AMBA™ 3 TrustZone Interrupt Controller (SP890)

Revision: r0p0

Technical Overview

ARM®

ARM DTO 0013B

AMBA 3 TrustZone Interrupt Controller (SP890)

Technical Overview

Release Information

The following changes have been made to this book.

Change history

Date	Issue	Confidentiality	Change
31 January 2005	A	Confidential	First release for r0p0
30 September 2008	B	Non-Confidential Unrestricted Access	Second release for r0p0

Proprietary Notice

Words and logos marked with ® or ™ are registered trademarks or trademarks of ARM Limited in the EU and other countries, except as otherwise stated below in this proprietary notice. Other brands and names mentioned herein may be the trademarks of their respective owners.

Neither the whole nor any part of the information contained in, or the product described in, this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder.

The product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given by ARM in good faith. However, all warranties implied or expressed, including but not limited to implied warranties of merchantability, or fitness for purpose, are excluded.

This document is intended only to assist the reader in the use of the product. ARM Limited shall not be liable for any loss or damage arising from the use of any information in this document, or any error or omission in such information, or any incorrect use of the product.

Where the term ARM is used it means “ARM or any of its subsidiaries as appropriate”.

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.

Web Address

`http://www.arm.com`

Contents
AMBA 3 TrustZone Interrupt Controller (SP890) Technical Overview

Chapter 1 **Technical Overview**

1.1 About the TrustZone Interrupt Controller 1-2
1.2 Functional description 1-3
1.3 Programmer's model 1-6
1.4 Programmer's model for test 1-18
1.5 Physical data 1-21
1.6 Signal descriptions 1-22
1.7 Revisions 1-24

Chapter 1
Technical Overview

This technical overview describes the functionality of the *TrustZone Interrupt Controller* (TZIC) in the following sections:

- *About the TrustZone Interrupt Controller* on page 1-2
- *Functional description* on page 1-3
- *Programmer's model* on page 1-6
- *Programmer's model for test* on page 1-18
- *Physical data* on page 1-21
- *Signal descriptions* on page 1-22
- *Revisions* on page 1-24.

1.1 About the TrustZone Interrupt Controller

The *TrustZone Interrupt Controller* (TZIC) is an *Advanced Microcontroller Bus Architecture* (AMBA) compliant, *System-on-Chip* (SoC) peripheral that is developed, tested and licensed by ARM Limited.

The TZIC provides a software interface to the secure interrupt system in a TrustZone design. It provides secure control of the **nFIQ** interrupt and masks the interrupt source(s) from the interrupt controller on the non-secure side of the system. You can then use the latter to generate the **nIRQ** signal. This can be a simple or vectored design. Figure 1-1 shows the TZIC a typical configuration.

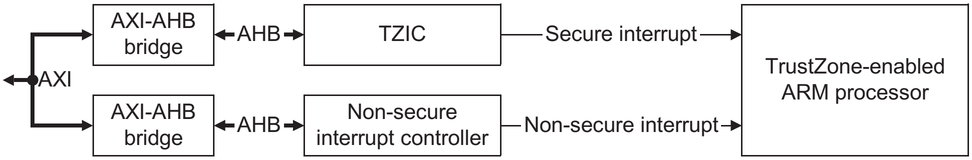

Figure 1-1 TZIC overview

The TZIC has the following features:

- **nFIQ** generation, under secure control, from any system interrupt source
- masking of chosen **nFIQ** interrupt from non-secure interrupt controller
- software access to raw and **nFIQ** interrupt status
- secure **nSFIQ** input for daisy-chaining of TZICs
- non-secure **nNSFIQ** input, with bypass, to allow access to **nFIQ** for operating systems that do not support TrustZone
- AHB system interface.

Note

- You must use this device with an AXI-AHB bridge as shown in Figure 1-1. AXI supports TrustZone and this bridge reads the secure bit and provides the correct response. The AHB interface in the TZIC does not have to handle the secure bit. The 4KB block in which the TZIC registers reside must be in a secure area of memory.
- An error response is generated on **HRESP** if a SEQ or NSEQ transaction is made and the transaction size, **HSIZE**, is not 32-bit.

1.2 Functional description

The TZIC provides a software interface to control the generation of the **nFIQ** interrupt. In addition, it masks out the interrupt source(s) chosen for **nFIQ** from the interrupts that are passed onto a non-secure interrupt controller.

The following sections provide an overview of the TZIC in:

- *Two-level TrustZone interrupt configuration*
- *Two-level daisy-chained TrustZone interrupt configuration* on page 1-4.

Caution

You must use a secure software protocol before relying on any security settings that can be changed. This might include, but is not limited to:

- verifying that instructions to change the security settings are propagated across the interconnect to their final destination
- clearing any storage locations that can change the security status;
- flushing caches and page tables
- stopping other masters.

