 1 to enable the CMD_FAIL_IRQ bit to generate interrupts on the interrupt signal, **irq0** or **irq1**, that belongs to the APB requester.

Bits	Name	Function
[1]	CMD_SUCCESS_IRQ_EN_SET	Set to 1 to enable the CMD_SUCCESS_IRQ bit to generate interrupts on the interrupt signal, **irq0** or **irq1**, that belongs to the APB requester.
[0]	CMD_ACCEPT_IRQ_EN_SET	Set to 1 to enable the CMD_ACCEPT_IRQ bit to generate interrupts on the interrupt signal, **irq0** or **irq1**, that belongs to the APB requester.

Reading these bits indicates which events are enabled to generate an interrupt.

4.4.2 Interrupt enable clear register, IRQ_ENABLE_CLR

The IRQ_ENABLE_CLR register controls which events cannot generate an interrupt to the APB requester.

The IRQ_ENABLE_CLR register characteristics are:

Usage constraints

Setting a bit to zero has no effect.

Bits[7:6] functionality is accessible to the primary requester only.

Configurations

Available in all configurations. Each APB interface has its own instance of an IRQ_ENABLE_CLR register.

Attributes

Offset

0x004

Type

Read/write

Reset

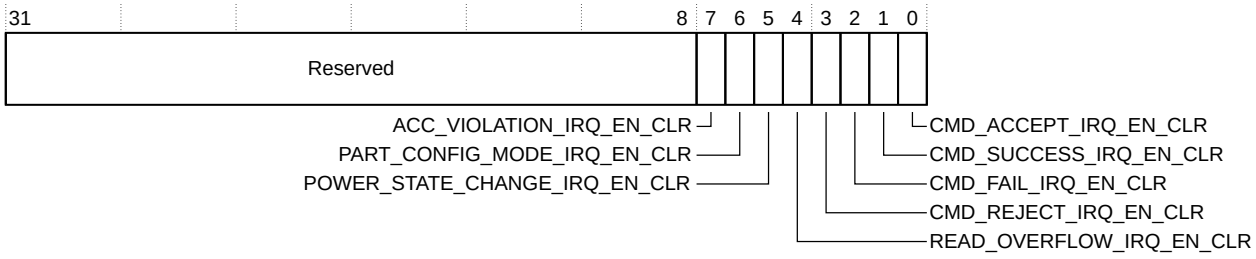
0x0

Width

32

The following figure shows the bit assignments.

Figure 4-7: IRQ_ENABLE_CLR register bit assignments



The following table shows the register bit assignments.

Table 4-3: IRQ_ENABLE_CLR

Bits	Name	Function
[31:8]	-	Reserved.
[7]	ACC_VIOLATION_IRQ_EN_CLR	Set to 1 to disable the ACC_VIOLATION_IRQ bit from generating interrupts on the **irq0** signal. For the secondary requester, this bit is reserved and behaves RAZ/WI.
[6]	PART_CONFIG_MODE_IRQ_EN_CLR	Set to 1 to disable the PART_CONFIG_MODE_IRQ bit from generating interrupts on the **irq0** signal. For the secondary requester, this bit is reserved and behaves RAZ/WI.
[5]	PWR_STATE_CHANGE_IRQ_EN_CLR	Set to 1 to disable the PWR_STATE_CHANGE_IRQ bit from generating interrupts on the interrupt signal, **irq0** or **irq1**, that belongs to the APB requester.
[4]	READ_OVERFLOW_IRQ_EN_CLR	Set to 1 to disable the READ_OVERFLOW_IRQ bit from generating interrupts on the interrupt signal, **irq0** or **irq1**, that belongs to the APB requester.
[3]	CMD_REJECT_IRQ_EN_CLR	Set to 1 to disable the CMD_REJECT_IRQ bit from generating interrupts on the interrupt signal, **irq0** or **irq1**, that belongs to the APB requester.
[2]	CMD_FAIL_IRQ_EN_CLR	Set to 1 to disable the CMD_FAIL_IRQ bit from generating interrupts on the interrupt signal, **irq0** or **irq1**, that belongs to the APB requester.
[1]	CMD_SUCCESS_IRQ_EN_CLR	Set to 1 to disable the CMD_SUCCESS_IRQ bit from generating interrupts on the interrupt signal, **irq0** or **irq1**, that belongs to the APB requester.
[0]	CMD_ACCEPT_IRQ_EN_CLR	Set to 1 to disable the CMD_ACCEPT_IRQ bit from generating interrupts on the interrupt signal, **irq0** or **irq1**, that belongs to the APB requester.

Reading these bits indicates which events are enabled to generate an interrupt.

4.4.3 Interrupt status set register, IRQ_STATUS_SET

The IRQ_STATUS_SET register can set interrupts for multiple interrupt sources, irrespective of whether an interrupt is disabled. You can use this register to test a software interrupt processing routine or for debug purposes. You can also use this register to read the raw, or unmasked, status of the interrupt bits.

The IRQ_STATUS_SET register characteristics are:

Usage constraints

Setting a bit to zero has no effect.

Bits[7:6] functionality is accessible to the primary requester only.

Configurations

Available in all configurations. Each APB interface has its own instance of an IRQ_STATUS_SET register.

Attributes

Offset

0x008

Type

Read/write

Reset

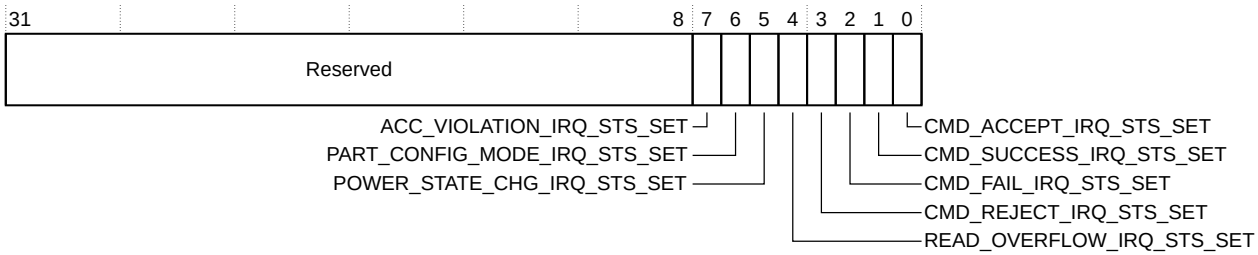
0x0

Width

32

The following figure shows the bit assignments.

Figure 4-8: IRQ_STATUS_SET register bit assignments



The following table shows the register bit assignments.

Table 4-4: IRQ_STATUS_SET

Bits	Name	Function
[31:8]	-	Reserved.
[7]	ACC_VIOLATION_IRQ_STS_SET	Set to 1 to set the ACC_VIOLATION_IRQ bit to 1. For the secondary requester, this bit is reserved and behaves as RAZ/WI.
[6]	PART_CONFIG_MODE_IRQ_STS_SET	Set to 1 to set the PART_CONFIG_MODE_IRQ bit to 1. For the secondary requester, this bit is reserved and behaves as RAZ/WI.
[5]	PWR_STATE_CHG_IRQ_STS_SET	Set to 1 to set the PWR_STATE_CHANGE_IRQ bit to 1.
[4]	READ_OVERFLOW_IRQ_STS_SET	Set to 1 to set the READ_OVERFLOW_IRQ bit to 1.
[3]	CMD_REJECT_IRQ_STS_SET	Set to 1 to set the CMD_REJECT_IRQ bit to 1.
[2]	CMD_FAIL_IRQ_STS_SET	Set to 1 to set the CMD_FAIL_IRQ bit to 1.
[1]	CMD_SUCCESS_IRQ_STS_SET	Set to 1 to set the CMD_SUCCESS_IRQ bit to 1.
[0]	CMD_ACCEPT_IRQ_STS_SET	Set to 1 to set the CMD_ACCEPT_IRQ bit to 1.

Reading these bits provides the status of the interrupt bits.

The IRQ_MASKED_STATUS also provides the status of the interrupt bits. However, the values that IRQ_MASKED_STATUS returns might differ from the IRQ_STATUS_SET value, because the IRQ_MASKED_STATUS value depends on whether an interrupt is enabled.

4.4.4 Interrupt status clear register, IRQ_STATUS_CLR

The IRQ_STATUS_CLR register can clear interrupts for multiple interrupt sources. You can also use this register to read the raw, or unmasked, status of the interrupt bits.

The IRQ_STATUS_CLR register characteristics are:

Usage constraints

Setting a bit to zero has no effect.

Bits[7:6] functionality is accessible to the primary requester only.

Configurations

Available in all configurations. Each APB interface has its own instance of an IRQ_STATUS_CLR register.

Attributes

Offset

0x00C

Type

Read/write

Reset

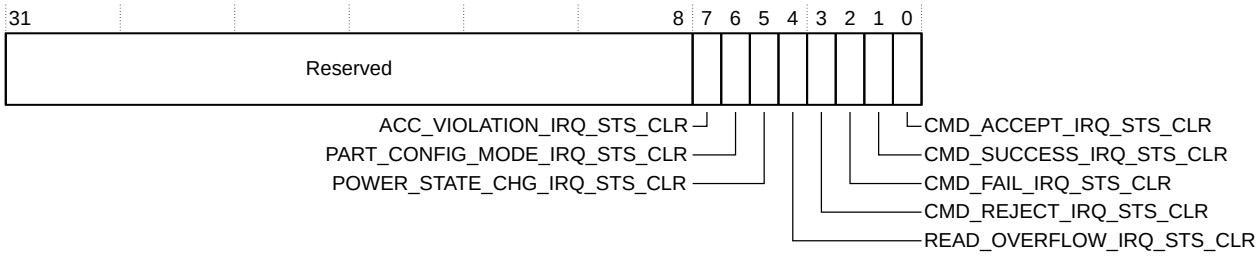
0x0

Width

32

The following figure shows the bit assignments.

Figure 4-9: IRQ_STATUS_CLR register bit assignments



The following table shows the register bit assignments.

Table 4-5: IRQ_STATUS_CLR

Bits	Name	Function
[31:8]	-	Reserved.
[7]	ACC_VIOLATION_IRQ_STS_CLR	Set to 1 to clear the ACC_VIOLATION_IRQ bit to 0. For the secondary requester, this bit is reserved and behaves as RAZ/WI.

Bits	Name	Function
[6]	PART_CONFIG_MODE_IRQ_STS_CLR	Set to 1 to clear the PART_CONFIG_MODE_IRQ bit to 0. For the secondary requester, this bit is reserved and behaves as RAZ/WI.
[5]	PWR_STATE_CHG_IRQ_STS_CLR	Set to 1 to clear the PWR_STATE_CHANGE_IRQ bit to 0.
[4]	READ_OVERFLOW_IRQ_STS_CLR	Set to 1 to clear the READ_OVERFLOW_IRQ bit to 0.
[3]	CMD_REJECT_IRQ_STS_CLR	Set to 1 to clear the CMD_REJECT_IRQ bit to 0.
[2]	CMD_FAIL_IRQ_STS_CLR	Set to 1 to clear the CMD_FAIL_IRQ bit to 0.
[1]	CMD_SUCCESS_IRQ_STS_CLR	Set to 1 to clear the CMD_SUCCESS_IRQ bit to 0.
[0]	CMD_ACCEPT_IRQ_STS_CLR	Set to 1 to clear the CMD_ACCEPT_IRQ bit to 0.

