Intelligent Energy Controller

Revision: r0p1

Technical Reference Manual

ARM DDI 0304C

Intelligent Energy Controller
Technical Reference Manual

Release Information

The table below shows the release state and change history of this document.

Change history

Date	Issue	Confidentiality	Change
17 December 2003	A	Confidential	First release for r0p0
26 July 2005	B	Confidential	Updates for r0p1
04 June 2008	C	Non-Confidential	Second release for r0p1

Proprietary Notice

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.

Product Status

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

Web Address

`http://www.arm.com`

Contents
Intelligent Energy Controller Technical Reference Manual

List of Tables
Intelligent Energy Controller Technical Reference Manual

List of Figures
Intelligent Energy Controller Technical Reference Manual

Preface

This preface introduces the ARM *Intelligent Energy Controller Revision r0p1 Technical Reference Manual*. It contains the following sections:

- *About this manual* on page xii
- *Feedback* on page xvi.

About this manual

This is the *Technical Reference Manual* for the ARM *Intelligent Energy Controller* (IEC).

Product revision status

The r*n*p*n* identifier indicates the revision status of the product described in this manual, where:

r*n* Identifies the major revision of the product.

p*n* Identifies the minor revision or modification status of the product.

Intended audience

This document is aimed at experienced hardware and software engineers who want to use the delivered ARM SoC product in a *System-on-Chip* (SoC) design. These engineers might not have experience of working with ARM products.

The document is written for a target audience whose technical skill technical level is medium to high skilled.

Using this manual

This manual is organized into the following chapters:

Chapter 1 *Introduction*

Read this chapter for a description of the *Intelligent Energy Manager* (IEM) and IEC. It also contains typical system configurations consisting of both open loop voltage control and closed loop voltage control.

Chapter 2 *Functional Overview*

Read this chapter for a description of the major functional blocks of the IEC and detailed functional descriptions.

Chapter 3 *Programmer's Model*

Read this chapter for a description of the IEC registers and programming details.

Chapter 4 *Programmer's Model for Test*

Read this chapter for a description of the logic in the IEC for integration testing.

Appendix A *Signal Descriptions*

Read this appendix for details of the IEC signals.

Appendix B *Revisions*

Read this for a description of the technical changes between released issues of this book.

Glossary Read the Glossary for definitions of terms used in this manual.

Conventions

Conventions that this manual can use are described in:

- *Typographical*
- *Timing diagrams* on page xiv
- *Signals* on page xiv
- *Numbering* on page xv.

Typographical

The typographical conventions are:

italic Highlights important notes, introduces special terminology, denotes internal cross-references, and citations.

bold Highlights interface elements, such as menu names. Denotes ARM processor signal names. Also used for terms in descriptive lists, where appropriate.

`monospace` Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.

`mono`space Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.

`monospace italic` Denotes arguments to monospace text where the argument is to be replaced by a specific value.

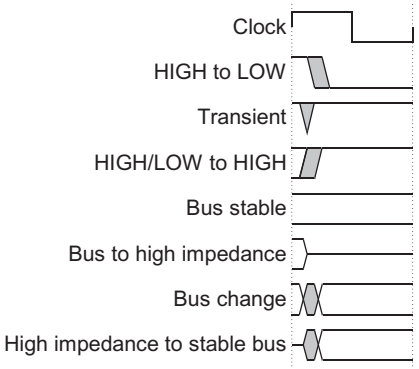
`monospace bold` Denotes language keywords when used outside example code.

< and > Angle brackets enclose replaceable terms for assembler syntax where they appear in code or code fragments. They appear in normal font in running text. For example:

- `MRC p15, 0 <Rd>, <CRn>, <CRm>, <Opcode_2>`
- The Opcode_2 value selects which register is accessed.

Timing diagrams

The figure named *Key to timing diagram conventions* 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.



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 and LOW for active-LOW signals.

Prefix H Denotes *Advanced High-performance Bus* (AHB) signals.

Prefix n Denotes active-LOW signals except in the case of AHB or *Advanced Peripheral Bus* (APB) reset signals.

Prefix P Denotes APB signals.

Suffix n AHB **HRESETn** and APB **PRESETn** reset signals.

Numbering

The numbering convention is:

<size in bits>'<base><number>

This is a Verilog method of abbreviating constant numbers. For example:

- 'h7B4 is an unsized hexadecimal value.
- 'o7654 is an unsized octal value.
- 8'd9 is an eight-bit wide decimal value of 9.
- 8'h3F is an eight-bit wide hexadecimal value of `0x3F`. This is equivalent to b00111111.
- 8'b1111 is an eight-bit wide binary value of b00001111.

Further reading

This section lists publications from both ARM Limited and third parties that provide additional information on developing code for the ARM family of processors.

ARM periodically provides updates and corrections to its documentation. See `http://www.arm.com` for current errata sheets, addenda, and the ARM Frequently Asked Questions list.

ARM publications

This document contains information that is specific to the IEC. See the following document(s) for other relevant information:

- *ARM AMBA® Specification (Rev 2.0) (ARM IHI 0011)*
- *Intelligent Energy Controller User Guide* (ARM DUI 0252)
- *Intelligent Energy Controller Integration Manual* (ARM DII 0085).

Feedback

ARM Limited welcomes feedback on the IEC, and its documentation.

Feedback on this product

If you have any comments or suggestions about this product, contact your supplier giving:

- the product name
- a concise explanation of your comments.

Feedback on this manual

If you have any comments on this manual, send email to `errata@arm.com` giving:

- the title
- the number
- the relevant page number(s) to which your comments apply
- a concise explanation of your comments.

ARM Limited also welcomes general suggestions for additions and improvements.

Chapter 1
Introduction

This chapter describes the IEM and the IEC. It contains the following sections:

1.1 About the Intelligent Energy Manager

The IEM solution is designed primarily for battery powered equipment, where the requirement to have as long a battery life as possible is paramount. The IEM solution is ideal for portable applications, for example smartphones, feature phones, *Personal Digital Assistants* (PDA), hand held games consoles and portable media players.

A complete IEM solution is made up of a number of hardware and software components. Figure 1-1 shows a high-level block diagram of a complete IEM solution.

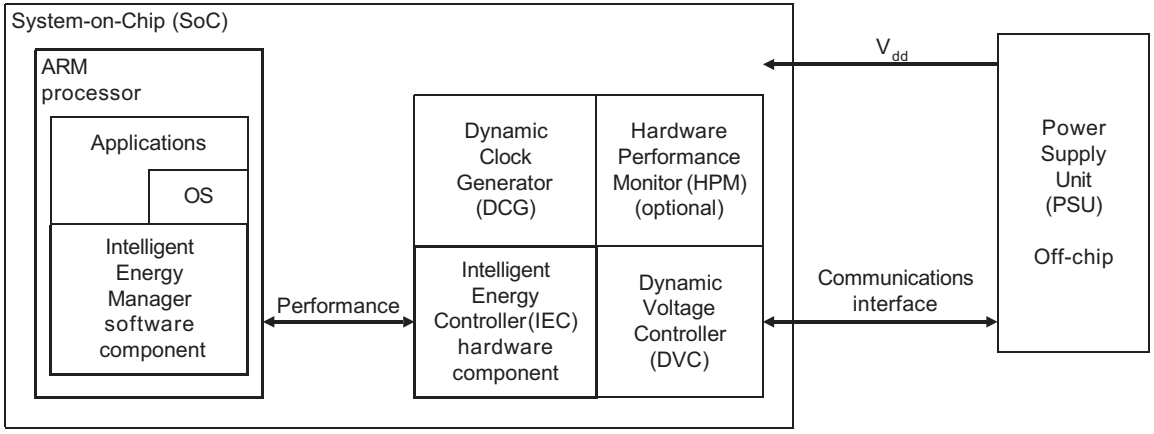


Figure 1-1 Intelligent Energy Manager solution

To build an IEM capable SoC, you require three or four on-chip components. These are required in addition to the other blocks in your SoC such as the processor and the peripherals. The additional components are:

- *Dynamic Clock Generator* on page 1-3
- *Hardware Performance Monitor* on page 1-4, this is optional and only required for a closed loop system
- *Dynamic Voltage Controller* on page 1-4
- *Intelligent Energy Controller* on page 1-5.

The *Power Supply Unit* (PSU) is the only off-chip component and is described in *PSU* on page 1-8.

The IEM software is described in *IEM software* on page 1-8.

Figure 1-2 on page 1-3 shows the on-chip IEM components required for a complete solution and how they connect together. This figure does not show the other components that are required for a complete SoC, for example a processor, memory controller, display controller or serial interfaces. Each of the IEM components is described briefly.

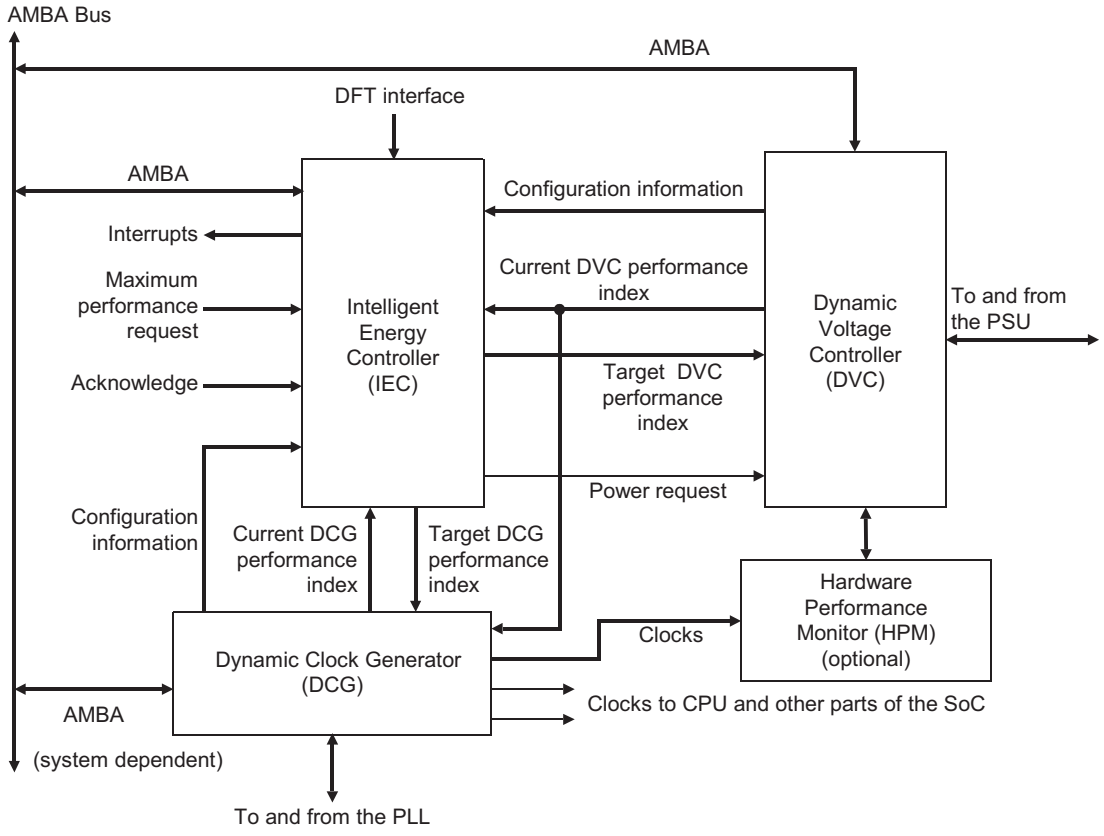


Figure 1-2 IEM on-chip components

1.1.1 Dynamic Clock Generator

The *Dynamic Clock Generator* (DCG) is not provided by ARM Limited. It is system-dependent and consequently is the responsibility of the system designer.