1.2.1 Two-level TrustZone interrupt configuration

The interrupt structure in a TrustZone system consists of two levels of interrupt controller described in:

- *Secure interrupt controller*
- *Non-secure interrupt controller* on page 1-4.

Within each of these levels, more than one interrupt controller can be daisy-chained. This is necessary if more than 32 interrupts must be supported.

Secure interrupt controller

The task of the TZIC is to create the secure interrupt, **nFIQ**, from the raw interrupt source(s). It must also stop the source(s) that you chose for the **nFIQ** from propagating to the non-secure interrupt controller. This creates a secure interrupt that the non-secure side of a TrustZone system cannot control or influence. To the non-secure side, this is in effect a non-maskable interrupt. To ensure security, the TZIC must reside in a secure area of memory. By doing this, only a secure access can control the **nFIQ**.

Non-secure interrupt controller

The non-secure interrupt controller creates the non-secure interrupt, **nIRQ**, from the interrupt sources that are enabled by the TZIC. This can be either a simple or *Vectored Interrupt Controller* (VIC), for example the PrimeCell VIC (PL192). This controller can reside in the non-secure side of a TrustZone system and provides the user mode interrupt control. Because of the core architecture, the **nIRQ** does not provide a path into the secure side of a TrustZone system. You do not use the **nFIQ** output of this controller in a TrustZone system, but in some circumstances, you can route it back into the TZIC to enhance system flexibility.

Figure 1-2 shows a block diagram of the major component blocks of the TZIC in a two-level configuration.

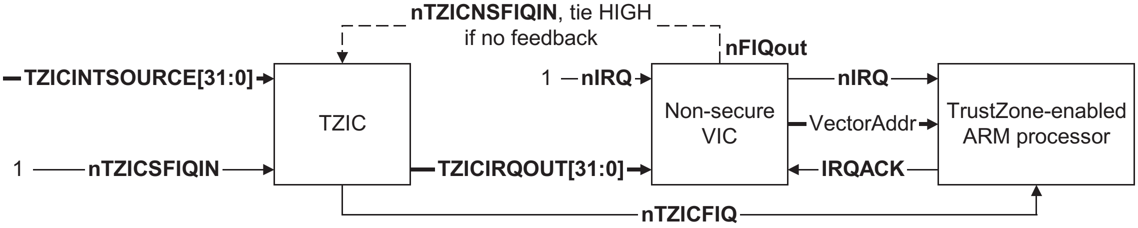

Figure 1-2 TZIC components in a two-level configuration

1.2.2 Two-level daisy-chained TrustZone interrupt configuration

Figure 1-3 on page 1-5 shows a block diagram of the major component blocks when the TZIC is daisy-chained in a two-level configuration.

Note

Figure 1-2 and Figure 1-3 on page 1-5 show the **nTZICNSFIQIN** input to the TZIC from the VIC. This signal connects to the VIC **nFIQ** output.

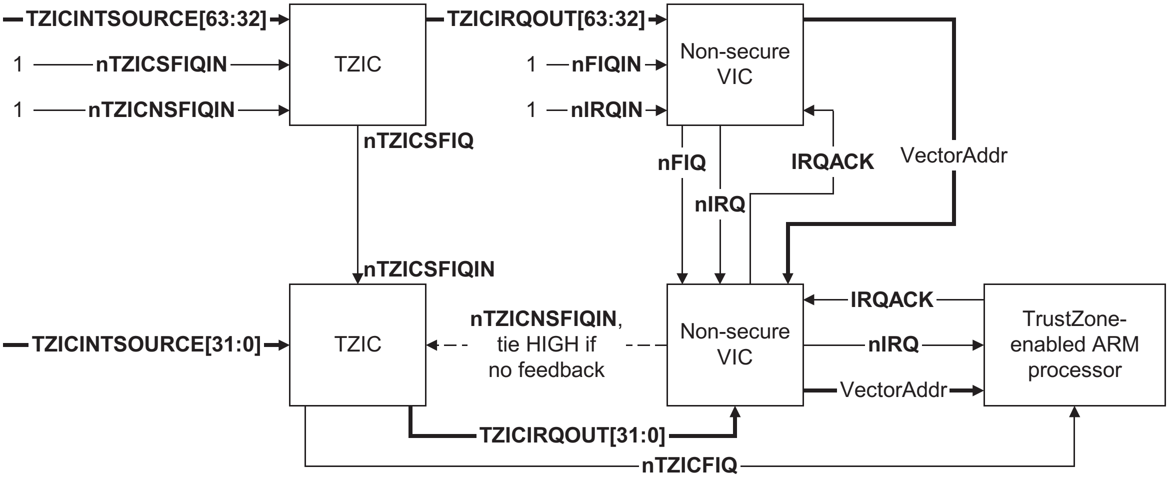

Figure 1-3 TZIC components in a daisy-chained configuration

1.3 Programmer's model

The following apply to the registers used in the TZIC:

- To ensure that interrupts in a TrustZone system are handled securely, you must place the TZIC in a secure area of memory.
- The base address of the node is not fixed and can be different for any particular system implementation. However, the offset of any particular register from the base address is fixed.
- You must not access reserved or unused address locations because this can result in unpredictable behavior of the device.
- You must write reserved or unused bits of registers as zero, and ignored them on read unless otherwise stated.
- A system or power-on reset resets all register bits to a logic 0 unless otherwise stated.
- All registers support read and write access unless otherwise stated.
- A write updates the contents of a register and a read returns the contents of the register.
- You must access all registers using zero wait states unless otherwise stated.

The following sections describe the TZIC registers:

- *TZIC register summary*
- *Register descriptions* on page 1-7.

1.3.1 TZIC register summary

Table 1-1 summarizes the registers in base offset order.

Table 1-1 TZIC register summary

Name	Offset	Type	Reset value	Description
TZICFIQStatus	`0x000`	RO	`0x00000000`	See *FIQ status register* on page 1-8
TZICRawIntr	`0x004`	RO	-	See *Raw interrupt status register* on page 1-8
TZICIntSelect	`0x008`	R/W	`0x00000000`	See *Interrupt select register* on page 1-8
TZICFIQEnable	`0x00C`	R/W	`0x00000000`	See *FIQ enable register* on page 1-9
TZICFIQENClear	`0x010`	WO	-	See *FIQ enable clear register* on page 1-9

Table 1-1 TZIC register summary (continued)

Name	Offset	Type	Reset value	Description
TZICFIQBypass	`0x014`	R/W	`0x00000000`	See *FIQ bypass register* on page 1-9
TZICProtection	`0x018`	R/W	`0x00000000`	See *Protection register* on page 1-10
TZICLock	`0x01C`	WO	-	See *Lock enable register* on page 1-11
TZICLockStatus	`0x020`	RO	`0x00000001`	See *Lock status register* on page 1-11
TZICITCR	`0x300`	R/W	`0x00000000`	See *Test control register* on page 1-18
TZICITIP1	`0x304`	R/W	`0x00000440`	See *Test input 1 register* on page 1-18
TZICITIP2	`0x308`	R/W	`0x00000000`	See *Test input 2 Register* on page 1-19
TZICITOP1	`0x30C`	R/W	`0x00000040`	See *Test output 1 register* on page 1-20
TZICITOP2	`0x310`	R/W	`0x00000000`	See *Test output 2 register* on page 1-20
TZICPeriphID0	`0xFE0`	RO	`0x00000090`	See *Peripheral identification registers* on page 1-12
TZICPeriphID1	`0xFE4`	RO	`0x00000018`	
TZICPeriphID2	`0xFE8`	RO	`0x00000004`	
TZICPeriphID3	`0xFEC`	RO	`0x00000000`	
TZICPCellID0	`0xFF0`	RO	`0x0000000D`	See *Identification registers* on page 1-15
TZICPCellID1	`0xFF4`	RO	`0x000000F0`	
TZICPCellID2	`0xFF8`	RO	`0x00000005`	
TZICPCellID3	`0xFFC`	RO	`0x000000B1`	

The following sections describe the registers:

1.3.2 Register descriptions

This section provides the register description, A cross-reference to each register description is provided in Table 1-1 on page 1-6.

FIQ status register

The read-only TZICFIQStatus Register has a reset value of `0x00000000`. It provides the status of the interrupts after FIQ masking. Table 1-2 lists the register bit assignments.

Table 1-2 TZICFIQStatus Register bit assignments

Bits	Name	Function
[31:0]	FIQStatus	Shows the status of the interrupts after masking by the TZICFIQIntEnable and TZICFIQIntEnClear Registers. A HIGH bit indicates that the interrupt is active, and generates an nFIQ interrupt to the processor.

Raw interrupt status register

The read-only TZICRawIntr Register provides the status of the source interrupts, and software interrupts, to the interrupt controller. Table 1-3 lists the register bit assignments.

Table 1-3 TZICRawIntr Register bit assignments

Bits	Name	Function
[31:0]	RawIntr	Shows the status of the interrupts before masking by the TZICFIQIntEnable and TZICFIQIntEnClear Registers. A HIGH bit indicates that the interrupt is active before masking.

Interrupt select register

The read-write TZICIntSelect Register has a reset value of `0x00000000`. It selects whether you can use the corresponding input source to generate a FIQ interrupt or whether it passes through to **TZICIRQOUT**. Table 1-4 lists the register bit assignments.