Reading these bits provides the status of the interrupt bits.

The IRQ_MASKED_STATUS also provides the status of the interrupt bits. However, the values that IRQ_MASKED_STATUS returns might differ from the IRQ_STATUS_CLR value, because the IRQ_MASKED_STATUS value depends on whether an interrupt is enabled.

4.4.5 Interrupt masked status register, IRQ_MASKED_STATUS

The IRQ_MASKED_STATUS register shows the status of the interrupt bits. The value that it returns depends on whether an interrupt is enabled. Software can use this register to discover the cause of an interrupt assertion for all enabled interrupts.

The IRQ_MASKED_STATUS register characteristics are:

Usage constraints

Bits[7:6] functionality is accessible to the primary requester only.

Configurations

Available in all configurations. Each APB interface has its own instance of an IRQ_MASKED_STATUS register.

Attributes

Offset

`0x010`

Type

Read only.

Reset

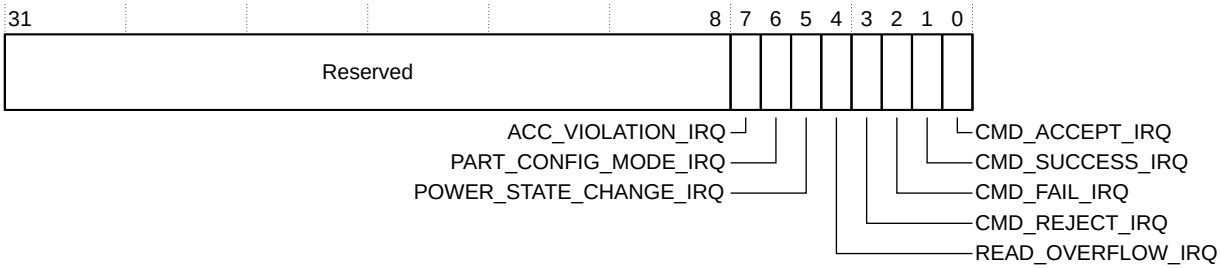
`0x0`

Width

32

The following figure shows the bit assignments.

Figure 4-10: IRQ_MASKED_STATUS register bit assignments



The following table shows the register bit assignments.

Table 4-6: IRQ_MASKED_STATUS

Bits	Name	Function
[31:8]	-	Reserved.
[7]	ACC_VIOLATION_IRQ	If this bit returns 1, the ACC_VIOLATION_IRQ is the cause of the interrupt assertion. For the secondary requester, this bit is reserved and behaves as RAZ.
[6]	PART_CONFIG_MODE_IRQ	If this bit returns 1, the PART_CONFIG_MODE_IRQ is the cause of the interrupt assertion. For the secondary requester, this bit is reserved and behaves as RAZ.
[5]	PWR_STATE_CHANGE_IRQ	If this bit returns 1, the PWR_STATE_CHANGE_IRQ is the cause of the interrupt assertion.
[4]	READ_OVERFLOW_IRQ	If this bit returns 1, the READ_OVERFLOW_IRQ is the cause of the interrupt assertion.
[3]	CMD_REJECT_IRQ	If this bit returns 1, the CMD_REJECT_IRQ is the cause of the interrupt assertion.
[2]	CMD_FAIL_IRQ	If this bit returns 1, the CMD_FAIL_IRQ is the cause of the interrupt assertion.
[1]	CMD_SUCCESS_IRQ	If this bit returns 1, the CMD_SUCCESS_IRQ is the cause of the interrupt assertion.
[0]	CMD_ACCEPT_IRQ	If this bit returns 1, the CMD_ACCEPT_IRQ is the cause of the interrupt assertion.

To obtain the raw unmasked interrupt status, you can read the IRQ_STATUS_SET or IRQ_STATUS_CLR register.

4.4.6 Control register, CTRL

The CTRL register initiates Flash accesses and can abort an active command for that APB interface.

The CTRL register characteristics are:

Usage constraints

There are no usage constraints.

Configurations

Available in all configurations. Each APB interface has its own instance of a CTRL register.

Attributes

Offset

`0x014`

Type

Read/write

Reset

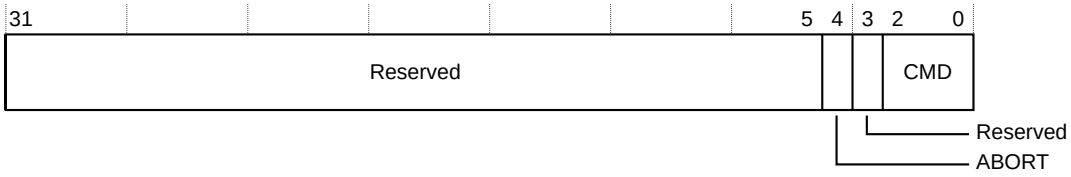
0x0

Width

32

The following figure shows the bit assignments.

Figure 4-11: CTRL register bit assignments



The following table shows the register bit assignments.

Table 4-7: CTRL

Bits	Name	Function
[31:5]	-	Reserved.
[4]	ABORT	Set to 1 to abort an ongoing command on the GFB interface that originated from this APB interface. ABORT is effective only when STATUS.CMD_ACCEPT == 1. **Note:** You might typically use this feature to abort commands that take many clock cycles to complete such as the ERASE and MASS ERASE commands. When ABORT is set to 1, the GFC-200 ignores the CMD field. Reading the ABORT bit shows whether the APB requester has requested an abort for the ongoing command.
[3]	-	Reserved.
[2:0]	CMD	Initiates a command to the embedded Flash, using the address in the ADDR register and the data in the DATA0 register. The commands are: 0b001 READ. 0b010 WRITE. Software must ensure that the addressed region in Flash is erased before the WRITE occurs. 0b011 ROW WRITE. Software must ensure that the addressed region in Flash is erased before the ROW WRITE occurs. 0b100 ERASE. 0b111 MASS ERASE. 0b000, 0b101, 0b110 Reserved. Writing reserved commands has no effect. When the GFC-200 accepts a command, it clears the CMD field to 0b000 and sets STATUS.CMD_ACCEPT = 1. Reading the CMD field shows the pending GFB command.

Related information

GFB interface on page 22

4.4.7 Status register, STATUS

The STATUS register returns the state of the embedded Flash accesses for that APB interface.

The STATUS register characteristics are:

Usage constraints

There are no usage constraints.

Configurations

Available in all configurations. Each APB interface has its own instance of a STATUS register.

Attributes

Offset

`0x018`

Type

Read only

Reset

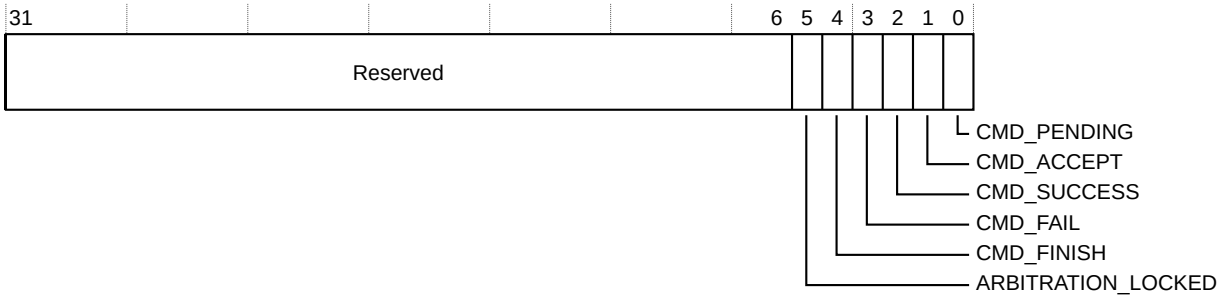
`0x0`

Width

32

The following figure shows the bit assignments.

Figure 4-12: STATUS register bit assignments



The following table shows the register bit assignments.

Table 4-8: STATUS

Bits	Name	Function
[31:6]	-	Reserved.

Bits	Name	Function
[5]	ARBITRATION_LOCKED	When this bit returns 1, it indicates that the AHB interface has the arbitration lock, and no APB commands are serviced until the lock is removed. Software can use this bit to detect why an APB request is stalled.
[4]	CMD_FINISH	When this bit returns 1, it indicates that the previously accepted command has finished. The GFC-200 clears this bit to 0, when software clears the CMD_SUCCESS_IRQ or CMD_FAIL_IRQ status bits by writing to the IRQ_STATUS_CLR register.
[3]	CMD_FAIL	When this bit returns 1, it indicates that the previously accepted command has finished with a failure. The GFC-200 sets this bit to 1 when CMD_FAIL_IRQ == 1. The GFC-200 clears this bit to 0, when software sets IRQ_STATUS_CLR.CMD_FAIL_IRQ_STS_CLR = 1, which clears the CMD_FAIL_IRQ bit.
[2]	CMD_SUCCESS	When this bit returns 1, it indicates that the previously accepted command has finished successfully. The GFC-200 sets this bit to 1 when CMD_SUCCESS_IRQ == 1. The GFC-200 clears this bit to 0, when software sets IRQ_STATUS_CLR.CMD_SUCCESS_IRQ_STS_CLR = 1, which clears the CMD_SUCCESS_IRQ bit.
[1]	CMD_ACCEPT	When this bit returns 1, it indicates that the embedded Flash accepted the command. The GFC-200 clears this bit to 0, when software clears the CMD_SUCCESS_IRQ or CMD_FAIL_IRQ status bits by writing to the IRQ_STATUS_CLR register. However, if the GFC-200 receives a command before software clears the status bits, then CMD_ACCEPT remains set after software clears the status bits.
[0]	CMD_PENDING	When the CTRL register is written, the command goes into the arbitration queue and waits to be arbitrated towards the embedded Flash. The GFC-200 sets this bit when either: • The command is initiated, but still pending in the queue. • Software sets CTRL.ABORT = 1. The GFC-200 clears this bit when either: • The embedded Flash arbitrates and accepts the command. • A command is aborted.

4.4.8 Address register, ADDR

The ADDR register contains the address for a read or write access to the embedded Flash.

The ADDR register characteristics are:

Usage constraints

The value that is written to the ADDR field must be either:

- 32-bit aligned for write accesses to the embedded Flash.
- 128-bit aligned for read accesses from the embedded Flash.

Configurations

Available in all configurations. Each APB interface has its own instance of an ADDR register.