The DCG receives target performance requests from the IEC. It generates the necessary clocks for the SoC, for example:

- processor clock
- peripheral clocks
- AMBA clock.

Additionally, for a more efficient design, the DCG must be capable of generating the different performance levels as indicated by the IEC. The DCG also provides the IEC with configuration information, for example:

- fractional index map, indicating the fractional levels supported
- performance map, providing the mapping of the performance levels onto the clock frequencies supported by the DCG
- maximum processor performance.

The DCG can also be a memory mapped AMBA peripheral and can contain both control and status registers. This is a system design choice.

The design of the DCG must meet the requirements set by the IEC and the *Dynamic Voltage Controller* (DVC). These constraints are necessary to ensure optimum and correct performance of the Hardware Performance Monitor (HPM).

1.1.2 Hardware Performance Monitor

The *Hardware Performance Monitor* (HPM) is not provided by ARM Limited. This optional component is system-dependent and consequently is the responsibility of the system designer. The HPM is not a memory mapped device. An HPM is required for closed loop control, but not for an open loop control system.

Note

An example of an HPM is the one provided by National Semiconductor. ARM Limited can provide the National Semiconductor HPM for licensing.

The HPM is connected to the voltage domain of the SoC that is to be controlled. Each voltage domain that is to be controlled requires an HPM. Figure 1-2 on page 1-3 shows the HPM interfaces to the DVC and the DCG blocks. The DCG supplies the target frequency required by the IEM software for that voltage domain. The HPM translates voltage level into speed performance information. Based on this information, the APC determines if the optimum voltage level is achieved for the target frequency.

1.1.3 Dynamic Voltage Controller

The *Dynamic Voltage Controller* (DVC) is not provided by ARM Limited. It is system-dependent and is consequently the responsibility of the system designer.

The DVC receives the target performance requests from the IEC. It is an AMBA slave (See Figure 1-3 on page 1-10 and Figure 1-4 on page 1-12). It is required for both closed and open loop control. For open loop control, some of the features of the DVC might not be used. The DVC provides a link to an off-chip PSU.

Note

- The *Advanced Power Controller* (APC) from National Semiconductor is an example of such a DVC.
- The *Energy Management Unit* (EMU) from National Semiconductor is an example of an optimal PSU.

The DVC receives the required target performance request from the IEC. This performance request is then translated to a voltage level that is communicated to the PSU through an interface such as the *PowerWise Interface*™ (PWI). The PWI has been developed jointly by ARM and National Semiconductor to provide a high-speed, low-power control interface between an IEM-enabled SoC and an external power supply unit.

Note

- The DVC provides configuration information to the IEC that indicates the different supported voltage steps. This information is used by the IEC when setting the target performance level.
- PWI is a standard defined jointly by ARM and National Semiconductor.

For an open loop system, the DVC can either:

- wait a programmed time that is dependent on the response time of the PSU, before signaling to the DCG that the target performance can be achieved
- interrogate the PSU through the PWI for a **VDD_OK** signal indication.

If the PSU, such as the EMU provides intermediate stable voltage level indication, then the DVC can also determine this through the PWI.

An example open loop system is described in *Open loop voltage control* on page 1-9.

For a closed loop system the interface to the HPM determines when the requested voltage, and consequently performance, is achieved. An example closed loop system is described in *Closed loop voltage control* on page 1-10.

1.1.4 Intelligent Energy Controller

The *Intelligent Energy Controller* (IEC) is designed for reuse in a wide variety of AMBA based designs and has a standard APB slave interface for programming the registers. This provides an *Applications Programming Interface* (API) for the IEM software. The IEC connects through defined interfaces to SoC-specific components such as the DVC.

The IEC uses prediction performance level requests from the IEM software. The performance setting is communicated to the IEC so that the System-on-Chip specific and product platform scaling hardware can be controlled to bring the system to that performance point. Battery life is extended by lowering the operating frequency and voltage of SoC components, such as the processor, and consequently reducing energy consumption.

The IEC provides an abstracted view of the SoC-specific performance scaling hardware. It is responsible for translating the performance prediction made by the IEM software (0-100% of maximum performance) into an appropriate performance point at which the system runs and then controlling the scaling hardware to achieve operation at that target point. This is achieved through passing a target performance request to the DCG and DVC.

The IEC also measures the work done in the system to ensure that the software deadlines are not going to be missed. Additionally, the IEC supports a maximum performance hardware request feature.

The IEC is designed to map to an implementation-defined set of index levels. You must configure the IEC to define the DCG frequencies and DVC voltage levels that can be selected. These frequencies and voltages depend on the capabilities of the dynamic or adaptive power supply technology to support multiple operating performance points.

The IEC interfaces to the DCG and DVC blocks through a thermometer encoded interface protocol, that indicates to the IEC the current performance level. This protocol is specified to support interfacing across asynchronous clock domains between high-speed PLL and clock-generator and low-speed voltage scaling hardware. The IEC provides an encoded performance index to SoC specific DCG and DVC blocks.

An additional feature of the encoding is that it supports operation between one or more IEC-enabled processing subsystems and one or more DVCs. That is, it is multiprocessor enabled.

The IEC also includes a *Design for Test* (DFT) interface. This enables easier control over the scaling hardware during production testing of the SoC device.

The IEC configuration and thermometer encoded interfaces are described in more detail later in the document, see:

- *Configuration interface* on page 2-3
- *Processor frequency configuration* on page 2-18
- *DPM frequency configuration* on page 2-18
- *Multiprocessor system support* on page 2-24
- *IEC and SoC DFT* on page 2-37
- *Register descriptions* on page 3-6.

Features

The IEC is an AMBA compliant, SoC peripheral that is developed, tested, and licensed by ARM Limited. The IEC features are as follows:

- AMBA APB compliant.
- Defined interfaces between the IEC and the other on-chip peripherals that are necessary for a complete energy management solution:
  - DCG
  - DVC.
- An abstract interface to the underlying system-specific clock multiplexing and dynamic voltage or power control. This is through mapping to an implementation-defined set of index levels:
  - that correspond with the DCG frequencies that can be selected, and
  - that enable the voltage steps for the corresponding dynamic or adaptive power supply technology and consequently supports multiple operating performance points.
- An encoded interface protocol that provides a performance index to SoC specific DCG and DVC blocks.
- Signaling codes to support operation between one or more IEC-enabled processing subsystems and one or more DVCs. It is therefore multiprocessor enabled.
- *Dynamic Voltage Scaling* (DVS) emulation support enables a run fast then idle mode of operation.
- An API interface for efficient control and monitoring:
  - implementation-independent fractional performance setting interface to support performance prediction algorithms without hard-coded frequencies.
  - implementation-independent interrogation of performance-level quantization mapping levels to enable performance prediction software to adapt to the processor clock frequencies provided.
  - SoC-specific configuration interrogation, consisting of processor and IEC clock frequencies in kHz, and performance level mapping provided by the SoC specific DCG.
- Support for maximum performance signaling for real time subsystems that enables:
  - the maximum performance level to be requested regardless of the current programmed target performance level.
  - you to decide the events that activate this mode.
- Monitoring for IEM-specific algorithms, through a multi-channel interface designed to support automatic accumulation of system metrics.

- Support for synchronization handshaking with synchronous and asynchronous bridges to control entry and exit from maximum performance mode.
- Test registers for use in block and system level integration testing.
- System level integration testing using externally applied integration vectors.
- Debug mode for testing clock generation with maximum voltage.
- ID support registers for porting software driver compliance.
- DFT interface to control the target index outputs during SoC DFT.

1.1.5 PSU

The PSU is the only off-chip component and is not provided by ARM Limited. The PSU provides the requested voltage to the SoC. It interfaces to the DVC through an interface such as the PWI. It ensures that the voltage targets specified by the DVC are provided to the SoC.

1.1.6 IEM software

The IEM software component, uses information from the *Operating System* (OS) to build up a historical view of the execution of the application software running on the system.

A number of different software algorithms are applied to classify the types of activity and to analyze their processor utilization patterns. The results of each analysis are combined to make a global prediction about the future performance requirement for the system. This prediction is used by the IEC.

The IEC hardware component works in close association with the IEM software component. The IEM software is a separate, licensable product available from ARM and is not described in this document.

1.2 Typical system configuration

There are two different IEM system configurations:

- *Open loop voltage control*
- *Closed loop voltage control* on page 1-10.

1.2.1 Open loop voltage control

Figure 1-3 on page 1-10 shows an IEM enabled system with open loop voltage control. It shows an example ARM1176JZF-S processor based system including *Tightly Coupled Memories* (TCMs) and cache RAMs. Both the processor and RAM subsystems can be voltage controlled while the rest of the SoC subsystem remains at a fixed voltage. This partitioning enables the processor subsystem to be turned off with all the state information stored in the RAM subsystem that remains in a retention state. This enables the system to wake-up and execute out of the TCMs and reduce the response time to any events.