Table 1-4 TZICIntSelect Register bit assignments

Bits	Name	Function
[31:0]	IntSelect	Selects whether the interrupt source generates an FIQ interrupt or passes straight through to **TZICIRQOUT**. 0 = interrupt passes through to **TZICIRQOUT** 1 = interrupt is available for FIQ generation.

FIQ enable register

The read/write TZICFIQEnable Register has a reset value of `0x00000000`. It enables the corresponding FIQ-selected input source. This interrupt source can then generate an FIQ interrupt. Table 1-5 lists the register bit assignments.

Table 1-5 TZICFIQEnable Register bit assignments

Bits	Name	Function
[31:0]	FIQEnable	Enables the FIQ-selected interrupt lines, allowing the interrupts to reach the processor. Read: 0 = interrupt disabled 1 = interrupt enabled. You can only enable the interrupt using this register. You must use the TZICFIQEnClear Register to disable the interrupt enable. Write: 0 = no effect 1 = interrupt enabled. Resetting disables all interrupts. There is 1 bit of the register for each interrupt source.

FIQ enable clear register

The write-only TZICFIQEnClear Register clears bits in the TZICFIQEnable Register. See *FIQ enable register*. Table 1-6 lists the register bit assignments.

Table 1-6 TZICFIQEnClear Register bit assignments

Bits	Name	Function
[31:0]	FIQEnClear	Clears bits in the TZICFIQEnable Register. Writing a HIGH clears the corresponding bit in the TZICFIQEnable Register. Writing a LOW has no effect.

FIQ bypass register

The read/write TZICFIQBypass Register has a reset value of `0x00000000`. It enables **nNSFIQIN** to be routed directly to nFIQ. By doing this, it bypasses all internal TZIC logic.

Figure 1-4 on page 1-10 shows the register bit assignments.

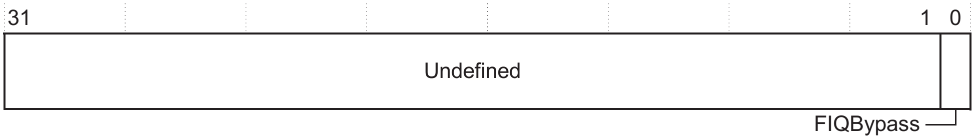

Figure 1-4 TZICFIQBypass Register bit assignments

Table 1-7 lists the register bit assignments.

Table 1-7 TZICFIQBypass Register bit assignments

Bits	Name	Function
[31:1]	-	Read undefined. Write as zero.
[0]	FIQBypass	Enables **nNSFIQIN** to route directly to nFIQ. 0=No Bypass 1=Bypass.

Protection register

The read/write TZICProtection Register has a reset value of `0x00000000`. It enables or disables protected register access, stopping register accesses when the processor is in user-mode.

Figure 1-5 shows the register bit assignments.

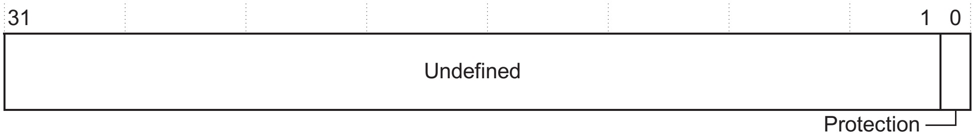

Figure 1-5 TZICProtection Register bit assignments

Table 1-8 lists the register bit assignments.

Table 1-8 TZICProtection Register bit assignments

Bits	Name	Function
[31:1]	-	Read undefined. Write as zero.
[0]	Protection	Enables or disables protected register access: 0 = protection mode disabled 1 = protection mode enabled. When enabled, you can only make privileged mode accesses (reads and writes) to the TZIC. You can only access this register in privileged mode, even when protection mode is disabled.

Lock enable register

The write-only TZICLock Register enables or disables all other register write access. You must write the access code to this register before you can modify any other register. To disable access, you must write any other value to the register. The register protects the TZIC from spurious writes. The Lock bit in the TZICLockStatus Register indicates the status of the lock. See *FIQ status register* on page 1-8. Table 1-9 lists the register bit assignments.

Table 1-9 TZICLock Register bit assignments

Bits	Name	Function
[31:0]	Lock	To enable access to the other registers in the TZIC, you must write the correct access code of `0x0ACCE550` to this register. To disable access to the other TZIC registers, you must write any other value than `0x0ACCE550` to this register.

Lock status register

The read-only TZICLockStatus Register provides the lock status of the TZIC registers.

Figure 1-6 shows the register bit assignments.