Attributes

Offset

0x01C

Type

Read/write

Reset

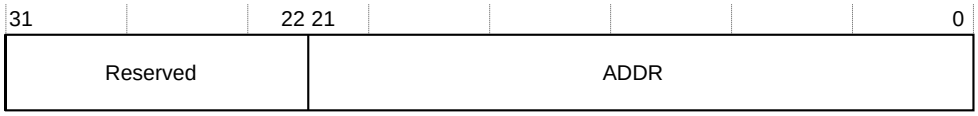
0x0

Width

32

The following figure shows the bit assignments.

Figure 4-13: ADDR register bit assignments



The following table shows the register bit assignments.

Table 4-9: ADDR

Bits	Name	Function
[31:22]	-	Reserved.
[21:0]	ADDR	The address for the current Flash access.

4.4.9 Data 0 register, DATA0

The DATA0 register contains data bits[31:0], for a read or write access to the embedded Flash.

The DATA0 register characteristics are:

Usage constraints

There are no usage constraints.

Configurations

Available in all configurations. Each APB interface has its own instance of a DATA0 register.

Attributes

Offset

0x020

Type

Read/write

Reset

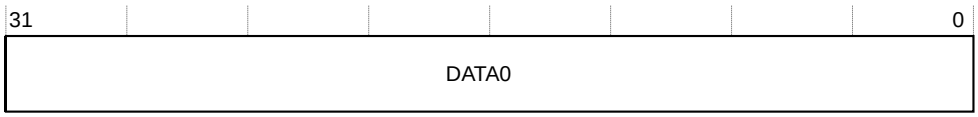
0x0

Width

32

The following figure shows the bit assignments.

Figure 4-14: DATA0 register bit assignments



The following table shows the register bit assignments.

Table 4-10: DATA0

Bits	Name	Function
[31:0]	DATA0	For reads from the embedded Flash, this field returns the read data bits[31:0]. For writes to the embedded Flash, this field contains the write data bits[31:0].

4.4.10 Data 1 register, DATA1

The DATA1 register contains data bits[63:32], for a read or write access to the embedded Flash.

The DATA1 register characteristics are:

Usage constraints

There are no usage constraints.

Configurations

Available in configurations where *FWDATA_WIDTH* > 32. When present, each APB interface has its own instance of a DATA1 register.

Attributes

Offset

0x024

Type

Read/write

Reset

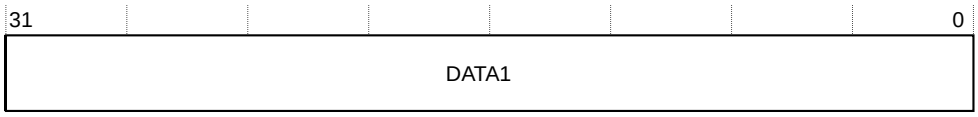
0x0

Width

32

The following figure shows the bit assignments.

Figure 4-15: DATA1 register bit assignments



The following table shows the register bit assignments.

Table 4-11: DATA1

Bits	Name	Function
[31:0]	DATA1	For reads from the embedded Flash, this field returns the read data bits[63:32]. For writes to the embedded Flash, this field contains the write data bits[63:32].

4.4.11 Data 2 register, DATA2

The DATA2 register contains data bits[95:63], for a read or write access to the embedded Flash.

The DATA2 register characteristics are:

Usage constraints

There are no usage constraints.

Configurations

Available in configurations where *FWDATA_WIDTH* > 64. When present, each APB interface has its own instance of a DATA2 register.

Attributes

Offset

0x028

Type

Read/write

Reset

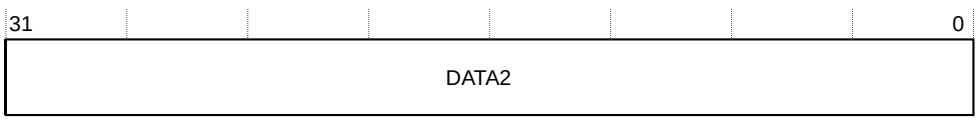
0x0

Width

32

The following figure shows the bit assignments.

Figure 4-16: DATA2 register bit assignments



The following table shows the register bit assignments.

Table 4-12: DATA2

Bits	Name	Function
[31:0]	DATA2	For reads from the embedded Flash, this field returns the read data bits[95:63]. For writes to the embedded Flash, this field contains the write data bits[95:63].

4.4.12 Data 3 register, DATA3

The DATA3 register contains data bits[127:96], for a read or write access to the embedded Flash.

The DATA3 register characteristics are:

Usage constraints

There are no usage constraints.

Configurations

Available in configurations where *FWDATA_WIDTH* > 64. When present, each APB interface has its own instance of a DATA3 register.

Attributes

Offset

0x02C

Type

Read/write

Reset

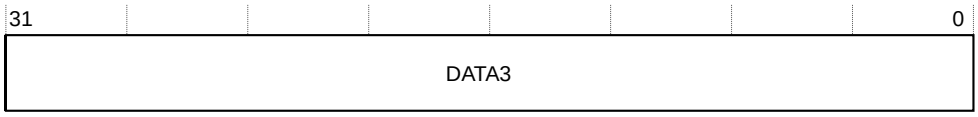
0x0

Width

32

The following figure shows the bit assignments.

Figure 4-17: DATA3 register bit assignments



The following table shows the register bit assignments.

Table 4-13: DATA3

Bits	Name	Function
[31:0]	DATA3	For reads from the embedded Flash, this field returns the read data bits[127:96]. For writes to the embedded Flash, this field contains the write data bits[127:96].

4.4.13 Partition ownership status register, PART_CTRL_RW_STATUS

The PART_CTRL_RW_STATUS register returns whether the primary or secondary domain is the owner of a partition. There are a maximum of 16 partitions.

The PART_CTRL_RW_STATUS register characteristics are:

Usage constraints

There are no usage constraints.

Configurations

Available in all configurations.

Attributes

Offset

0x030

Type

Read only.

Reset

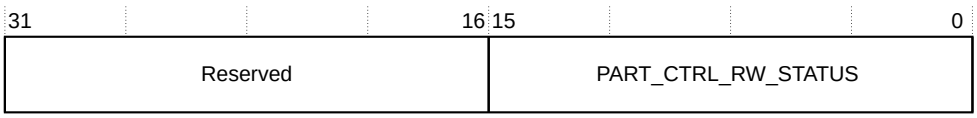
The value of the **partition_ctrl_rw[15:0]** signal. The GFC-200 reads this signal when it exits reset and while it is in partition configuration mode.

Width

32

The following figure shows the bit assignments.

Figure 4-18: PART_CTRL_RW_STATUS register bit assignments



The following table shows the register bit assignments.

Table 4-14: PART_CTRL_RW_STATUS

Bits	Name	Function
[31:16]	-	Reserved.
[15:0]	PART_CTRL_RW_STATUS	Each bit indicates which domain is the owner of the corresponding partition: Bit[x] == 0 The primary domain is the owner of partition x. Bit[x] == 1 The secondary domain is the owner of partition x. **Note:** The owner of a partition has read/write access to that partition unless the PART_CTRL_RO_STATUS register indicates that the partition is read only.

4.4.14 Partition control read-only status register, PART_CTRL_RO_STATUS

The PART_CTRL_RO_STATUS register returns whether the system configures a partition to be read only. There are a maximum of 16 partitions.

The PART_CTRL_RO_STATUS register characteristics are:

Usage constraints

There are no usage constraints.

Configurations

Available in all configurations.

Attributes

Offset

`0x034`

Type

Read only.

Reset

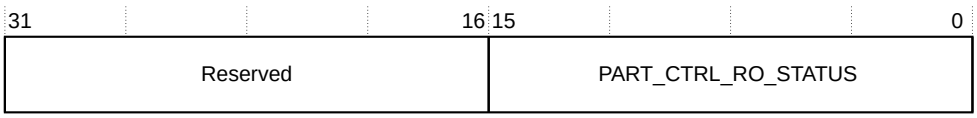
The value of the **partition_ctrl_ro[15:0]** signal. The GFC-200 reads this signal when it exits reset and while it is in partition configuration mode.

Width

32

The following figure shows the bit assignments.

Figure 4-19: PART_CTRL_RO_STATUS register bit assignments



The following table shows the register bit assignments.

Table 4-15: PART_CTRL_RO_STATUS

Bits	Name	Function
[31:16]	-	Reserved.
[15:0]	PART_CTRL_RO_STATUS	Each bit indicates if the system configures the corresponding partition as read only: Bit[x] == 0 Partition x is not read only, so the owner of the partition has read/write access. Bit[x] == 1 Partition x is read only.

4.4.15 Partition control read status register, PART_CTRL_RD_STATUS

The PART_CTRL_RD_STATUS register returns whether the system grants read access to the non-owner of a partition. There are a maximum of 16 partitions.

The PART_CTRL_RD_STATUS register characteristics are:

Usage constraints

There are no usage constraints.

Configurations

Available in all configurations.

Attributes

Offset

0x038

Type

Read only.

Reset

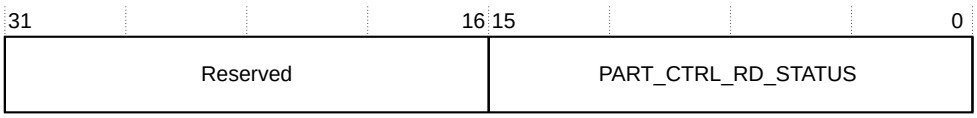
The value of the **partition_ctrl_rd[15:0]** signal. The GFC-200 reads this signal when it exits reset and while it is in partition configuration mode.

Width

32

The following figure shows the bit assignments.

Figure 4-20: PART_CTRL_RD_STATUS register bit assignments



The following table shows the register bit assignments.

Table 4-16: PART_CTRL_RD_STATUS

Bits	Name	Function
[31:16]	-	Reserved.
[15:0]	PART_CTRL_RD_STATUS	Each bit indicates if the system enables the non-owner of a partition to read the corresponding partition: Bit[x] == 0 The system does not permit the non-owner of a partition to read partition x. Therefore, only the owner of the partition has access to partition x. Bit[x] == 1 The system enables the non-owner of a partition to read partition x.

4.4.16 Partition configuration mode request register, PART_CONFIG_MODE_REQ

The PART_CONFIG_MODE_REQ register enables software to put the GFC-200 in partition configuration mode. You can use this mode to change the partition permissions, which might be necessary during a system firmware update.

The PART_CONFIG_MODE_REQ register characteristics are:

Usage constraints

Accessible to the primary APB requester only.

Configurations

Available in all configurations. This register is present in the primary APB interface register space only.

Attributes

Offset

0x040

Type

Read/write.

Reset

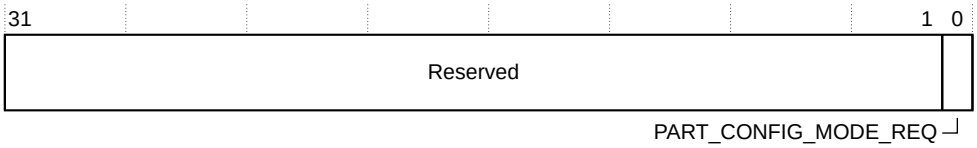
0x0.