Figure 1-3 on page 1-10 also shows that the power supply provides a **VDD_OK** signal to indicate that the requested voltage is now available. This is optional and the IEC can work with either this setup or with a setup that uses timers to wait an appropriate amount of time before proceeding. You must program the timers to leave enough time for the voltage to settle.

Note

The timers must be in a peripheral in your SoC and are not part of the IEC.

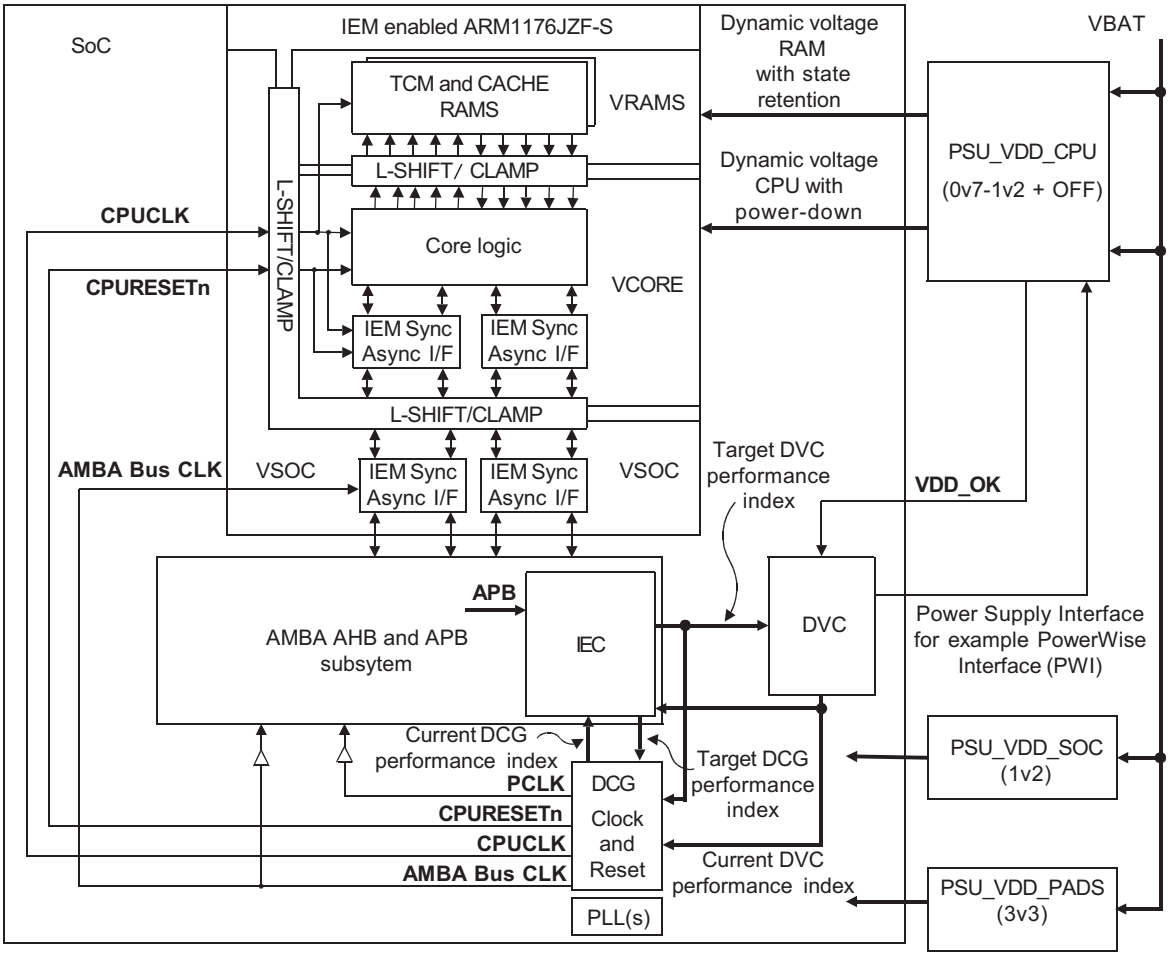


Figure 1-3 An example IEM enabled system with open loop voltage control

1.2.2 Closed loop voltage control

Figure 1-4 on page 1-12 shows an IEM enabled system with closed loop voltage control. It shows an example ARM1176JZF-S processor based system including TCMs and cache RAMs. Both the processor and RAM subsystem can be voltage controlled while the rest of the SoC subsystem remains at a fixed voltage. This partitioning enables the processor subsystem to be turned off with all the state information stored in the RAM subsystem that remains in a retention state. This enables the system to wake-up and execute out of the TCMs and reduce the response time to any events.

It includes an on-chip HPM block that is closely coupled with a DVC such as the APC. Both blocks, together with the IEM and the DCG, provide closed loop control and map the performance specified to a voltage level that can support the requested performance. In this system, a **VDD_OK** signal is not provided by the power supply. The HPM translates voltage level into speed performance information. Based on this information, the APC determines if the optimum voltage level is achieved for the target frequency.

Note

When using National Semiconductors APC, it is a requirement that each DVC performance index should be mapped to a unique DCG performance index.

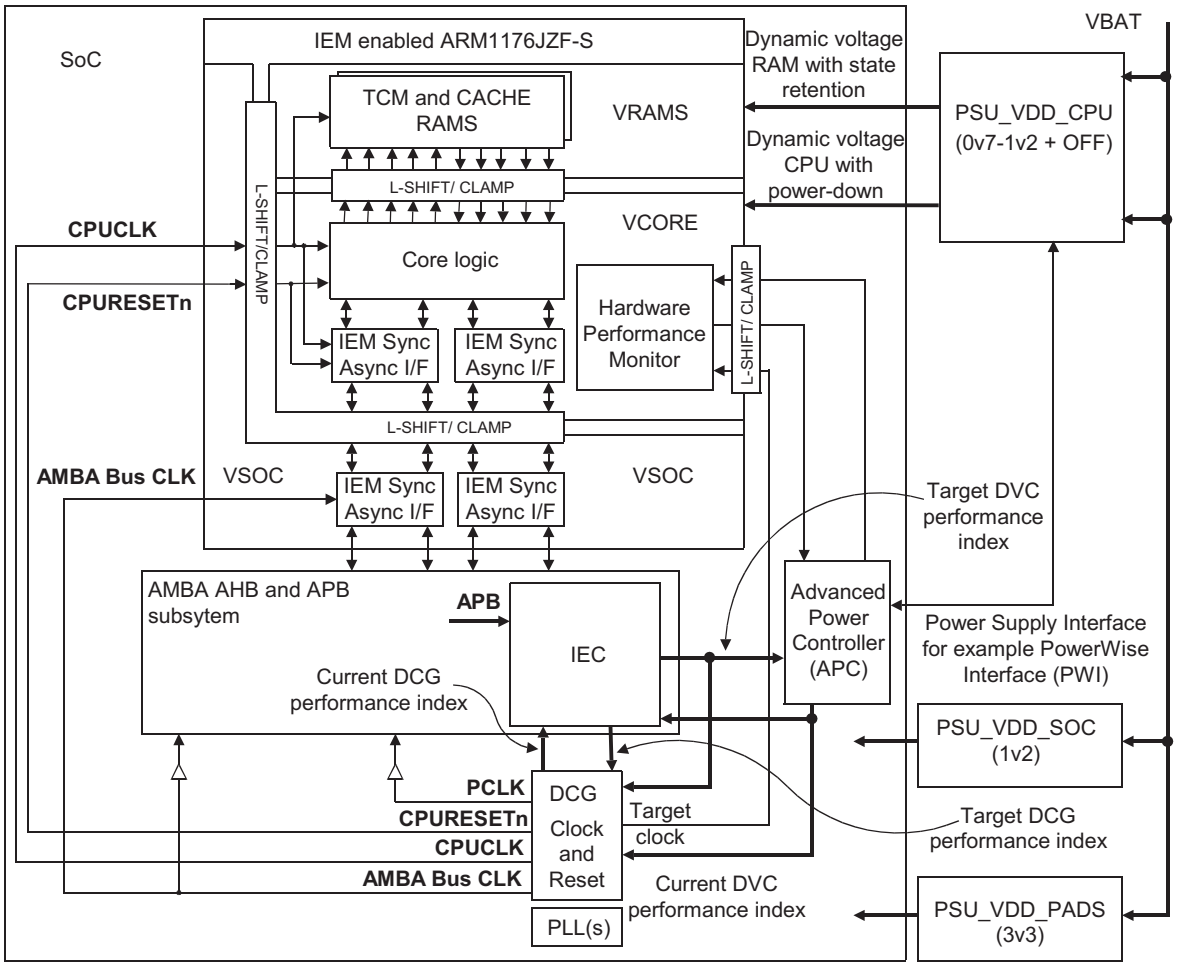


Figure 1-4 An example IEM-enabled system with closed loop voltage control

1.3 Product Revisions

This section describes differences in functionality between product revisions of the IEC:

r0p0-r0p1 Read-only IECPeriphID2 Register bits[7:4] value changes from `0x0` to `0x1` to reflect revision. See *Peripheral Identification Register 2* on page 3-24.

Chapter 2
Functional Overview

This chapter describes a functional overview and detailed functional description of the IEC. It contains the following sections:

- *Functional overview* on page 2-2
- *Functional operation* on page 2-6.

2.1 Functional overview

Figure 2-1 shows a simplified top-level diagram of the IEC with its main functional blocks.