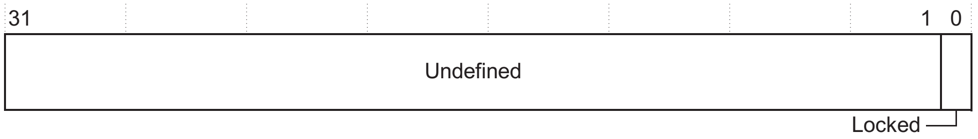

Figure 1-6 TZICLockStatus Register bit assignments

Table 1-10 lists the register bit assignments.

Table 1-10 TZICLockStatus Register bit assignments

Bits	Name	Function
[31:1]	-	Read undefined.
[0]	Locked	Shows the locked status of the TZIC: 0 = Access to the TZIC is not locked 1 = Access to the TZIC is locked (reset). You can unlock the access using the TZICLock Register.

Peripheral identification registers

The read-only TZICPeriphlD0-3 Registers are four 8-bit registers that span address locations `0xFE0` to `0xFEC`. You can treat the registers conceptually as a single 32-bit register.

Figure 1-7 shows the register bit assignments of the TZICPeriphlD0-3 registers.

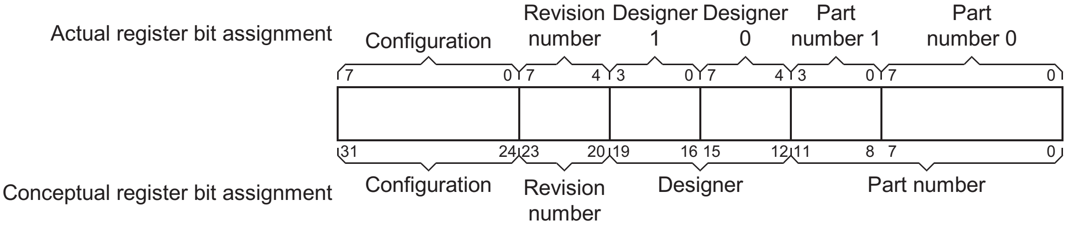

Figure 1-7 TZICPeriphID0-3 registers bit assignments

Table 1-11 lists the peripheral identification registers bit assignments.

Table 1-11 TZICPeriphID0-3 Register bit assignments

Bits	Name	Function
[31:24]	Configuration	Configuration option of the peripheral. The configuration value is 0.
[23:20]	Revision number	Revision number of the peripheral. The revision number starts from 0, and the value is revision independent.
[19:12]	Designer	Identification of the designer. ARM Limited is `0x41` (ASCII A).
[11:0]	Part number	Identification of the peripheral. Use the three-digit product code `0x890` for TZIC.

The following sections describe the four 8-bit peripheral identification registers:

- *TZICPeriphlD0 Register*
- *TZICPeriphlD1 Register*
- *TZICPeriphlD2 Register* on page 1-14
- *TZICPeriphlD3 Register* on page 1-14.

TZICPeriphID0 Register

The read-only TZICPeriphID0 Register is hard-coded and the register fields register determine the reset value.

Figure 1-8 shows the register bit assignments.

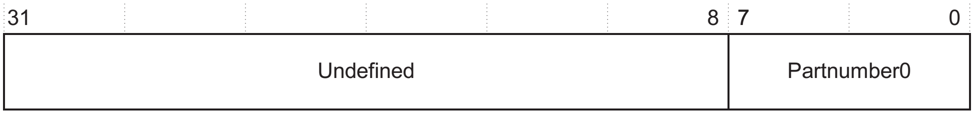

Figure 1-8 TZICPeriphIDO Register bit assignments

Table 1-12 lists the register bit assignments.

Table 1-12 TZICPeriphID0 Register bit assignments

Bits	Name	Function
[31:8]	-	Read undefined
[7:0]	Partnumber0	These bits read back as `0x90`

TZICPeriphID1 Register

The read-only TZICPeriphID1 Register is hard-coded and the fields within the register determine the reset value.

Figure 1-9 shows the register bit assignments.

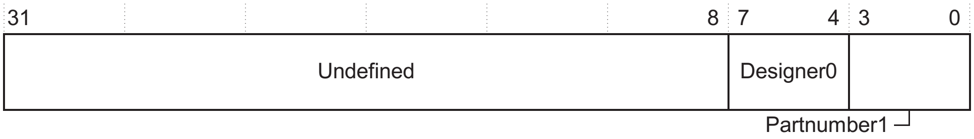

Figure 1-9 TZICPeriphID1 Register bit assignments

Table 1-13 lists the register bit assignments.

Table 1-13 TZICPeriphID1 Register bit assignments

Bits	Name	Function
[31:8]	-	Read undefined
[7:4]	Designer0	These bits read back as `0x1`
[3:0]	Partnumber1	These bits read back as `0x8`

TZICPeriphID2 Register

The read-only TZICPeriphID2 Register is hard-coded and the fields within the register determine the reset value.