Width

32

The following figure shows the bit assignments.

Figure 4-21: PART_CONFIG_MODE_REQ register bit assignments



The following table shows the register bit assignments.

Table 4-17: PART_CONFIG_MODE_REQ

Bits	Name	Function
[31:1]	-	Reserved.
[0]	PART_CONFIG_MODE_REQ	Set to 1, to request that the GFC-200 enters partition configuration mode. When the GFC-200 enters partition configuration mode, it notifies the primary manager by setting the **irq0** interrupt HIGH. When software services the interrupt, it must clear the interrupt by setting IRQ_STATUS_CLR.PART_CONFIG_MODE_IRQ_STS_CLR = 1. Set to 0, to request that the GFC-200 exits partition configuration mode.

Related information

Reconfiguring the partition access rights on page 80

4.4.17 Partition configuration mode status register, PART_CONFIG_MODE_STATUS

The PART_CONFIG_MODE_STATUS register returns whether the GFC-200 is in partition configuration mode.

The PART_CONFIG_MODE_STATUS register characteristics are:

Usage constraints

Accessible to the primary APB requester only.

Configurations

Available in all configurations. This register is present in the primary APB interface register space only.

Attributes

Offset

`0x044`

Type

Read only.

Reset

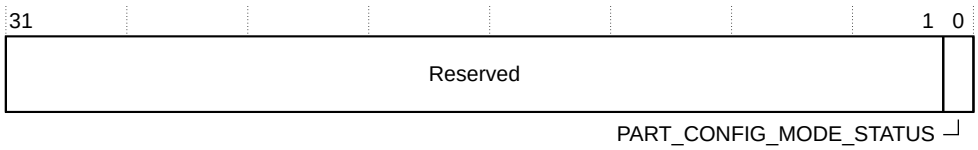
`0x0`.

Width

32

The following figure shows the bit assignments.

Figure 4-22: PART_CONFIG_MODE_STATUS register bit assignments



The following table shows the register bit assignments.

Table 4-18: PART_CONFIG_MODE_STATUS

Bits	Name	Function
[31:1]	-	Reserved.

Bits	Name	Function
[0]	PART_CONFIG_MODE_STATUS	Indicates whether the GFC-200 is in partition configuration mode: 0 == The GFC-200 is not in partition configuration mode. 1 == The GFC-200 is in partition configuration mode.

4.4.18 Access violation response register, ACCESS_ERR_RESP_CTRL

The ACCESS_ERR_RESP_CTRL register controls whether the GFC-200 responds with an error response or RAZ/WI, if the primary domain performs an access violation.

The ACCESS_ERR_RESP_CTRL register characteristics are:

Usage constraints

Accessible to the primary APB requester only.

Configurations

Available in all configurations. This register is present in the primary APB interface register space only.

Attributes

Offset

0x048

Type

Read/write.

Reset

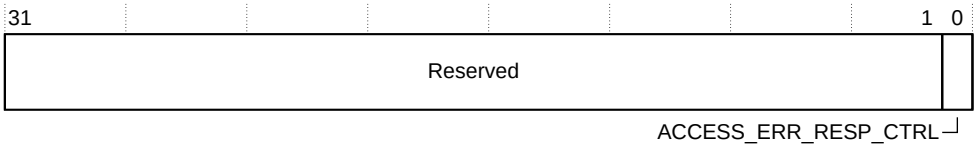
0x0.

Width

32

The following figure shows the bit assignments.

Figure 4-23: ACCESS_ERR_RESP_CTRL register bit assignments



The following table shows the register bit assignments.

Table 4-19: ACCESS_ERR_RESP_CTRL

Bits	Name	Function
[31:1]	-	Reserved.

Bits	Name	Function
[0]	ACCESS_ERR_RESP_CTRL	This bit controls the type of error response that the GFC-200 provides to the primary domain: • 0 = Error. If the primary domain performs an: **AHB read access violation** The GFC-200 returns a 2-cycle ERROR response to the AHB manager. **APB access violation** The GFC-200 sets CMD_FAIL_IRQ to 1. The GFC-200 does not return an APB bus error. • 1 = RAZ/WI. If the primary domain performs a: **Read access violation** The GFC-200 returns the read data as all zeros. **Write access violation** The GFC-200 ignores the write transfer. **Note:** If the secondary domain performs any access violation, the GFC-200 always provides an error response.

4.4.19 Access violation response register, ACCESS_ERR_INFO

The ACCESS_ERR_INFO register returns the embedded Flash address location for the most recent GFB access violation by either domain.

The ACCESS_ERR_INFO register characteristics are:

Usage constraints

Accessible to the primary APB requester only.

Configurations

Available in all configurations. This register is present in the primary APB interface register space only.

Attributes

Offset

`0x04C`

Type

Read only.

Reset

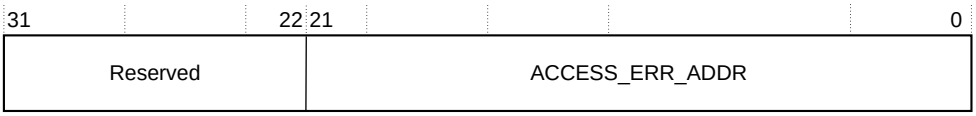
`0x0`.

Width

32

The following figure shows the bit assignments.

Figure 4-24: ACCESS_ERR_INFO register bit assignments



The following table shows the register bit assignments.

Table 4-20: ACCESS_ERR_INFO

Bits	Name	Function
[31:22]	-	Reserved.
[21:0]	ACCESS_ERR_ADDR	Returns the 22-bit GFB address for the most recent access violation by either domain.

4.4.20 Power state status register, POWER_STATE

The POWER_STATE register returns the Flash macro power state and whether the Flash macro is in low-voltage mode.

The POWER_STATE register characteristics are:

Usage constraints

There are no usage constraints.

Configurations

Available in all configurations.

Attributes

Offset

0x050

Type

Read only.

Reset

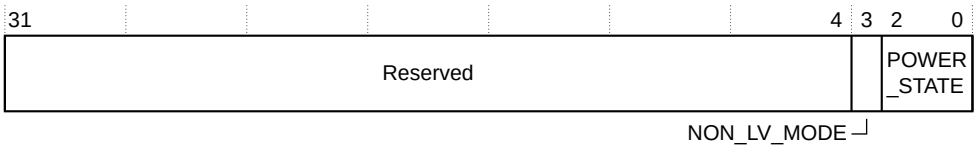
0x8.

Width

32

The following figure shows the bit assignments.

Figure 4-25: POWER_STATE register bit assignments



The following table shows the register bit assignments.

Table 4-21: POWER_STATE

Bits	Name	Function
[31:4]	-	Reserved.
[3]	NON_LV_MODE	This bit indicates whether the Flash macro is in low-voltage mode: 0 == The Flash macro is in low-voltage mode. 1 == The Flash macro is on and is not in low-voltage mode.
[2:0]	POWER_STATE	This field indicates the current Flash macro power state: `0b000` == OFF. The Flash macro is powered off. `0b001` == PD. The Flash macro is in powerdown mode. `0b010` == SL. The Flash macro is in sleep mode. `0b100` == ON. The Flash macro is fully powered on.

4.4.21 Power state request register, POWER_STATE_REQ

The POWER_STATE_REQ register enables software to set its minimum required Flash macro power state and its minimum required Flash macro operating mode.

The POWER_STATE_REQ register characteristics are:

Usage constraints

There are banked copies of this register between the two APB interfaces. The GFC-200 compares the requirements from both APB interfaces and selects the higher requirements.

Configurations

Available in all configurations. Each APB interface has its own instance of a POWER_STATE_REQ register.

Attributes

Offset

`0x054`

Type

Read/write.

Reset

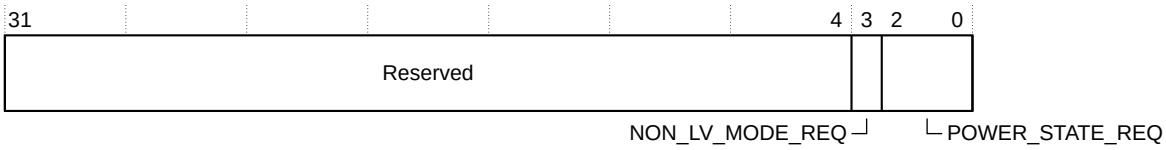
`0x8`.

Width

32

The following figure shows the bit assignments.

Figure 4-26: POWER_STATE_REQ register bit assignments



The following table shows the register bit assignments.

Table 4-22: POWER_STATE_REQ

Bits	Name	Function
[31:4]	-	Reserved.
[3]	NON_LV_MODE_REQ	This bit controls whether the corresponding APB domain allows the Flash macro to operate in low-voltage mode: 0 == The Flash macro can operate in low-voltage mode. 1 == The Flash macro must operate in Full-On mode and must not use low-voltage mode.
[2:0]	POWER_STATE_REQ	The corresponding APB domain programs this field to set the minimum required Flash macro power: `0b000` == OFF. The Flash macro can power off. `0b001` == PD. The Flash macro can enter powerdown mode. `0b01x` == SL. The Flash macro can enter sleep mode. `0b1xx` == ON. The Flash macro must remain fully powered on. Where `x` is a don't care value.

Related information

Flash macro power control on page 81

4.4.22 Hardware parameters register, HWPARAMS

The HWPARAMS register returns the values of the silicon configuration parameters that are set during the product design phase. You can read this register to discover the data widths of the GFB and AHB interfaces, and the size of the Flash partition.

The HWPARAMS register characteristics are:

Usage constraints

There are no usage constraints.

Configurations

Available in all configurations.

Attributes

Offset

`0x060`

Type

Read only.

Reset

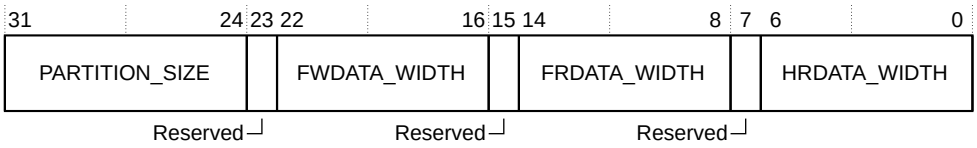
Implementation defined.

Width

32

The following figure shows the bit assignments.

Figure 4-27: HWPARAMS register bit assignments



The following table shows the register bit assignments.