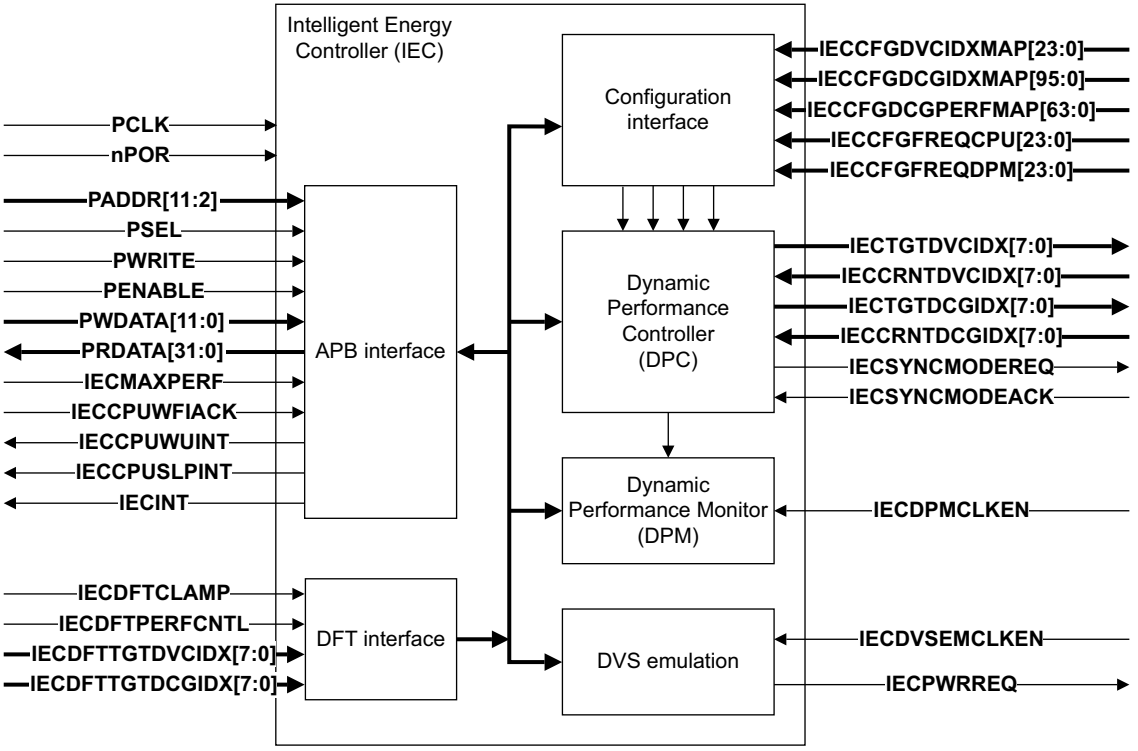


Figure 2-1 IEC simplified block diagram

The functional operation of the IEC is subdivided into the following:

- *APB interface* on page 2-3
- *Configuration interface* on page 2-3
- *Dynamic Performance Controller (DPC)* on page 2-4
- *Dynamic Performance Monitor (DPM)* on page 2-4
- *DVS emulation* on page 2-4.
- *IEC clock* on page 2-4
- *IEC reset* on page 2-5

2.1.1 APB interface

The APB interface block provides the IEM software access to the IEC registers. This module contains all of the registers and performs the register read and write address decoding. A standard API interface is defined that enables the IEM software to communicate with the *Dynamic Performance Controller* (DPC), *Dynamic Performance Monitor* (DPM), and *Pulse Width Modulation* (PWM) based DVS emulation blocks.

The APB interface also enables software to get status and diagnostic information about the support provided by the system. The register bank uses a 32-bit wide APB interface.

2.1.2 Configuration interface

The IEC is made reusable through a wide hardwired configuration interface. This configuration interface is necessary because of a number of system and technology specific dependencies:

- The clock generator is system and technology dependent and there might be a limited number of divider ratios available, for example:
  - — the system can have specific requirements on clock mark space ratios and consequently might only be able to tolerate square clocks
  - — the number of PLLs, and consequently clocks, that are available is system dependent.
- Voltage scaling is highly technology specific and consequently the number of voltage steps that can be supported are restricted.

Index mappings offer the highest efficiency and consequently the configuration interface specifies:

- The fractional performance points that are supported. This is through the **IECCFGDCGIDXMAP** inputs.
- The performance point for each of the DCG index levels. This is through the **IECCFGDCGPERFMAP** inputs.
- The DVC index values to determine the relationship between the DVC index levels and the corresponding DCG index levels. This is through the **IECCFGDVCIDXMAP** inputs. This enables the IEC to support multiple frequencies at each available voltage level.

Additionally, the configuration interface also specifies:

- The maximum processor clock frequency in kHz. This is through the **IECCFGFREQCPU** inputs.
- The DPM clock frequency in kHz. This is through the **IECCFGFREQDPM** inputs.

Diagnostic support for integration validation is included in the programmer's model to verify that the index level and fractional performance mappings are monotonic.

See *Configuration setting* on page 2-9 for more information about the configuration interface and the encoding of the various inputs that permit this configuration.

2.1.3 Dynamic Performance Controller (DPC)

The DPC block maps the target performance level into index values for the DCG and DVC interfaces. It also performs reverse mapping when the IEM software reads the current performance level.

See *Functional operation* on page 2-6 for more details.

2.1.4 Dynamic Performance Monitor (DPM)

The IEC provides hardware support for the IEM software to monitor the work done for various tasks. Three channels are available. Each channel is in the form of a 64-bit accumulator that counts the fractional performance every **PCLK** clock cycle when **IECDPMCLKEN** is HIGH.

Channel 1 accumulates at the processor fractional performance while channels 2 and 3 accumulate at a software programmed rate.

See *DPM channels* on page 2-30 for more information.

2.1.5 DVS emulation

The DVS emulation mode of the IEC enables performance scaling to be performed on systems that do not implement voltage scaling. This is done through a PWM style power request.

The PWM rate of this output depends on the current performance target specified by the IEM software. This enables re-use of the DVS interface abstraction to emulate multiple levels of performance on a system that only supports basic run-fast and idle modes of operation.

See *DVS emulation with PWM* on page 2-31 for more details.

2.1.6 IEC clock

The IEC is clocked by the AMBA APB clock signal **PCLK**. After reset this clock must be free-running and never stopped.

2.1.7 IEC reset

The IEC is reset when **nPOR** is asserted. This input must be tied to the power-on reset of the SoC. The IEC must maintain state over all other resets including soft resets. Consequently, the IEC reset input must not be tied to the APB reset signal **PRESETn**.

The **nPOR** input to the IEC can be asynchronously asserted but must be deasserted synchronous to **PCLK**.

2.2 Functional operation

The function of the IEC is to convert the fractional performance level requested by the IEM software to a performance level as supported by the DCG and DVC. This conversion depends on the number of performance levels supported by the DCG and DVC.

This section describes:

- *Performance level programming* and explains:
  - the format of the fractional numbers used for performance level setting and how these are interpreted by the IEC
  - the requirements on the minimum and maximum performance levels that must be supplied by the DCG
- *Configuration setting* on page 2-9 and explains:
  - why this is necessary with respect to making the IEC reusable over a wide range of systems
  - how the IEC quantizes the software programmed performance levels to those supported by the hardware
- *Target index and current index* on page 2-19 and explains:
  - the coding used on the IEC interfaces to the DCG and DVC
- *Maximum performance request IECMAXPERF signal* on page 2-24
- *Synchronization mode handshaking signals* on page 2-26
- *Maskable interrupts* on page 2-28
- *Acknowledgement of WaitForInterrupt signal* on page 2-29
- *DPM channels* on page 2-30
- *DVS emulation with PWM* on page 2-31
- *IEC and software development support* on page 2-36
- *IEC and SoC DFT* on page 2-37.

2.2.1 Performance level programming

The performance is expressed as the fraction of the maximum frequency required to ensure that the programming interface is portable and scalable across designs as they migrate with technology and product evolution. The primary interface for the IEM software is an implementation-independent programming mechanism to set the target performance level. An 8-bit fractional programming interface is provided to support up to 129 levels. Thirty-three levels are supported in this version of the IEC.

Note

Only 33 of the 129 levels are supported in this version of the IEC. The lowest two bits are discarded in hardware.

The IECDPCTGTPERF Register must be used to set this performance level. The format for programmable bits of the IECDPCTGTPERF Register has an inherent binary point in it. This is as listed in columns one and two of Table 2-1.

Table 2-1 Fractional performance map

IECDPCTGTPERF[7:0]		Fractional(decimal)	%	Performance level
Programmed bit value	Fractional bit value			
b'1xxxxxxx[a]	RESERVED	RESERVED	-	-
b'10000000	b'1.00000xx	1.00000	100.0%	32
b'011111xx	b'0.11111xx	0.96875	96.9%	31
b'011110xx	b'0.11110xx	0.93750	93.8%	30
b'011101xx	b'0.11101xx	0.90625	90.6%	29
b'011100xx	b'0.11100xx	0.87500	87.5%	28
b'011011xx	b'0.11011xx	0.84375	84.4%	27
b'011010xx	b'0.11010xx	0.81250	81.3%	26
b'011001xx	b'0.11001xx	0.78125	78.1%	25
b'011000xx	b'0.11000xx	0.75000	75.0%	24
b'010111xx	b'0.10111xx	0.71875	71.9%	23
b'010110xx	b'0.10110xx	0.68750	68.8%	22
b'010101xx	b'0.10101xx	0.65625	65.6%	21
b'010100xx	b'0.10100xx	0.62500	62.5%	20
b'010011xx	b'0.10011xx	0.59375	59.4%	19
b'010010xx	b'0.10010xx	0.56250	56.3%	18
b'010001xx	b'0.10001xx	0.53125	53.1%	17
b'010000xx	b'0.10000xx	0.50000	50.0%	16

Table 2-1 Fractional performance map (continued)

IECDPCTGTPERF[7:0]		Fractional(decimal)	%	Performance level
Programmed bit value	Fractional bit value			
b'001111xx	b'0.01111xx	0.46875	46.9%	15
b'001110xx	b'0.01110xx	0.43750	43.8%	14
b'001101xx	b'0.01101xx	0.400625	40.6%	13
b'001100xx	b'0.01100xx	0.37500	37.5%	12
b'001011xx	b'0.01011xx	0.34375	34.4%	11
b'001010xx	b'0.01010xx	0.31250	31.3%	10
b'001001xx	b'0.01001xx	0.28125	28.1%	9
b'001000xx	b'0.01000xx	0.25000	25.0%	8
b'000111xx	b'0.00111xx	0.21875	21.9%	7
b'000110xx	b'0.00110xx	0.18750	18.8%	6
b'000101xx	b'0.00101xx	0.15625	15.6%	5
b'000100xx	b'0.00100xx	0.12500	12.5%	4
b'000011xx	b'0.00011xx	0.09375	9.4%	3
b'000010xx	b'0.00010xx	0.06250	6.3%	2
b'000001xx	b'0.00001xx	0.03125	3.1%	1
b'00000000	b'0.0000000	0.00000	0.0%	0

a. x is an ignored bit.