Figure 1-10 shows the register bit assignments.

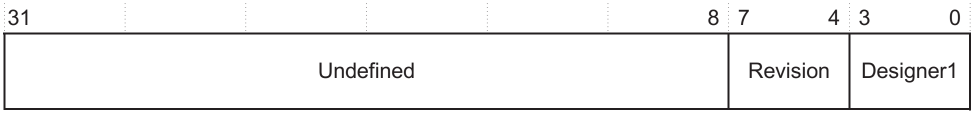

Figure 1-10 TZICPeriphID2 Register bit assignments

Table 1-14 lists the register bit assignments.

Table 1-14 TZICPeriphID2 Register bit assignments

Bits	Name	Function
[31:8]	-	Read undefined
[7:4]	Revision	These bits read back as the revision number and can be between 0 and 15
[3:0]	Designer1	These bits read back as `0x4`

TZICPeriphID3 Register

The read-only TZICPeriphID3 Register is hard-coded and the fields within the register determine the reset value.

Figure 1-11 on page 1-15 shows the register bit assignments.

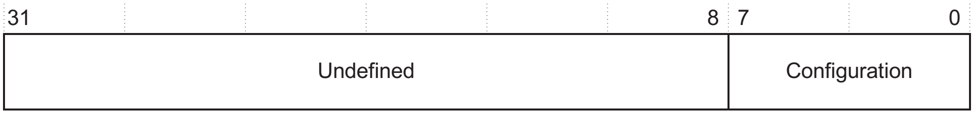

Figure 1-11 TZICPeriphID3 Register bit assignments

Table 1-15 lists the register bit assignments.

Table 1-15 TZICPeriphID3 Register bit assignments

Bits	Name	Function
[31:8]	-	Read undefined
[7:0]	Configuration	These bits read back as `0x00`

Identification registers

The read-only TZICPCellID0-3 registers are four 8-bit registers that span address locations `0xFF0` to `0xFFC`. You can treat the registers conceptually as a single 32-bit register. This register acts as a standard cross-peripheral identification system. Figure 1-12 shows the register bit assignments.

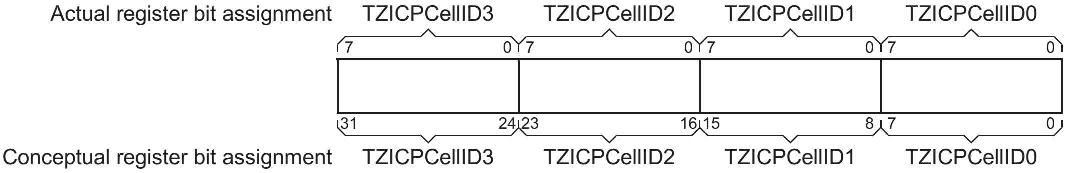

Figure 1-12 TZICPCellID0-3 Registers bit assignments

The four 8-bit peripheral identification registers are described in the following sections:

- *TZICPCellID0 Register* on page 1-16
- *TZICPCellID1 Register* on page 1-16
- *TZICPCellID2 Register* on page 1-16
- *TZICPCellID3 Register* on page 1-17.

TZICPCellID0 Register

The read-only TZICPCellID0 Register is hard-coded and the fields within the register determine the reset value. Table 1-16 lists the register bit assignments.

Table 1-16 TZICPCellID0 Register bit assignments

Bits	Name	Function
[31:8]	-	Read undefined
[7:0]	TZICPCellID0	These bits read back as `0x0D`

TZICPCellID1 Register

The read-only TZICPCellID1 Register is hard-coded and the fields within the register determine the reset value. Table 1-17 lists the register bit assignments.

Table 1-17 TZICPCellID1 Register bit assignments

Bits	Name	Function
[31:8]	-	Read undefined
[7:0]	TZICPCellID1	These bits read back as `0xF0`

TZICPCellID2 Register

The read-only TZICPCellID2 Register is hard-coded and the fields within the register determine the reset value. Table 1-18 lists the register bit assignments.

Table 1-18 TZICPCellID2 Register bit assignments

Bits	Name	Function
[31:8]	-	Read undefined
[7:0]	TZICPCellID2	These bits read back as `0x05`

TZICPCellID3 Register

The read-only TZICPCellID3 Register is hard-coded and the fields within the register determine the reset value. Table 1-19 lists the register bit assignments.

Table 1-19 TZICPCellID3 Register bit assignments

Bits	Name	Function
[31:8]	-	Read undefined
[7:0]	TZICPCellID3	These bits read back as `0xB1`

1.4 Programmer's model for test

The TZIC test registers are summarized in Table 1-1 on page 1-6. Descriptions of the registers are provided in:

- *Test control register*
- *Test input 1 register*
- *Test input 2 Register* on page 1-19
- *Test output 1 register* on page 1-20
- *Test output 2 register* on page 1-20.