Table 4-23: HWPARAMS

Bits	Name	Function
[31:24]	PARTITION_SIZE	Returns the value of *PARTITION_SIZE* − 1. The possible values are: 0x0F == 16KB Flash partition size. 0x1F == 32KB Flash partition size. 0x3F == 64KB Flash partition size. 0x7F == 128KB Flash partition size. 0xFF == 256KB Flash partition size.
[23]	-	Reserved, returns 0.
[22:16]	FWDATA_WIDTH	Returns the value of *FWDATA_WIDTH* − 1. The possible values are: 0b0011111 == 32-bit GFB write data bus width. 0b0111111 == 64-bit GFB write data bus width. 0b1111111 == 128-bit GFB write data bus width.
[15]	-	Reserved, returns 0.
[14:8]	FRDATA_WIDTH	Returns the value of *FRDATA_WIDTH* − 1. The possible values are: 0b0011111 == 32-bit GFB read data bus width. 0b0111111 == 64-bit GFB read data bus width. 0b1111111 == 128-bit GFB read data bus width.
[7]	-	Reserved, returns 0.

Bits	Name	Function
[6:0]	HRDATA_WIDTH	Returns the value of *HRDATA_WIDTH* – 1. The possible values are: `0b0011111` == 32-bit AHB read data bus width. `0b0111111` == 64-bit AHB read data bus width. `0b1111111` == 128-bit AHB read data bus width.

4.4.23 Peripheral ID register 4, PIDR4

The PIDR4 register returns byte[4] of the peripheral identifier. A debugger during system discovery can use the peripheral ID to discover which peripherals are in the system and the size of the programming register space.

The PIDR4 register characteristics are:

Usage constraints

There are no usage constraints.

Configurations

Available in all configurations.

Attributes

Offset

`0xFD0`

Type

Read only.

Reset

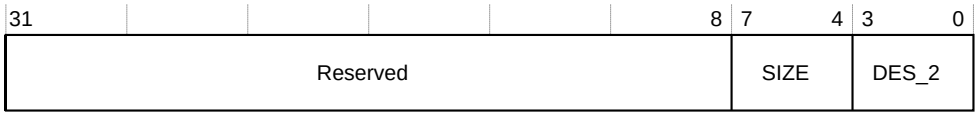
`0x4`

Width

32

The following figure shows the bit assignments.

Figure 4-28: PIDR4 register bit assignments



The following table shows the register bit assignments.

Table 4-24: PIDR4

Bits	Name	Function
[31:8]	-	Reserved, returns 0.

Bits	Name	Function
[7:4]	SIZE	Returns 0, which indicates that the GFC-200 registers occupy a single 4KB page.
[3:0]	DES_0	Indicates how many Continuation Codes (`0x7F`) an Arm device requires. For identifying an Arm device or product, the *Standard Manufacturer's Identification Code* specifies a requirement of four Continuation Codes.

4.4.24 Peripheral ID register 0, PIDR0

The PIDR0 register returns byte[0] of the peripheral identifier. A debugger during system discovery can use the peripheral ID to discover which peripherals are in the system.

The PIDR0 register characteristics are:

Usage constraints

There are no usage constraints.

Configurations

Available in all configurations.

Attributes

Offset

`0xFE0`

Type

Read only.

Reset

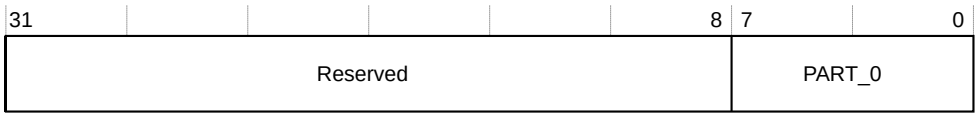
`0x33`

Width

32

The following figure shows the bit assignments.

Figure 4-29: PIDR0 register bit assignments



The following table shows the register bit assignments.

Table 4-25: PIDR0

Bits	Name	Function
[31:8]	-	Reserved, returns 0.
[7:0]	PART_0	Part number, bits[7:0], for the GFC-200. See also PIDR1.PART_1. The GFC-200 part number is 833.

4.4.25 Peripheral ID register 1, PIDR1

The PIDR1 register returns byte[1] of the peripheral identifier. A debugger during system discovery can use the peripheral ID to discover which peripherals are in the system.

The PIDR1 register characteristics are:

Usage constraints

There are no usage constraints.

Configurations

Available in all configurations.

Attributes

Offset

`0xFE4`

Type

Read only.

Reset

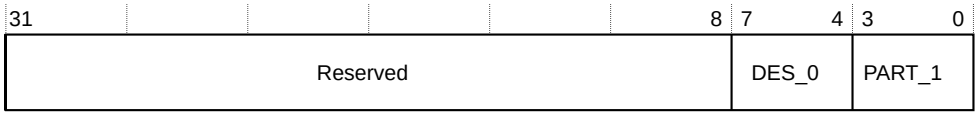
`0xB8`

Width

32

The following figure shows the bit assignments.

Figure 4-30: PIDR1 register bit assignments



The following table shows the register bit assignments.

Table 4-26: PIDR1

Bits	Name	Function
[31:8]	-	Reserved, returns 0.
[7:4]	DES_0	The JEDEC JEP106 ID code [3:0], which identifies Arm as the designer of the GFC-200. See also PIDR2.DES_1 and the *Standard Manufacturer's Identification Code*.
[3:0]	PART_1	Part number, bits[11:8], for the GFC-200. See also PIDR0.PART_0. The GFC-200 part number is 833.

4.4.26 Peripheral ID register 2, PIDR2

The PIDR2 register returns byte[2] of the peripheral identifier. A debugger during system discovery can use the peripheral ID to discover which peripherals are in the system and its `rmpn` revision status.

The PIDR2 register characteristics are:

Usage constraints

There are no usage constraints.

Configurations

Available in all configurations.

Attributes

Offset

`0xFE8`

Type

Read only.

Reset

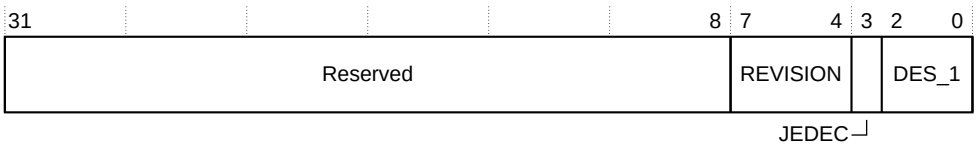
`0x0B`

Width

32

The following figure shows the bit assignments.

Figure 4-31: PIDR2 register bit assignments



The following table shows the register bit assignments.

Table 4-27: PIDR2

Bits	Name	Function
[31:8]	-	Reserved, returns 0.
[7:4]	REVISION	Revision identifier for the GFC-200: • `0x0` = r0p0.
[3]	JEDEC	Returns 1, which indicates the use of a JEDEC-assigned ID value.
[2:0]	DES_1	The JEDEC JEP106 ID code [6:4], which identifies Arm as the designer of the GFC-200. See also PIDR1.DES_0[3:0] and the *Standard Manufacturer's Identification Code*.

4.4.27 Peripheral ID register 3, PIDR3

The PIDR3 register returns byte[3] of the peripheral identifier. A debugger during system discovery can use the peripheral ID to discover which peripherals are in the system and whether the peripheral has any modifications applied.

The PIDR3 register characteristics are:

Usage constraints

There are no usage constraints.

Configurations

Available in all configurations.

Attributes

Offset

`0xFEC`

Type

Read only.

Reset

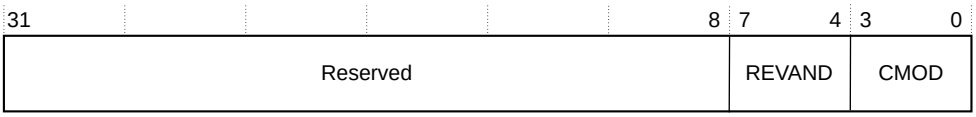
`0x00`

Width

32

The following figure shows the bit assignments.

Figure 4-32: PIDR3 register bit assignments



The following table shows the register bit assignments.

Table 4-28: PIDR3

Bits	Name	Function
[31:8]	-	Reserved, returns 0.
[7:4]	REVAND	A nonzero value indicates that Arm has approved the application of a post-manufacture metal layer fix to the GFC-200 silicon.
[3:0]	CMOD	Customer modification number. A nonzero value indicates that the customer has modified the GFC-200 RTL, which might affect its behavior. Do not modify this field unless you have permission from Arm.

4.4.28 Component ID register 0, CIDR0

The CIDR0 register returns byte[0] of the component ID. A debugger during system discovery can use the component ID to discover that the peripheral contains a programmers register block.

The CIDR0 register characteristics are:

Usage constraints

There are no usage constraints.

Configurations

Available in all configurations.

Attributes

Offset

`0xFF0`

Type

Read only.

Reset

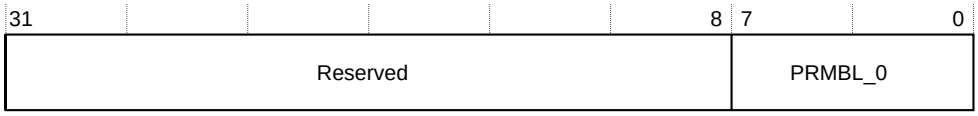
`0x0D`

Width

32

The following figure shows the bit assignments.

Figure 4-33: CIDR0 register bit assignments



The following table shows the register bit assignments.

Table 4-29: CIDR0

Bits	Name	Function
[31:8]	-	Reserved, returns 0.
[7:0]	PRMBL_0	Preamble[0]. Returns segment 1 of the component identification code.

4.4.29 Component ID register 1, CIDR1

The CIDR1 register returns byte[1] of the component ID. A debugger during system discovery can use the component ID to discover that the peripheral contains a programmers register block and which component class the GFC-200 belongs to.

The CIDR1 register characteristics are:

Usage constraints

There are no usage constraints.

Configurations

Available in all configurations.

Attributes

Offset

`0xFF4`

Type

Read only.

Reset

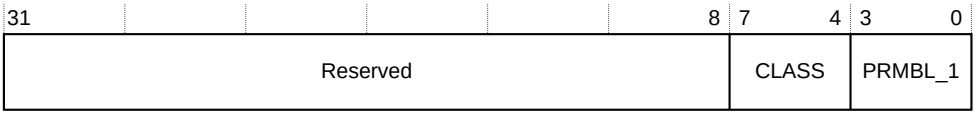
`0xF0`

Width

32

The following figure shows the bit assignments.

Figure 4-34: CIDR1 register bit assignments



The following table shows the register bit assignments.

Table 4-30: CIDR1

Bits	Name	Function
[31:8]	-	Reserved, returns 0.
[7:4]	CLASS	Component class. Returns `0xF`, which indicates that the GFC-200 belongs to the CoreLink™ family.
[3:0]	PRMBL_1	Preamble[1]. Returns segment 2 of the component identification code.

4.4.30 Component ID register 2, CIDR2

The CIDR2 register returns byte[2] of the component ID. A debugger during system discovery can use the component ID to discover that the peripheral contains a programmers register block.

The CIDR2 register characteristics are:

Usage constraints

There are no usage constraints.