Minimum performance levels required

The system-specific DCG is rarely able to generate all of the 33 possible performance levels explicitly so the requested performance level is quantized by rounding up to the next highest supported performance level. There is however a minimum requirement on the basic performance levels that the DCG must support. These are:

- the maximum value (level 32) of 100%
- the idle or clock-stopped case (level 0), that is 0%.

Note

See *Performance quantization* on page 2-17 for more details.

2.2.2 Configuration setting

Each performance level specified by the IEM software must be translated into the next highest available processor frequency, and therefore an index value. Two different index values are required:

- one for the target frequency available for the DCG
- one for the target voltage level for the DVC.

A configuration interface has been defined to make the IEC reusable because the number of levels supported by the DCG and DVC are implementation dependent.

This configuration interface enables you to map the software programmed fractional performance levels into system specified indexes. This has the advantage of supporting various SoC independent control algorithms and avoids table look up overheads in the OS. The configuration interface definition has also been implemented to enable mapping into non-linear coarse clock frequencies. It also permits an inverse mapping to be implemented in hardware for performance monitoring.

Figure 2-2 on page 2-10 shows IEC performance level flow. A target performance is converted into one of 33 performance levels. Through a process described in *DCG index mapping configuration* on page 2-11, using the DCG index mapping signal **IECCFGDCGIDXMAP[95:0]** and the 33 performance levels, a DCG index map (0-8) is created. An output target signal **IECTGTDCGIDX[7:0]** carries the DCG index value to both the DCG and a converter that converts it to a voltage level index.

At the DCG an index signal **IECCRNTDCGIDX[7:0]** is produced. The DCG index value is carried through this signal. The DCG index value is converted back into a current performance level by use of the DCG fractional performance signal **IECCFGDCGPERFMAP[63:0]** through a process described in *DCG fractional performance level mapping* on page 2-14.

At the voltage mapping procedure, described in *DVC index level mapping* on page 2-14, conversion of the DCG index value to a voltage level is accomplished by using the DVC index mapping signal **IECCFGDVCIDXMAP[23:0]**. This voltage level is then carried to the DVC.

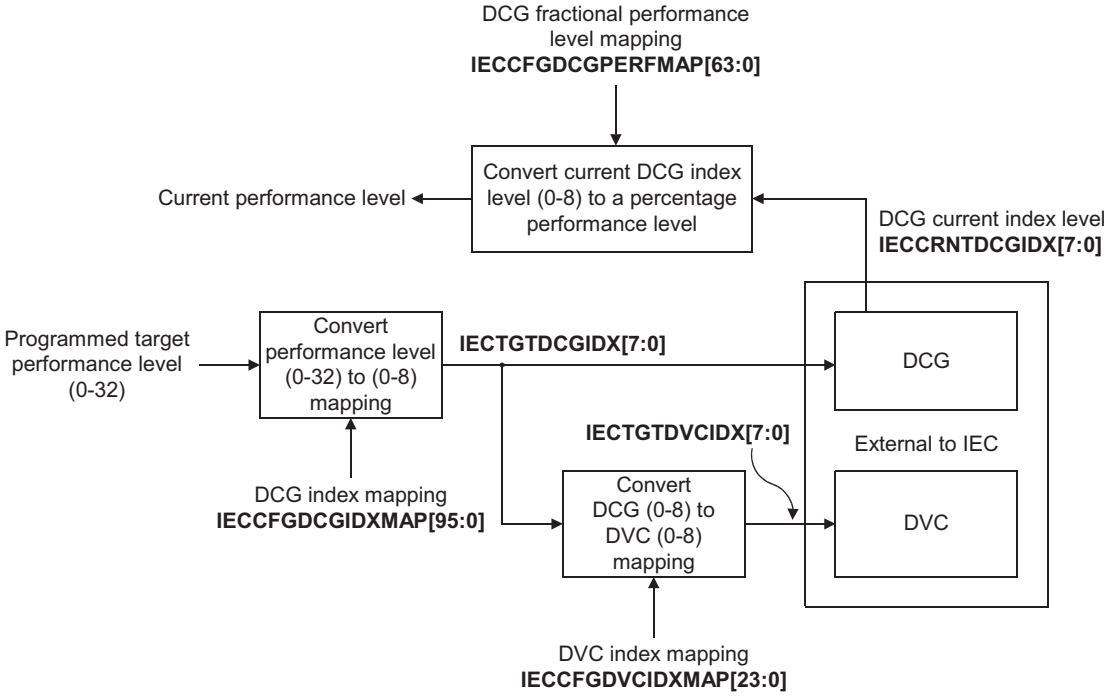


Figure 2-2 IEC performance level flow

A number of configuration bits are defined. These are defined for a system that can supply up to a maximum of eight non-zero performance levels. These are described in the following sections:

- *DCG index mapping configuration* on page 2-11
- *DCG fractional performance level mapping* on page 2-14
- *DVC index level mapping* on page 2-14
- *Performance quantization* on page 2-17
- *Processor frequency configuration* on page 2-18
- *DPM frequency configuration* on page 2-18.

DCG index mapping configuration

Each of the 32 active performance levels is assigned one of 8 DCG indexes using a 3-bit value for each performance level. The 96 bit wide **IECCFGDCGIDXMAP** signal defines these levels (see *Performance level programming* on page 2-6). Table 2-2 shows the wiring of each 3-bit slice that defines the performance level in the DCG for that slice.

Table 2-2 IECCFGDCGIDXMAP signal 3-bit slicing and performance level

IECCFGDCGIDXMAP bits	Performance level
[2:0]	1
[5:3]	2
[8:6]	3
. . .	. . .
[89:87]	30
[92:90]	31
[95:93]	32

Note

The index for performance level 0 is implicitly index 0, and so no **IECCFGDCGIDXMAP** bits are required to specify performance level 0.

Table 2-3 on page 2-13 shows that each of the 3-bit slices shown in Table 2-2 has a binary coding.

Table 2-3 IECCFGDCGIDXMAP slice coding

Binary coding	Octal value	Index level
b’000	0	1
b’001	1	2
b’010	2	3
b’011	3	4
b’100	4	5

Table 2-3 IECCFGDCGIDXMAP slice coding (continued)

Binary coding	Octal value	Index level
b’101	5	6
b’110	6	7
b’111	7	8

Note

You must wire the **IECCFGDCGIDXMAP** bits so that it represents monotonically increasing index values. Failure to do so results in unpredictable behavior of the system.

You must tie the **IECCFGDCGIDXMAP** signal to map performance levels to the supported hardware index levels. This is system dependent. If for example you tie the **IECCFGDCGIDXMAP** signal bits to the following octal value: 777777777666665555555554444444444 (expressed in Verilog as: 96'o777777777666665555555554444444444), then this can be shown in Figure 2-3 on page 2-13.

Note

- the performance levels can be non-linear
- the tie-offs for the bits must be monotonically increasing
- the index levels are 1-8, but the tie-off values are 0-7.

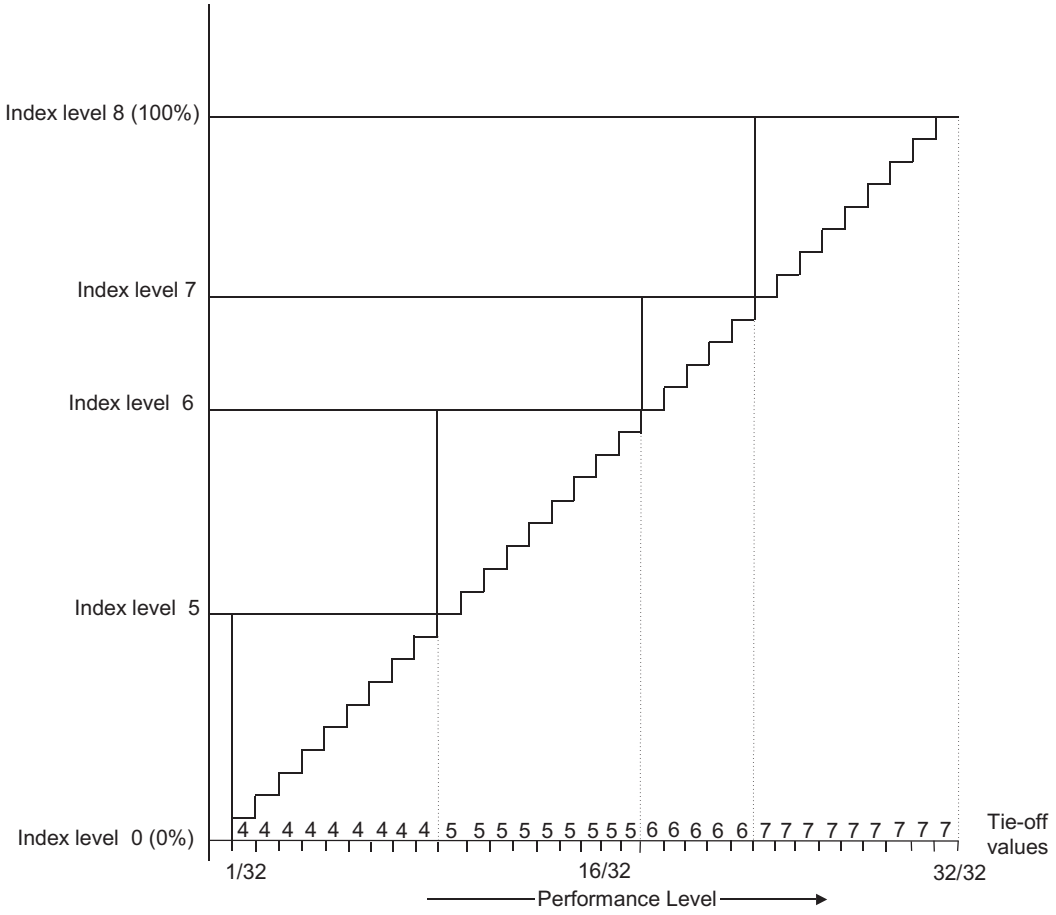


Figure 2-3 IECCFGDCGIDXMAP example

If for example the software programs a performance level of 50%, the IEC works out that this is equivalent to a value of 16 on the performance level scale. However, the IEC performs performance quantization on this value (see *Performance quantization* on page 2-17) and rounds the value up to the next value supported by the hardware. In this example the value is rounded up to 18. Table 2-1 on page 2-7 shows that the value of 18 is equivalent to 56.3%.