1.4.1 Test control register

The read/write TZICITCR Register has a reset value of `0x00000000`. The ITEN bit in this register controls the input and output test control registers. You must only use this register in test mode.

Figure 1-13 shows the register bit assignments.

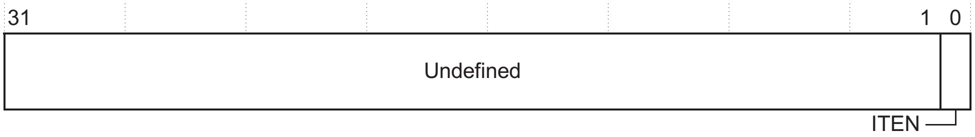

Figure 1-13 TZICITCR Register bit assignments

Table 1-20 lists the register bit assignments.

Table 1-20 TZICITCR Register bit assignments

Bits	Name	Function
[31:1]	-	Read undefined. Write as zero.
[0]	ITEN	Integration test enable: 0 = normal mode 1 = test mode enabled.

1.4.2 Test input 1 register

The read/write TZICITIP1 Register has a reset value of `0x00000440`. It controls and reads the values of the **nTZICSFIQIN** and **nTZICNSFIQIN** inputs. You must only use this register in test mode.

Figure 1-14 on page 1-19 shows the register bit assignments.

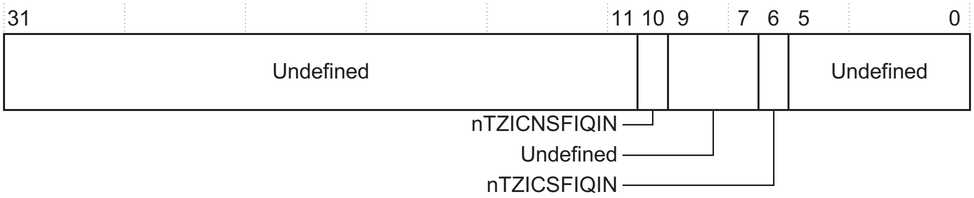

Figure 1-14 TZICITIP1 Register bit assignments

Table 1-21 lists the register bit assignments.

Table 1-21 TZICITIP1 Register bit assignments

Bits	Name	Function
[31:11]	-	Read undefined. Write as zero.
[10]	nTZICNSFIQIN	Reads the value of the **nTZICNSFIQIN** input when the TZICITCR ITEN bit is LOW. Reads the value of this field when the TZICITCR ITEN bit is HIGH. Write sets the input to the written value when the TZICITCR ITEN bit is HIGH.
[9:7]	-	Read undefined. Write as zero.
[6]	nTZICSFIQIN	Reads the value of the **nTZICSFIQIN** input when the TZICITCR ITEN bit is LOW. Reads the value of this field when the TZICITCR ITEN bit is HIGH. Write sets the input to the written value when the TZICITCR ITEN bit is HIGH.
[5:0]	-	Read undefined. Write as zero.

1.4.3 Test input 2 Register

The read/write TZICITIP2 Register has a reset value of `0x00000000`. It controls and reads the values of the **TZICINTSOURCE** input. You must only use this register in test mode.

Table 1-22 lists the register bit assignments.

Table 1-22 TZICITIP2 Register bit assignments

Bits	Name	Function
[31:0]	TZICINTSOURCE	Reads the value of the **TZICINTSOURCE** input when the TZICITCR ITEN bit is LOW. Reads the value of this field when the TZICITCR ITEN bit is HIGH. Write sets the input to the written value when the TZICITCR ITEN bit is HIGH.

1.4.4 Test output 1 register

The read/write TZICITOP1 Register has a reset value of `0x00000040`. It controls and reads the values of the **nTZICFIQ** output. You must only use this register in test mode.

Figure 1-15 shows the register bit assignments.

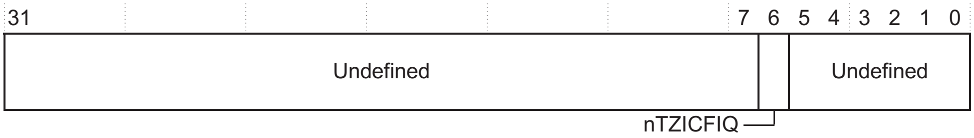

Figure 1-15 TZICITOP1 Register bit assignments

Table 1-23 lists the register bit assignments.