Configurations

Available in all configurations.

Attributes

Offset

0xFF8

Type

Read only.

Reset

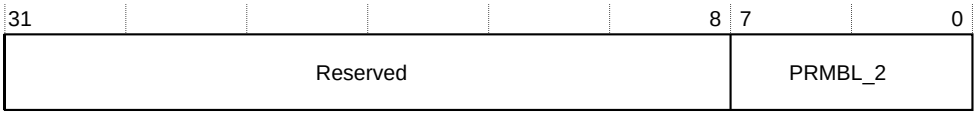
0x05

Width

32

The following figure shows the bit assignments.

Figure 4-35: CIDR2 register bit assignments



The following table shows the register bit assignments.

Table 4-31: CIDR2

Bits	Name	Function
[31:8]	-	Reserved, returns 0.
[7:0]	PRMBL_2	Preamble[2]. Returns segment 2 of the component identification code.

4.4.31 Component ID register 3, CIDR3

The CIDR3 register returns byte[3] of the component ID. A debugger during system discovery can use the component ID to discover that the peripheral contains a programmers register block.

The CIDR3 register characteristics are:

Usage constraints

There are no usage constraints.

Configurations

Available in all configurations.

Attributes

Offset

0xFFC

Type

Read only.

Reset

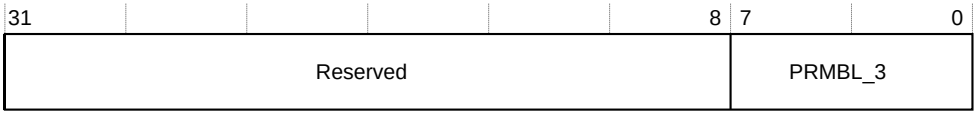
0xB1

Width

32

The following figure shows the bit assignments.

Figure 4-36: CIDR3 register bit assignments



The following table shows the register bit assignments.

Table 4-32: CIDR3

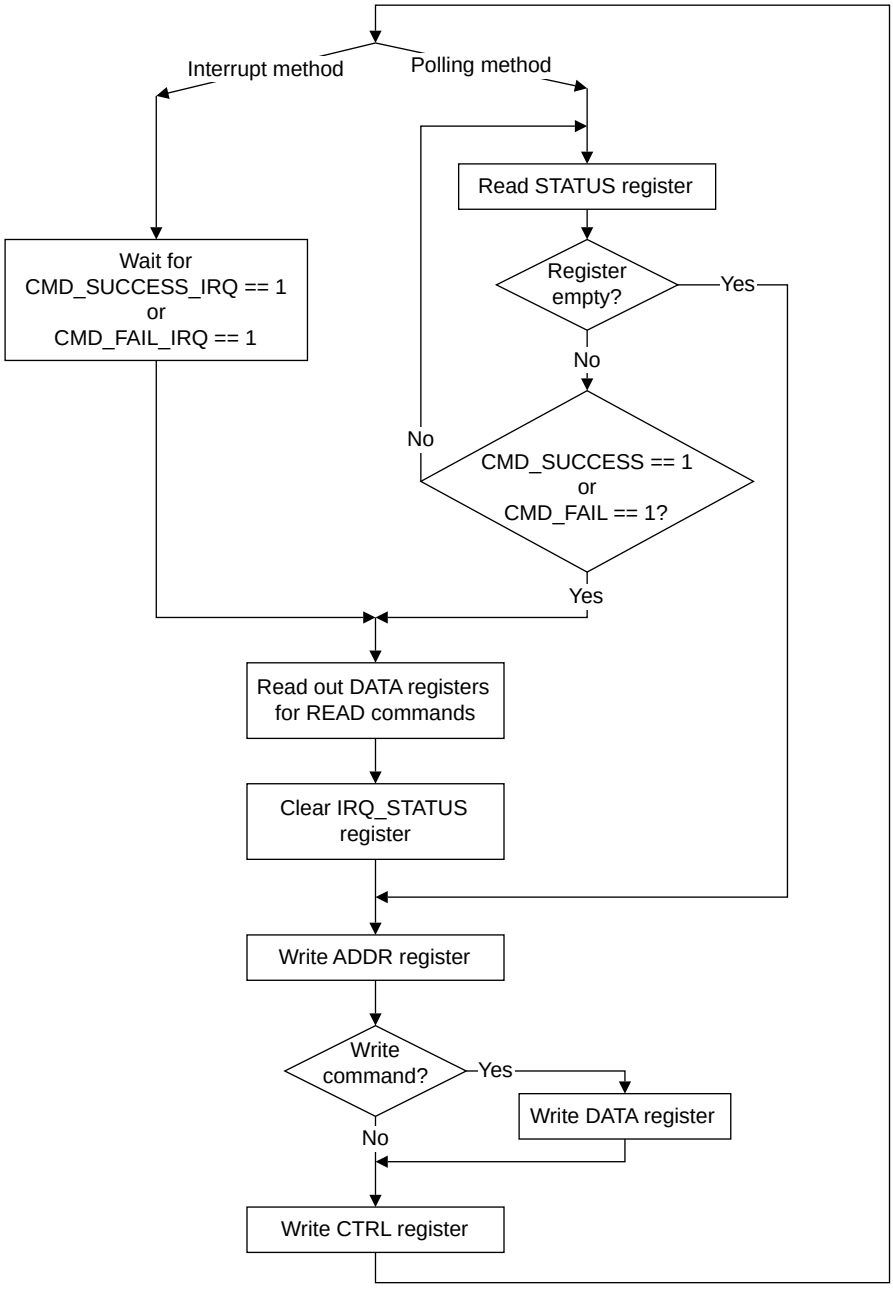
Bits	Name	Function
[31:8]	-	Reserved, returns 0.
[7:0]	PRMBL_3	Preamble[3]. Returns segment 3 of the component identification code.

4.5 Accessing Flash from the APB completer interfaces

Software must follow a particular flow to access the embedded Flash from the APB completer interfaces.

The following figure shows how to access the embedded Flash from an APB completer interface.

Figure 4-37: APB access to Flash flowchart



The APB completer interface can use either of two modes to trigger a new command:

Polling method

Software polls the STATUS register for any active bits, where an active bit indicates that GFC-200 is processing a command.

Interrupt method

Software enables the CMD_SUCCESS_IRQ and CMD_FAIL_IRQ interrupts, and waits for either interrupt to be triggered. An interrupt occurs when a command enters the SUCCESS or FAIL state.

To initiate a new transfer, software must write to the ADDR, DATA0, and CTRL registers. The CTRL register must be written last because it triggers access to the embedded Flash. The command then enters the queue until it is executed.

The CTRL, ADDR, and DATA0 register cannot be written in the following cases:

- CTRL register is not equal to 0.
- IRQ_STATUS register, bits[4:0], contain a pending interrupt bit.

IRQ_STATUS is an internal register that is not present in the programmers model. However, you can access the value of IRQ_STATUS by reading the IRQ_STATUS _SET or IRQ_STATUS_CLR register.

If these conditions are not met, the GFC-200 ignores the write to the CTRL, ADDR, or DATA0 registers and sets CMD_REJECT_IRQ = 1, to indicate a programming fault. This behavior ensures that software does not change the value of a pending command on the GFB interface, and forces software to process the result of the previously executed transaction.

Related information

Interrupt masked status register, IRQ_MASKED_STATUS on page 46
Status register, STATUS on page 49

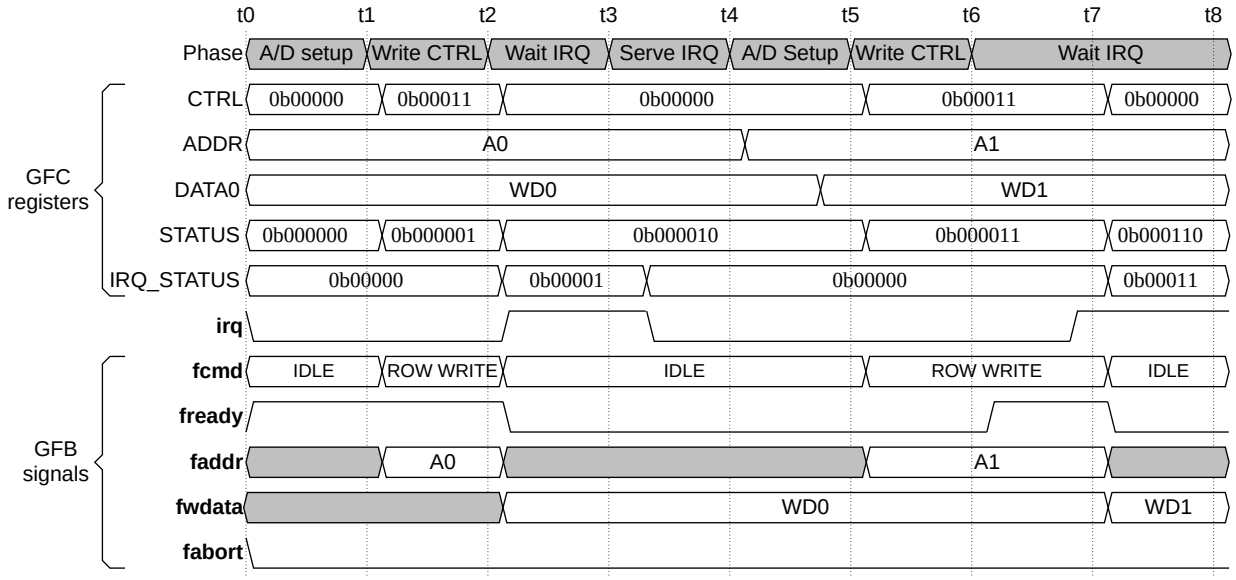
4.6 Preloading transfers

To improve the efficiency of ROW WRITE commands that use the command queue, transfers can be preloaded to the CTRL register by enabling the CMD_ACCEPT_IRQ interrupt.

When the APB requester wants to achieve the efficiency benefits of a ROW WRITE command, it must send the command for Flash transfers back-to-back, and keep the Flash in the high-voltage programming state.

The following figure shows the ROW WRITE timing diagram.

Figure 4-38: ROW WRITE timing diagram



ROW WRITE command sequence

To support the preloading mechanism, software must implement the following phases in the ROW WRITE command sequence:

A/D setup

Software writes to the ADDR and DATA0 registers for the upcoming command.

Write CTRL

Software writes to the CTRL register and initiates the transfer. In this example, the GFB interface is ready to immediately accept the command at t2, and the command then passes to the GFB.

After t2, the GFC-200 clears the CTRL register automatically, and asserts the CMD_ACCEPT_IRQ interrupt.

Wait IRQ

Software waits until it receives the interrupt.