The value of 18 is equal to the 3-bit slice of bits [53:51] from the **IECCFGDCGIDXMAP** signal. These bits are tied to the binary coding value b'101. This equates to an index level of 6 in Table 2-3 on page 2-11.

DCG fractional performance level mapping

You must use this configuration interface to indicate the fractional performance level supported at each index level. The 64-bit wide **IECCFGDCGPERFMAP** signal is made up of 8x8-bit slices that support the eight non-zero index levels in the DCG.

Therefore, the wiring of each 8-bit slice corresponds to the fractional performance level for that index level. For example, Table 2-4 shows the values that would result if you wired 8 slices up with the following hexadecimal numbers: `80_5C_48_24_00_00_00_00`.

Table 2-4 IECCFGDCGPERFMAP example

Slice and index level	Hex value	Binary value	Fractional value	Performance level (%)
1	00	b'00000000	0.0000000	0%
2	00	b'00000000	0.0000000	0%
3	00	b'00000000	0.0000000	0%
4	00	b'00000000	0.0000000	0%
5	24	b'00100100	0.0100100	28.1%
6	48	b'01001000	0.1001000	56.3%
7	5C	b'01011100	0.1011100	71.9%
8	80	b'10000000	1.0000000	100%

Note

Table 2-4 shows that even though eight non-zero performance levels are permitted, only four are actually required because the DCG can only generate four frequencies

The IEC uses the wiring of the **IECCFGDCGPERFMAP** signal to report the current performance level that the processor is running at. That is, when a read of the IECDPCCRNTPERF Register is performed, the value returned is based on the current index level, as indicated by the **IECCRNTDCGIDX** inputs, and the state of the **IECCFGDCGPERFMAP** inputs.

DVC index level mapping

This configuration enables the IEC to support multiple frequencies at each available voltage level. This 24-bit wide **IECCFGDVCIDXMAP** signal is made up of 8x3-bit slices. Each 3-bit slice corresponds to an index level for the DVC. This configuration enables the IEC to support multiple frequencies at each available voltage level.

Table 2-5 shows the format of the 24-bit **IECCFGDVCIDXMAP** inputs.

Table 2-5 IECCFGDVCIDXMAP signal bits and DCG slice index level

IECCFGDVCIDXMAP bits	DCG index level
[2:0]	1
[5:3]	2
[8:6]	3
[11:9]	4
[14:12]	5
[17:15]	6
[20:18]	7
[23:21]	8

Table 2-6 shows that each of the 3-bit slices shown in Table 2-6 has a binary coding.

Table 2-6 IECCFGDVCIDXMAP slice coding

Binary coding	Octal value	DVC index level
b’000	0	1
b’001	1	2
b’010	2	3
b’011	3	4
b’100	4	5
b’101	5	6
b’110	6	7
b’111	7	8

Note

You must wire the **IECCFGDVCIDXMAP** bits must be so that it represents monotonically increasing index values. Failure to do so results in unpredictable behavior of the system.

The DVC index map can be designed to have a different voltage level for each DCG index level:

- If, for example, the **IECCFGDVCIDXMAP** bits are tied to octal value `77666650` (expressed in Verilog as `24'o77666650`), then this forms the relationship shown in Table 2-7 between the DCG and DVC indexes. There is no longer a unique voltage level for each DCG index level.

Table 2-7 IECCFGDVCIDXMAP example 2

DCG index level	DVC index level	Octal value
1	1	0
2	6	5
3	7	6
4	7	6
5	7	6
6	7	6
7	8	7
8	8	7

- If for example the **IECCFGDVCIDXMAP** bits are tied to octal value `76543210` (expressed in Verilog as `24'o76543210`), then this forms the relationship shown in Table 2-8 on page 2-17 between the DCG and DVC indexes. There might be a unique voltage level for each DCG index level.

Note

— you must tie-off the bits so that they represent monotonically increasing index values

— the index levels are 1-8, but the tie-off values are 0-7.

Table 2-8 IECCFGDVCIDXMAP example 1

DCG index level	DVC index level	Octal value
1	1	0
2	2	1
3	3	2
4	4	3
5	5	4
6	6	5
7	7	6
8	8	7

As with the previous example, you must tie-off the bits so that they represent monotonically increasing index values.

From this example it can be seen that this configuration of the IEC supports multiple frequencies per voltage level for DVC index levels at 8 and 7, and that only four voltage levels are available.

Note

The maximum number of non-zero frequencies supported is fixed at eight.

Performance quantization

The DCG is only able to support up to 8 non-zero performance indexes. These 8 indexes are mapped onto the 32 possible performance levels that might be requested using the **IECCFGDCGPERFMAP** input bits. The configuration must be arranged so that the performance index associated with a level is always equal to or greater than that specified by the level.

For example, consider the case where the levels shown in Table 2-4 on page 2-14 are available in the system. In this case, if the software programs a performance target of 46.9%, that is 15/32, index level 6 is chosen using the configuration mapping, providing an actual performance of at least 18/32.

Similarly if software programs 22/32, then this is mapped to index level 7, providing 71.9%, and if it programs 29/32, then the IEC uses index level 8, providing 100%.

Processor frequency configuration

This configuration is the maximum processor clock frequency in kHz. The **IECCFGFREQCPU** input is a 24-bit binary value, that gives the clock frequency of the processor in kHz. Table 2-9 lists three examples.

Table 2-9 IECCFGFREQCPU examples

IECCFGFREQCPU [23:0]	Verilog expression	Processor frequency
0x004E20	24'd020_000	20000kHz = 20MHz
0x03A980	24'd240_000	240000kHz = 240MHz
0x0003E8	24'd001_000	1000kHz = 1MHz

DPM frequency configuration

This configuration is the DPM frequency in kHz. The **IECCFGFREQDPM** input is a 24-bit binary that gives the rate that the DPM is accumulating in kHz. This value indicates the frequency that the **IECDPMCLKEN** input is pulsed. Table 2-10 lists three examples.

Table 2-10 IECCFGFREQDPM examples

IECCFGFREQDPM [23:0]	Verilog expression	DPM accumulate rate
0x004E20	24'd020_000	20000kHz = 20MHz
0x002710	24'd010_000	10000kHz = 10MHz
0x0003E8	24'd001_000	1000kHz = 1MHz

The IEM software uses the above configuration to convert the DPM counter values back to real frequencies.

2.2.3 Target index and current index

The IEC interfaces to two on-chip components to set the target performance level and getting the current performance level, see also *About the Intelligent Energy Manager* on page 1-2. These are:

- the DCG
- the DVC.

The interfaces to both the DCG and DVC pass index values to and from the IEC to indicate current and target performance levels. Both the DCG and DVC blocks can be in different clock domains to the IEC and consequently any coding of these interfaces must be robust for asynchronous clock domains, that is, support a clean synchronization across the clock domains. In addition, the coding is defined to support easy extension for multiprocessor systems.

A thermometer coded interface, for example see Table 2-11 on page 2-20, is chosen to indicate the indexes for the target and current performance interfacing to both the DCG and the DVC. The following signals are all coded using this scheme:

- **IECTGTDCGIDX**
- **IECCRNTDCGIDX**
- IECTGTDVCIDX
- **IECCRNTDVCIDX**.

This coding scheme enables single-ended signaling, without a requirement for an acknowledge, across asynchronous clock domains. There is a best level implicit acknowledge in the scheme. The following subsections describe the coding schemes:

- *Target index coding* on page 2-20
- *Current index coding* on page 2-21
- *Example 4-level index coding* on page 2-21
- *Performance requirement optimization* on page 2-22
- *Multiprocessor system support* on page 2-24.

Target index coding

The IEC interfaces support a maximum of eight non-zero performance levels, that is, performance levels 0 to 8 are supported. Table 2-11 shows the target performance levels that are coded in 8-bit buses.

Table 2-11 Target index coding for DVC and DCG

DCG and DVC target index	Code	Note
8	8'b11111111	Performance level 8 select (maximum level)
7	8'b01111111	Performance level 7 select
6	8'b00111111	Performance level 6 select
5	8'b00011111	Performance level 5 select
4	8'b00001111	Performance level 4 select
3	8'b00000111	Performance level 3 select
2	8'b00000011	Performance level 2 select
1	8'b00000001	Performance level 1 select (minimum level)
0	8'b00000000	Retention level (clock stopped)

From analyzing Table 2-11, you can see that, for a maximum performance level index, all bits are set to 1. Stepping through each performance level, the most significant bit from the previous performance level is set to zero. To decode the current performance level, the most significant bit that is set to 1 determines the performance level.

Note

- When bit 0 is set to zero, it signifies 0% performance level. That is, you can stop the clock if required, and you can lower the voltage to the processor to the retention level.
- This interface supports future extension to cope with clocks derived from more than one PLL. This is done by expanding the number of bits used for the DCG and DVC target index.

Current index coding

Table 2-12 shows the extensible coding scheme that are coded in 8-bit buses. This is used to determine the valid current level of performance.

Table 2-12 Current index coding for DVC and DCG

DCG and DVC current index	Code	Note
8	8'b1xxxxxxx	Performance level 8 valid (MAX level)
7	8'b01xxxxxx	Performance level 7 valid
6	8'b001xxxxx	Performance level 6 valid
5	8'b0001xxxx	Performance level 5 valid
4	8'b00001xxx	Performance level 4 valid
3	8'b000001xx	Performance level 3 valid
2	8'b0000001x	Performance level 2 valid
1	8'b00000001	Performance level 1 valid (MIN level)
0	8'b00000000	Retention level (Clock stopped)

Example 4-level index coding

Table 2-13 on page 2-22 shows that the IEC interfaces support eight non-zero index levels. Both the current and target performance level indexes are coded in 8-bit buses. If your system only implements five performance levels, that is, four non-zero levels, then you must connect the most significant bits to and from the IEC for the following signals:

- **IECTGTDVCIDX**
- **IECCRNTDVCIDX**
- **IECTGTDCGIDX**
- **IECCRNTDCGIDX**.

Note

You must tie the unused bits in the **IECCRNTDVCIDX** and **IECCRNTDCGIDX** signals to 1.