Table 1-23 TZICITOP1 Register bit assignments

Bits	Name	Function
[31:7]	-	Read undefined. Write as zero.
[6]	nTZICFIQ	Reads the value of the **nTZICFIQ** output when the TZICITCR ITEN bit is LOW. Reads the value of this field when the TZICITCR ITEN bit is HIGH. Write sets the output to the written value when the TZICITCR ITEN bit is HIGH.
[5:0]	-	Read undefined. Write as zero.

1.4.5 Test output 2 register

The read/write TZICITOP2 Register has a reset value of `0x00000000`. It controls and reads the values of the **TZICIRQOUT** output. You must only use this register in test mode.

Table 1-24 lists the register bit assignments.

Table 1-24 TZICITOP2 Register bit assignments

Bits	Name	Function
[31:0]	**TZICIRQOUT**	Reads the value of the **TZICIRQOUT** output when the TZICITCR ITEN bit is LOW. Reads the value of this field when the TZICITCR ITEN bit is HIGH. Write sets the output to the written value when the TZICITCR ITEN bit is HIGH.

1.5 Physical data

This section describes:

- *AC characteristics*
- *Gate count*.

1.5.1 AC characteristics

This section defines the AC characteristics.

AHB interface

Standard AHB characteristics to meet 100MHz operation. See the *AMBA Specification*.

Note

An error response is generated on **HRESP** if a SEQ or NSEQ transaction is made and the transaction size, **HSIZE**, is not 32-bit.

All other signals

There are no defined AC characteristics because the inputs are asynchronous.

1.5.2 Gate count

The total estimated gate count is 2,500 NAND2x1 equivalents.

1.6 Signal descriptions

This section describes the signals that interface with the TZIC. Figure 1-16 shows the TZIC signal connections.

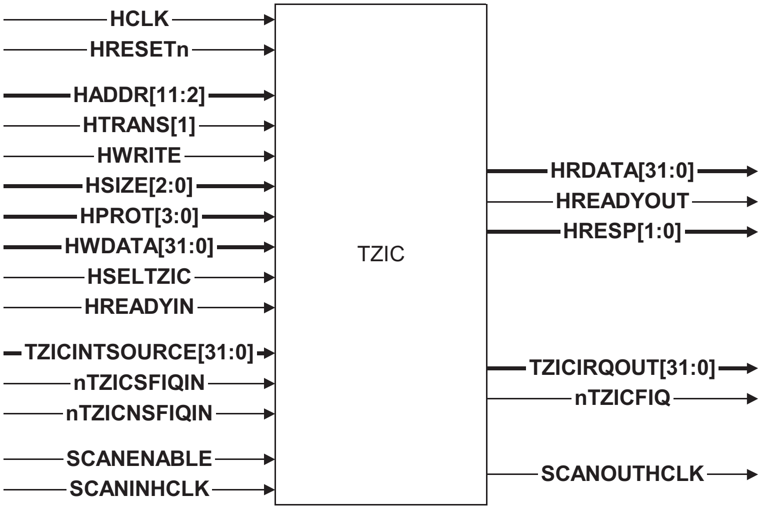

Figure 1-16 TZIC signal connections

1.6.1 AMBA signals

The AMBA signals shown in Figure 1-16 are standard AHB signals as described in the *AMBA 2 Specification*.

1.6.2 Non-AMBA signals

The TZIC module is connected to the AMBA AHB as a bus slave. Table 1-25 describes the TZIC non-AMBA signals.

Table 1-25 TZIC non-AMBA signals

Name	Type	Source/destination	Description
TZICINTSOURCE[31:0]	Input	Peripheral interrupt request	Interrupt source input
nTZICSFIQIN	Input	Additional interrupt controller	Connects to the **nFIQ** output of the previous interrupt controller if daisy chaining is employed. Connects to logic 1 if the interrupt controller is not daisy chained.
nTZICNSFIQIN	Input	Additional interrupt controller	Connects to the **nFIQ** output of the non-secure interrupt controller if **nFIQ** is required for a non-secure operating system operating on the platform. Connects to logic 1 if the interrupt controller is not daisy chained.
TZICIRQOUT[31:0]	Output	Additional interrupt controller	Connects to the interrupt input of the interrupt controller handling the **nIRQ** generation.
nTZICFIQ	Output	Core or additional interrupt controller	Fast interrupt request to processor core.

Table 1-26 describes the internal scan test control signals.

Table 1-26 Internal scan test control signals

Name	Type	Source/destination	Description
SCANENABLE	Input	Scan controller	Scan enable, for all clock domains
SCANINHCLK	Input	Scan controller	Scan data input for **HCLK** domain
SCANOUTHCLK	Output	Scan controller	Scan data output for **HCLK** domain

1.7 Revisions

This section describes the technical changes between released issues of this book.

Table 1-27 Differences between issue A and issue B

Change	Location
No technical changes	-