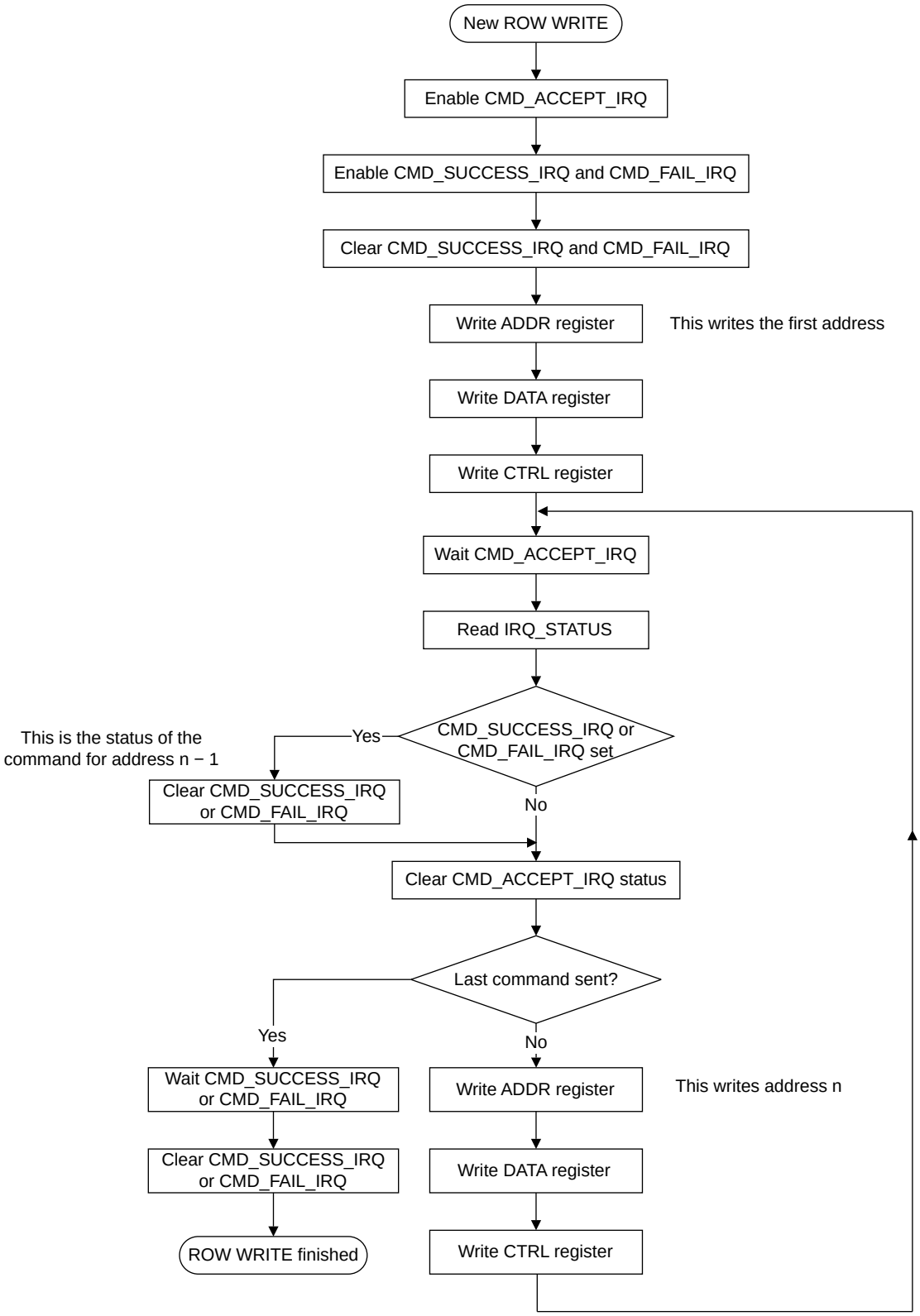
Serve IRQ

Software acknowledges the interrupt by writing to the IRQ_STATUS_CLR register. When the IRQ_STATUS register clears, the address and data for the next transfer can be set up before writing the next ROW WRITE command to the CTRL register.

To ensure that the preload mechanism works correctly, the *Write CTRL* phase in t5 must finish before t7, when the GFB interface is ready with the next currently executed command, and is able to accept new commands. Otherwise, the GFC-200 inserts an IDLE command, and software does not get the benefits of the ROW WRITE command. Arm expects the write commands to the embedded Flash to require thousands of clock cycles, so software can use this period to service the CMD_ACCEPT_IRQ interrupt and update the registers.

The following figure shows the ROW WRITE preloading flowchart.

Figure 4-39: ROW WRITE preloading flowchart



Using CMD_ACCEPT_IRQ with READ commands

The CMD_ACCEPT_IRQ is intended for ROW WRITE commands. However, all other commands can use this interrupt because the hardware implementation cannot restrict its use.

If software uses CMD_ACCEPT_IRQ for READ commands, the read data of the current READ command might be lost if the resulting interrupts are not processed in time. If software fails to service the CMD_ACCEPT_IRQ before the GFC-200 overwrites the DATA registers, then it sets the READ_OVERFLOW_IRQ bit to indicate this error.

Using an APB completer interface to read the Flash is intended for debug purposes only. Therefore, for APB reads, you can use CMD_SUCCESS_IRQ and CMD_FAIL_IRQ interrupts. If you require a higher read throughput, then Arm recommends using an AHB interface.

4.7 Reconfiguring the partition access rights

The GFC-200 supports on-the-fly reconfiguration of the partition access rights. You can use this feature during a firmware update, when software allocates additional Flash memory and then programs the new firmware.

You can also use this feature for ROM emulation, when software locks down a memory region to be read-only accessible.

During partition reconfiguration, software must ensure that any sensitive data is not compromised.

Example 4-4: Example firmware update process

During a firmware update, software must perform the following steps:

1. The primary and secondary domains negotiate with each other to enter a secure maintenance update mode that enables the primary manager to perform the reconfiguration.

   For a firmware update, the primary manager might take ownership of the entire Flash or at least any partitions that it must allocate to itself, while the secondary manager goes to an idle state such as reset or WFI.
2. The primary manager writes to the PART_CONFIG_MODE_REQ register, to request that the GFC-200 moves to configuration mode.
3. After the GFC-200 completes any pending GFB command, then it enters configuration mode and it sets the PART_CONFIG_MODE_IRQ interrupt to notify the primary manager.

   When the GFC-200 is in configuration mode, it ignores any requests from the secondary domain. If the secondary domain issues an AHB read, then the GFC-200 responds with a 2-cycle ERROR response and it sets CMD_FAIL_IRQ = 1. If the secondary domain issues an APB command, then the GFC-200 sets CMD_FAIL_IRQ = 1.

4. The primary manager changes the values of the partition control inputs to either:
   - Re-allocate the ownership of only the necessary partitions that the size of the new firmware requires.
   - Assigns all partitions to the primary domain.
     The GFC-200 loads the new values of the partition control inputs when the GFB is idle. See 3.2.9 Partition configuration interface on page 25 for more information.
5. The primary manager performs the firmware update.
6. The primary manager changes the values of the partition control inputs so that it re-assigns the partition ownership for both domains. The primary manager must make sure that it does not assign access rights to the secondary manager to partitions that contain sensitive data.
7. The primary manager writes zero to the PART_CONFIG_MODE_REQ register, and the GFC-200 exits configuration mode after it completes any pending GFB command.
8. Software instructs the primary and secondary managers to exit from the secure maintenance update mode.

Related information

Partition configuration mode request register, PART_CONFIG_MODE_REQ on page 57

4.8 Flash macro power control

Each domain programs the POWER_STATE_REQ register with their minimum power state that they require for the Flash macro. The GFC-200 dynamically controls the Flash macro power, and the minimum power state is the higher minimum requirement from both POWER_STATE_REQ registers.

If a domain does not have any requirements to access Flash, then Arm recommends that its POWER_STATE_REQ register is set to all zeros so that the other domain can control the Flash power requirements.

When there is a change in power state, the GFC-200 sets PWR_STATE_CHANGE_IRQ = 1 for both domains, which can generate an interrupt towards each domain.

To discover the Flash power state, software can read the POWER_STATE register.

Related information

Power state request register, POWER_STATE_REQ on page 63

Appendix A Signal Descriptions

Read this for a description of the input and output signals.

A.1 System interface signals

The system interface contains clock, reset, and interrupt signals. The interface also provides some Flash power control and isolation signals.

The following table shows the system interface signals.

Table A-1: System interface signals

Signal name	Direction	Description
clk	Input	Clock for all GFC-200 interfaces.
resetn	Input	Reset for GFC-200. Active-LOW. You can set this signal LOW asynchronously, but set it HIGH synchronously with **clk**.
irq0	Output	Interrupt request to the primary domain. Active-HIGH.
irq1	Output	Interrupt request to the secondary domain. Active-HIGH.
flash_pwr_rdy	Output	Indicates that power to the Flash macro is stable. Active-HIGH. The GFB receiver logic must monitor this signal and only initiate transactions towards the Flash macro when **flash_pwr_rdy** is HIGH.
flash_pwr_opmode	Output	Indicates the Flash power operating mode to the GFB receiver logic: 0 = The Flash is in OPMODE_0 operational mode (default). 1 - The Flash is in OPMODE_1 operational mode.
flash_devisolaten	Output	Active-LOW isolation control output. This signal controls isolation cells that isolate the Flash macro in the OFF and FULL_RET states.
flash_macro_resetn	Output	Active-LOW reset for Flash macros that require a powerup reset. This reset is not intended for the process-specific part.

Related information

System interface on page 27

A.2 AHB-Lite subordinate interface signals

The AHB-Lite subordinate interface is a read-only port for accessing the Flash contents. The interface includes the **hpart** sideband signal, that indicates whether the transfer originates from the primary or secondary domain. The transfers that the GFC-200 receives are forwarded to the GFB interface.

The following table shows the AHB-Lite subordinate interface signals.

Table A-2: AHB-Lite subordinate interface signals

Signal name	Direction	Description
hsel	Input	Subordinate select signal.
haddr[21:0]	Input	Address bus.
htrans[1:0]	Input	Indicates the current transfer type.
hwrite	Input	Only reads are allowed. Can be tied LOW externally. Writes generate an error.
hsize[2:0]	Input	Indicates the size of a data transfer. The transfer size must not exceed the smaller value of either *HRDATA_WIDTH* or *FRDATA_WIDTH*. Larger transfer sizes are not allowed and responded with error.
hburst[2:0]	Input	Indicates the transfer burst type.
hmastlock	Input	Indicates whether the transfer on the bus is atomic. Used for blocking arbitration over the GFB.
hready	Input	When HIGH, indicates that a bus transfer has completed.
hpart	Input	Domain identifier. **hpart** indicates which domain the transfer originates from: `0b0` = Primary domain. `0b1` = Secondary domain. **hpart** must be driven during the address phase of each AHB transfer, and also remain stable during waited transfers.
hreadyout	Output	When HIGH, indicates that a bus transfer has completed. Reset value = `0b1`.
hresp	Output	Transfer response status. Reset value = `0b0`.
hrdata[HRDATA_WIDTH−1:0]	Output	Read data. Reset value = `0x0`.