Table 2-13 Example performance level index coding

Performance level	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Performance level 8 (maximum)	1	1	1	1	x	x	x	x
Performance level 7	0	1	1	1	x	x	x	x
Performance level 6	0	0	1	1	x	x	x	x
Performance level 5 (minimum)	0	0	0	1	x	x	x	x
Performance level 4/3/2/1/0 (clocks stopped)	0	0	0	0	x	x	x	x

Note

- Bits [3:0] are treated as don't care. In Table 2-13 only performance levels 5-8 are used.
- For the maximum performance level index, bits [7:4] are set to 1.
- The minimum performance level is when bit 4 is 1 while bits [7:5] are 0.
- When bits [7:4] are 0, the clocks can be stopped.

Performance requirement optimization

The coding scheme enables performance requirement optimization as long as the DVC block provides the encoded index information to reflect the availability of stable voltage for each of the performance levels. For example, if the current performance is at a low level and the target performance level is set to maximum. The sequence of events is:

1. IEC requests
   - maximum performance to DVC
   - maximum frequency to DCG
2. DVC commands PSU to go to maximum voltage
   - DVC monitors feedback from PSU to determine when the next highest supported clock frequency can be changed to, as `Vdd` rises to `Vmax`, then:
   - DVC indicates to DCG to move to a new frequency.
3. All intermediate supported frequency points can be used, and as a result as much work as possible can be achieved on the way to `Vmax`.

The sequence of events is depicted in Figure 2-4. You can see that the **IECCRNTDVCIDX** values change as each intermediate stable voltage is reached. Figure 2-4 also shows how the **IECCRNTDVCIDX** changes when the voltage is lowered, 100% to 50% transition, and also what happens when a subsequent increase in performance is requested, 50% to 75% transition.

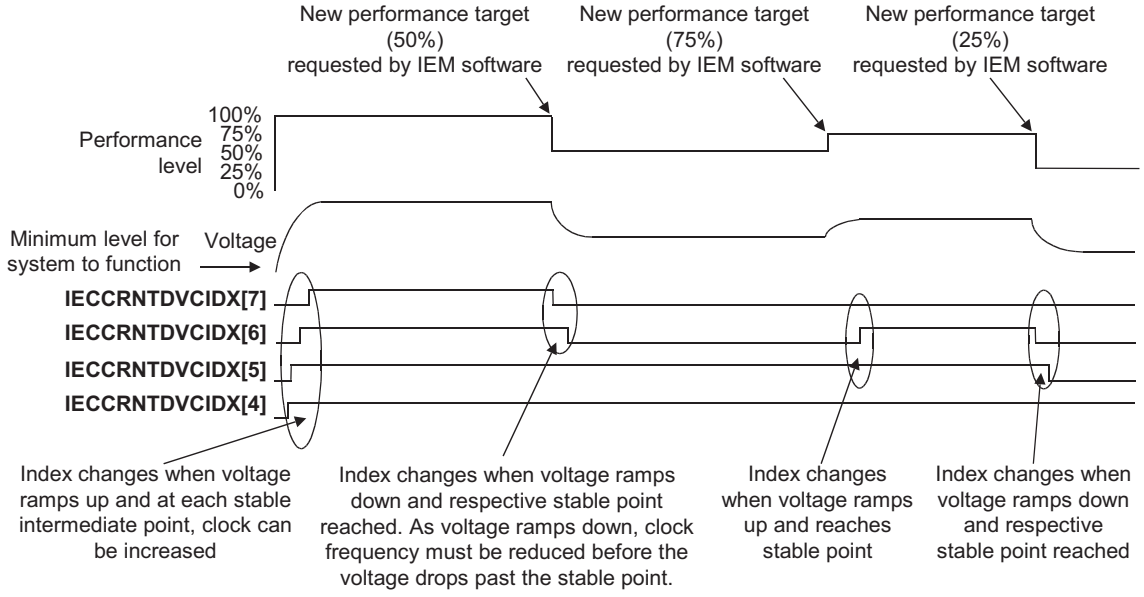


Figure 2-4 DVS and performance requirement optimization

The **IECCRNTDVCIDX** output from the DVC is also used by the DCG to supply the appropriate clock at each index position.

The DCG uses the same encoding on the **IECCRNTDCGIDX** output and Table 2-14 shows an example of clock ready coding where the DCG indicates the availability of clean stable clocks by setting the appropriate bit to HIGH. This shows the case where the DCG only supports four performance levels.

Table 2-14 Example clock ready coding

Clock level valid	[7]	[6]	[5]	[4]
Performance level clock 4	1	x	x	x

Table 2-14 Example clock ready coding (continued)

Clock level valid	[7]	[6]	[5]	[4]
Performance level clock 3	0	1	x	x
Performance level clock 2	0	0	1	x
Performance level clock 1	0	0	0	1

Multiprocessor system support

The thermometer coding schemes support the basic operations required to handle performance level comparisons and the determination of maximum or minimum levels in a multiprocessor system. In such a system, it is expected that there is one IEC per processor:

- The logical OR of two levels results in the higher of the two levels. For example, in a 2-processor system, the DVC index can be worked out using:
  - **IECTGTDVCIDX1[7:0] | IECTGTDVCIDX2[7:0]**

  That is, the highest voltage of all requests is selected.
- The logical AND of two levels results in the lower of the two levels. For example, in a 2-processor system, to ensure that the lower processor clock is selected before starting to reduce voltage the following calculation can be performed:
  - **IECTGTDCGIDX1[7:0]** & **IECTGTDCGIDX2[7:0]**

  That is, the lowest performance level of all requests is selected.

Because each bit can be synchronized safely and independently, this simple coding also supports efficient clock selection.

2.2.4 Maximum performance request IECMAXPERF signal

The IEC provides a mechanism for the hardware to respond to critical and unpredicted high-priority events. This is through a maskable **IECMAXPERF** input. It provides a hardware mechanism to override any software programmed performance level. You can assert this signal to force the IEC to request maximum performance from the DCG, DVC. This results in the maximum frequency and voltage in the shortest time.

This signal is level sensitive and must be de-asserted when the critical requirements have been met. The signal is treated as an asynchronous input and is internally synchronized.

A simple usage of this could be to force maximum performance to service interrupts or scheduled real-time requirements.

Figure 2-5 shows how the IEC requests maximum performance when **IECMAXPERF** is asserted.

Note

- when **IECMAXPERF** is cleared, the IEC then requests the previously programmed software performance level.
- because of the time taken for voltage to ramp, and the incremental performance increases (see *Performance requirement optimization* on page 2-22), it is possible that the condition causing maximum performance request, might be over before `Vmax` is reached.
- the **IECMAXPERF** input can be used by the system to wake-up the processor when it is in 0% performance level state.

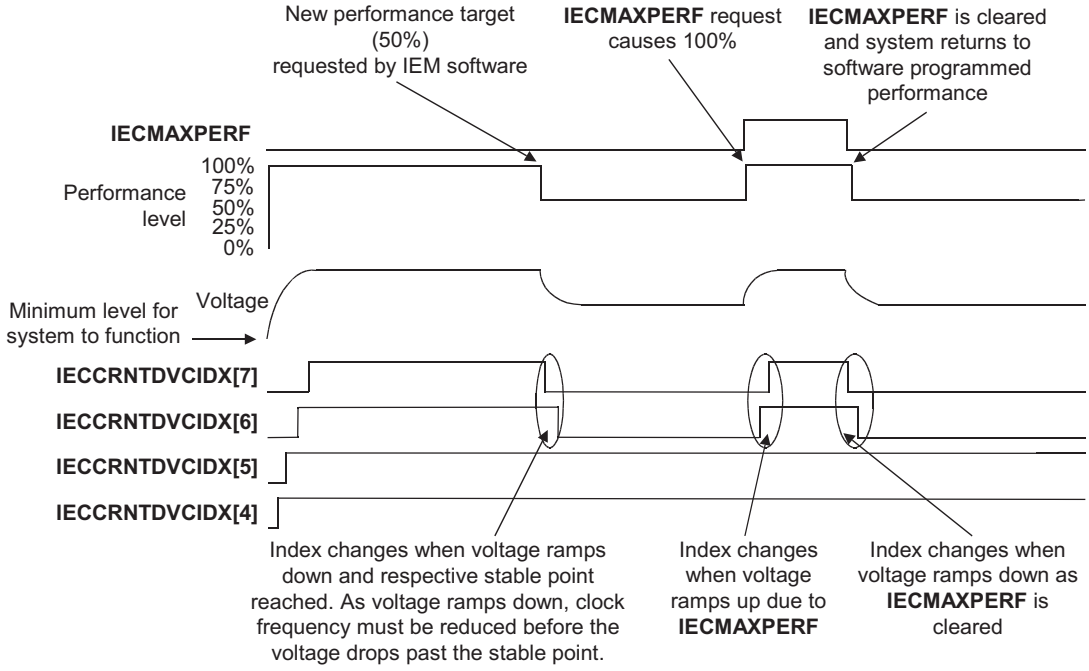


Figure 2-5 Requesting maximum performance when IECMAXPERF is asserted

Simple example usage of IECMAXPERF

In a SoC, you can use a number of different conditions to trigger the maximum performance request. The simplest usage of this is to request the maximum performance whenever an interrupt request is received. That is, **IECMAXPERF** could be tied to the output of the SoC interrupt controller.

For example, if the SoC has a PrimeCell *Vector Interrupt Controller* (VIC), then you can tie a logical OR of the inverse of the **nIRQ** and **nFIQ** outputs from the VIC to the **IECMAXPERF** input of the IEC. The inversion is required because the **IECMAXPERF** input is active HIGH.

Note

You must also connect the **nIRQ** and **nFIQ** outputs from the PrimeCell VIC to the **nIRQ** and **nFIQ** inputs of the ARM processor.

2.2.5 Synchronization mode handshaking signals

The IEC provides support for synchronization mode handshaking with synchronous and asynchronous bridges used in an asynchronous design based around the AMBA specification.When the system is running at the 100% performance level, the master and slave clocks to the synchronous and asynchronous bridges should be synchronous for optimal performance. At all other times the master and slave clocks are asynchronous because of unpredictable clock tree latencies in these performance levels.When the system and processor clocks are synchronous, the synchronous and asynchronous bridges can bypass synchronization logic to reduce the transfer latency. To do this, a handshaking mechanism is used to control entry to and exit from synchronous mode.