Note

AHB-Lite signals that are not shown in the table are not used in GFC-200. See the *AMBA® 3 AHB-Lite Protocol Specification v1.0* for more information.

Related information

AHB-Lite subordinate interface on page 17

A.3 APB completer interface signals

The GFC-200 provides two APB completer interfaces. One interface for the primary domain and one interface for the secondary domain. The primary domain can access some extra registers that enable the primary APB requester to control the Flash partitioning behavior. The primary domain can also access the registers in the process-specific part.

Primary APB completer interface signals, APB 0

The following table shows the APB completer interface signals that the primary APB requester connects to.

Table A-3: Primary APB completer interface (APB 0) signals

Signal name	Direction	Description
psel_s0	Input	Completer select signal.
penable_s0	Input	Indicates the start of the second cycle of an APB transfer.
paddr_s0[12:0]	Input	Address bus. **paddr_s0[12]** selects either the internal or an external register bank: 0 = Internal registers. 1 = External registers in the process-specific part.
pstrb_s0[3:0]	Input	Write strobe port. Each bit refers to a byte in the **pwdata_s0** signal.
pwrite_s0	Input	APB transfer direction.
pwdata_s0[31:0]	Input	32-bit write data bus.
pwakeup_s0	Input	The APB bridge sets this signal HIGH when a transfer is in progress on the primary APB interface, APB 0.
prdata_s0[31:0]	Output	32-bit read data bus. Reset value = `0x0`.
pready_s0	Output	Driven LOW when extra wait states are required to complete access to the external registers. Reset value = `0b0`.
pslverr_s0	Output	Driven HIGH when an error response is received from an access to the external registers. Reset value = `0b0`.

Secondary APB completer interface signals, APB 1

The following table shows the APB completer interface signals that the secondary APB requester connects to.

Table A-4: Secondary APB completer interface (APB 1) signals

Signal name	Direction	Description
psel_s1	Input	Completer select signal.
penable_s1	Input	Indicates the start of the second cycle of an APB transfer.
paddr_s1[11:0]	Input	Address bus.
pstrb_s1[3:0]	Input	Write strobe port. Each bit refers to a byte in the **pwdata_s1** signal.
pwrite_s1	Input	APB transfer direction.

Signal name	Direction	Description
pwdata_s1[31:0]	Input	32-bit write data bus.
pwakeup_s1	Input	The APB bridge sets this signal HIGH when a transfer is in progress on the secondary APB interface, APB 1.
prdata_s1[31:0]	Output	32-bit read data bus. Reset value = `0x0`.
pready_s1	Output	Reset value = `0b0`.
pslverr_s1	Output	Reset value = `0b0`.


Note

APB signals that are not shown in the tables are not used in GFC-200. See the *AMBA® APB Protocol Specification Version 2.0* for more information.

Related information

Primary APB completer interface on page 20
Secondary APB completer interface on page 21

A.4 APB requester interface signals

The APB requester interface connects to the process-specific part, and enables the GFC-200 to access the registers in the process-specific part.

The following table shows the signals that are used by the APB requester interface.

Table A-5: APB requester interface signals

Signal name	Direction	Description
psel_m	Output	Process-specific part select signal.
penable_m	Output	Indicates the start of the second cycle of an APB transfer.
paddr_m[11:0]	Output	Address bus.
pstrb_m[3:0]	Output	Write strobe port. Each bit refers to a byte in the **pwdata_m** signal.
pwrite_m	Output	APB transfer direction.
pwdata_m[31:0]	Output	32-bit write data bus.
prdata_m[31:0]	Input	32-bit read data bus.
pready_m	Input	Driven LOW when extra wait states are required to complete access to the external registers.
pslverr_m	Input	Driven HIGH when an error response is received from an access to the external registers.


Note

APB signals that are not shown in the table are not used in GFC-200. See the *AMBA® APB Protocol Specification Version 2.0* for more information.

Related information

APB requester interface on page 21

A.5 GFB manager interface signals

The *Generic Flash Bus* (GFB) manager interface connects to the process-specific part, and provides the memory interface to the eFlash.

The following table shows the GFB signals.

Table A-6: GFB manager interface signals

Signal name	Direction	Description
faddr[21:0]	Output	Address bus width is fixed at 22 bits, which allows access to a 4MB embedded Flash. The **faddr[1:0]** bits are not used because the minimum data width is 32 bits.
fcmd[2:0]	Output	Command bus: `0b000` = IDLE. `0b001` = READ. `0b010` = WRITE. `0b011` = ROW WRITE. `0b100` = ERASE. `0b101` = Reserved. `0b110` = Reserved. `0b111` = MASS ERASE.
fabort	Output	Abort indication. When HIGH, the manager requests to abort the command that is running.
fwdata[FWDATA_WIDTH−1:0]	Output	Write data bus.
frdata[FRDATA_WIDTH−1:0]	Input	Read data bus.
fready	Input	Command ready indication. Driven LOW if the process-specific part requires wait states to complete the access. Driven HIGH when the process-specific part is ready with the previous access and is able to accept a new command.
fresp	Input	Flash error indication for the previously accepted command. Driven HIGH for two cycles when an error is indicated for the command that is running.

Note

- GFB signals that are not shown in the table are not used in GFC-200. See the *AMBA® Generic Flash Bus Protocol Specification* for more information.
- The GFC-200 and the GFB receiver logic inside the process-specific part are expected to be in the same power domain.

Related information

GFB interface on page 22

A.6 Partition control interface signals

The GFC-200 supports 16 Flash partitions. The partition control interface signals controls whether the primary domain or the secondary domain is the owner of a Flash partition. It also controls whether the domain has read/write or read-only access to a partition.

The following table shows the partition control interface signals. These signals connect to a Secure enclave or an eFuse.

Table A-7: Partition control interface signals

Signal name	Direction	Description
partition_ctrl_rw[15:0]	Input	Each bit sets the owner of the corresponding partition in the Flash: **Bit[n] = 0** The primary domain is the owner of partition `n`. **Bit[n] = 1** The secondary domain is the owner of partition `n`. The owner of a partition has read and write access to it, unless **partition_ctrl_ro** sets the partition as read-only. When 16×*`PARTITION_SIZE`* is smaller than the address space of the Flash, the last control input sets the state of the remaining part of the Flash. When 16×*`PARTITION_SIZE`* is larger than the address space of the Flash, the remaining (MSB) control bits have no effect on the utilized parts of the memory space, but they must be considered for MASS ERASE.
partition_ctrl_rd[15:0]	Input	Each bit controls whether the domain owner grants read access to the other domain: **Bit[n] = 0** The primary domain has read access to partition `n`. **Bit[n] = 1** The secondary domain has read access to partition `n`.
partition_ctrl_ro[15:0]	Input	Each bit controls whether the domain owner has read/write or read-only access to the corresponding partition: **Bit[n] = 0** The owner of partition `n` has read/write access to that partition. **Bit[n] = 1** The owner of partition `n` has read-only access to that partition.

Signal name	Direction	Description
config_mode_me_en	Input	MASS ERASE enable in configuration mode. When HIGH, the primary domain can initiate a MASS ERASE operation, when the GFC-200 is in configuration mode. The mass erase occurs irrespective of the **partition_ctrl_rw** and **partition_ctrl_ro** settings.

Related information

Partition configuration interface on page 25

A.7 Q-Channel interface signals

The Q-Channel device interfaces provide clock and power control signals for the GFC-200. The **clk_*** signals connect to a clock controller and the **pwr_*** signals connect to a power controller.

The following table lists the signals for both Q-Channel device interfaces. See the *AMBA® Low Power Interface Specification, Arm® Q-Channel and P-Channel Interfaces* for more information about Q-Channels.

Table A-8: Clock control Q-Channel device interface signals

Signal	Direction	Description
clk_qreqn	Input	This signal indicates when the clock controller issues a quiescence entry or exit request to the GFC-200. The input contains a 2-stage synchronizer, so the signal can transition asynchronously.
clk_qacceptn	Output	This signal indicates when the GFC-200 accepts the quiescence request.
clk_qdeny		This signal indicates when the GFC-200 denies the quiescence request.
clk_qactive		This signal indicates when the GFC-200 is active or it is requesting to exit from quiescence.

Table A-9: Power control Q-Channel device interface signals

Signal	Direction	Description
pwr_qreqn	Input	This signal indicates when the power controller issues a quiescence entry or exit request to the GFC-200. The input contains a 2-stage synchronizer, so the signal can transition asynchronously.
pwr_qacceptn	Output	This signal indicates when the GFC-200 accepts the quiescence request.
pwr_qdeny		This signal indicates when the GFC-200 denies the quiescence request.
pwr_qactive		This signal indicates when the GFC-200 is active or it is requesting to exit from quiescence.

Related information

Q-Channel interface for clock on page 23
Q-Channel interface for power on page 24

A.8 P-Channel controller interface signals

The P-Channel controller interface sends power state information to the process-specific part. The process-specific part contains a *Power Control State Machine* (PCSM) that controls the power to the process-specific part.

The following table shows the P-Channel controller interface signals. See the *AMBA® Low Power Interface Specification, Arm® Q-Channel and P-Channel Interfaces* for more information about P-Channels.

Table A-10: P-Channel controller interface signals

Signal name	Direction	Description
preq	Output	Power state change request. Active-HIGH.
pstate[4:0]	Output	Requested power state value. See PSTATE to PPU and Flash power modes on page 31 for more information.
paccept	Input	Accept indication. Active-HIGH. Synchronized with double-flop synchronizer.

The GFC-200 does not provide the **pdeny** and **pactive** input signals. Therefore, the process-specific part cannot deny a power state transition nor can it request a certain power state.

Related information

P-Channel controller interface on page 25

A.9 DFT signals

The GFC-200 provides *Design for Test* (DFT) signals. A DFT controller controls these signals during *Automatic Test Pattern Generation* (ATPG) testing.

The following table shows the DFT signals.

Table A-11: DFT signals

Signal name	Direction	Description
DFTSCANMODE	Input	When HIGH, this signal selects scan mode testing.
DFTCGEN	Input	When HIGH, this signal enables all the clock gates in the GFC-200.
DFTISODISABLE	Input	When HIGH, this signal disables isolation gates.
DFTRSTDISABLE	Input	When HIGH, this signal prevents asynchronous reset signals from resetting the GFC-200 during shift, but allows the logic to be tested completely and at-speed during capture.

Appendix B Revisions

The following tables describe the technical changes between released issues of this book.

Table B-1: Issue 0000-00

Change	Location	Affects
First release.	-	-

Table B-2: Differences between issue 0000-00 and issue 0000-01

Change	Location	Affects
Removed offensive terms. Added inclusive language statement.	Throughout document. Inclusive language commitment on page 4	All versions