The request and acknowledge assertion and deassertion are shown in Figure 2-6.

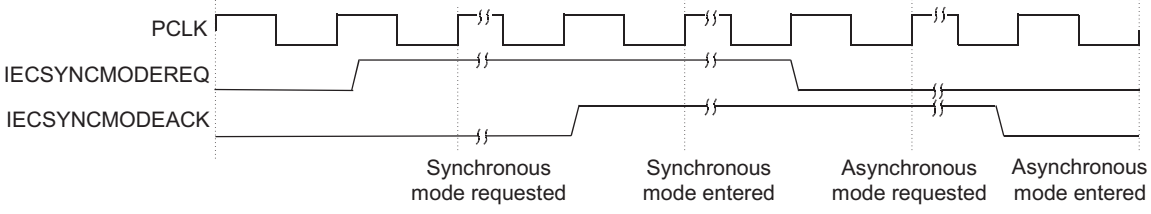


Figure 2-6 IEC and Synchronous/Asynchronous handshaking timing

When the synchronous and asynchronous bridges start to switch modes, it must be able to complete the change. That is, when the IEC has driven the **IEMSYNCMODEREQ** output to a new value it does not change again until the **IECSYNCMODEACK** input indicates the synchronous and asynchronous bridges have completed the mode change.

For example, if during a transition to synchronous mode the performance level is reduced from 100%, the IEC with the synchronous and asynchronous bridges complete the change to synchronous mode before starting to change back to asynchronous mode.

Any transitions between performance levels lower than 100% do not require interaction with the synchronous and asynchronous bridges, because it remains in asynchronous mode. The handshaking is only required for entry to or exit from the 100% performance level.

You can connect up multiple bridges at the same time. The request from IEC to bridges can be connected directly to all bridges, but acknowledges from bridges must be ANDed together before being passed to the IEC, ensuring that the IEC does not change mode until all bridges have changed mode.

Note

Synchronizer handshaking is not expected to be required in a DVS emulation system, as only a single clock frequency is used by the processor, rather than multiple synchronous and asynchronous frequencies. Therefore, synchronous and asynchronous bridge handshaking must be disabled when DVS emulation is used.

Asynchronous to synchronous mode

The following steps are performed when the system is running in asynchronous mode and the 100% performance level is requested:

1. The software programs the performance level to 100%.
2. The IEC drives the DVC and DCG target levels to 100%.
3. When the voltage has reached the maximum level, the processor clock is set to the maximum frequency and the DCG indicates the current performance level is 100%. The DCG also ensures that the system and processor clocks are synchronous.
4. The IEC asserts **IECSYNCMODEREQ** now that the processor can be run synchronized with the rest of the system.
5. The synchronous and asynchronous bridge flushes its FIFOs, switches into synchronous mode and asserts **IECSYNCMODEACK**.

Synchronous to asynchronous

The following steps are performed when the system is running at the 100% performance level in synchronous mode and a reduction in performance is requested:

1. The software programs the performance level less than100%.
2. The IEC deasserts **IECSYNCMODEREQ**.

3. The synchronous and asynchronous bridge switches into asynchronous mode and deasserts **IECSYNCMODEACK**.
4. The IEC drives the DVC and DCG target levels to the new value.
5. The DCG generates the slower asynchronous clock for the processor, and the DVC requests a lower voltage from the PSU after the DCG has reduced the clock frequency.

Note

If you do not use this handshaking then you must tie the **IECSYNCMODEACK** input HIGH.

2.2.6 Maskable interrupts

There are two maskable interrupts generated by the IEC. These individual interrupts are ORed together to form a third combined interrupt output. Provision of individual interrupts, in addition to a combined interrupt output enables you to use either a global interrupt service routine, or modular device drivers to handle interrupts.

You can enable or disable the individual interrupts by changing the mask bits in the IECIMSC Register. Setting the appropriate bit LOW clears the mask and enables the interrupt. The status of the individual interrupts can be read either from the IECRIS Register, for raw interrupt status, or from the IECMIS Register, for the masked interrupt status. The following subsections describe the interrupt signals:

- *Sleep mode*
- *Wake-up mode* on page 2-29
- *Combined interrupt* on page 2-29.

Sleep mode

The **IECCPUSLPINT** signal is asserted by the IEC to request the processor to enter a Sleep mode. When the processor sees this interrupt, it is expected to:

1. save state as appropriate
2. flush its write buffers
3. clear the **IECCPUSLPINT** interrupt
4. execute the *WaitForInterrupt* (WFI) command when the **IECCPUSLPINT** output from the IEC has been cleared, this causes the **STANDBYWFI** output from the processor to be asserted.

Note

The **STANDBYWFI** output from the ARM processor must be connected to the **IECCPUWFIACK** input of the IEC.

You can clear the interrupt by writing a 1 to the corresponding bit in the IECICR Register.

After a sleep interrupt has been generated, when the IEC detects an acknowledge on the **IECCPUWFIACK** input, it sets the target performance index outputs **IECTGTDVCIDX** and **IECTGTDCGIDX** to the minimum index level.

There are two conditions that cause this interrupt to be asserted by the IEC. These are:

- whenever the 0% performance level is requested by the software
- whenever the DVS emulation mode wants to de-assert the power request. See *DVS emulation with PWM* on page 2-31.

Wake-up mode

The **IECCPUWUINT** is asserted by the IEC to request the processor to wake from Sleep mode before it stops the clock. When the processor wakes-up, the interrupt is cleared by writing a 1 to the corresponding bit in the IECICR Register.

When the IEC asserts **IECCPUWUINT**, it also requests maximum performance level.

This interrupt is only asserted when the IEC is in the DVS emulation mode, see *DVS emulation with PWM* on page 2-31.

Combined interrupt

The interrupts are also combined into a single output signal **IECINT**, that is an OR function of the individual masked sources. You can connect this output to the system interrupt controller to provide another level of masking on an individual peripheral basis.

The combined IEC interrupt is asserted when any of the other two interrupts are asserted.

2.2.7 Acknowledgement of WaitForInterrupt signal

The **IECCPUWFIACK** input is used by the IEC as an acknowledgement that the processor has executed the WFI command. That is, it has entered sleep mode and ceased all activity on the bus. When a HIGH level on this input is seen, the IEC sets the target performance index outputs **IECTGTDVCIDX** and **IECTGTDCGIDX**, to the

minimum index level. The IEC does not request the clock to be stopped or the voltage to be lowered until the CPU has acknowledged that it has entered sleep mode by asserting the **IECCPUWFIACK** signal

2.2.8 DPM channels

The IEC provides hardware support for the IEM software to monitor the work done. Three channels are available. Each channel is in the form of a 64-bit accumulator that counts the fractional performance by incrementing the channel value with the current channel rate (0-32). The rate that each of the channels accumulates is described in the following sections.

- *Channel clocking*
- *Channel 1 accumulation rate* on page 2-31
- *Channels 2 and 3 accumulation rates* on page 2-31.

On reset all channels are frozen and held at their reset values.

Because the channels are 64-bit wide and the APB interface of the IEC is 32-bit, the software has to read each channel in a defined sequence. The low bits of the channel must be read first followed by the high bits of the channel. The IEC ensures that the high bits of the channel are stored when the low bits of the corresponding channel are read.

Channel clocking

The clocking for each of the channels is identical, and uses **PCLK**. However, each channel only increments on a rising edge of **PCLK** when the channel clock enable input, **IECDPMCLKEN** is HIGH. The design of the channels assumes that **IECDPMCLKEN** is synchronously derived from **PCLK**. This enables you to operate the channels at a lower effective frequency than **PCLK**. That is, **IECDPMCLKEN** is pulsed HIGH at the required frequency synchronous to **PCLK**. This relationship is shown in Figure 2-7.

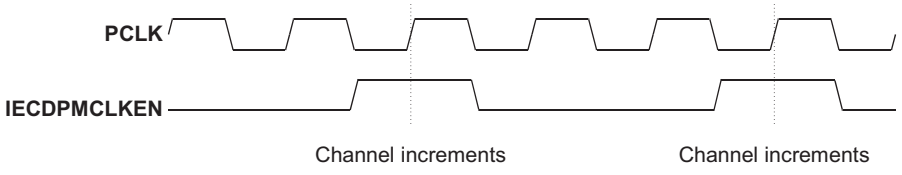


Figure 2-7 PCLK and IECDPMCLKEN relationship

Note

These notes are for additional information only and not connected to Figure 2-7 on page 2-30.

- **IECDPMCLKEN** can also be tied HIGH resulting in the channels being clocked at the **PCLK** frequency, and this might lead to a high power solution. In this case, the **IECCFGFREQDPM** must be tied to indicate the **PCLK** frequency.
- The minimum recommended frequency to pulse **IECDPMCLKEN** is 1MHz.
- The frequency of **PCLK** must always be faster than the frequency that **IECDPMCLKEN** is pulsed.

Channel 1 accumulation rate

The rate that Channel 1 accumulates is dependent on the fractional performance that the processor is currently running at. The rate is taken from the current DCG clock status **IECCRNTDCGIDX**. That is, the rate for channel 1 changes whenever a new fractional performance level is programmed and the DCG changes to a different clock frequency. That is, on every update cycle, the value in **IECCRNTDCGIDX[7:0]** is added to the Channel 1.

Channels 2 and 3 accumulation rates

The rates that channels 2 and 3 accumulate are software programmable. You can program the IECDPM2RATE and IECDPM3RATE Registers to set the rate that channels 2 and 3, respectively, accumulate. The IEM software can use this for a number of different functions depending on the monitoring algorithms that it is executing.

2.2.9 DVS emulation with PWM

The IEC has a mode of operation to enable support for systems that do not implement voltage scaling but still want to be able to do performance scaling. This enables reuse of the DVS interface abstraction to emulate multiple levels of performance on a system that only supports basic run-fast and idle modes of operation. The DVS emulation is performed through a PWM-style power request on the **IECPWRREQ** output. The PWM duty cycle for the output depends on the current performance target specified by the IEM software.

Note

In this mode of operation, the processor only executes at maximum 100% or minimum 0% performance levels. Consequently the only requirement of the DCG is to provide these performance levels for the processor.

This mode has a number of advantages and disadvantages as shown in Table 2-15.

Table 2-15 IEC DVS Emulation advantages and disadvantages

Advantages	Disadvantages
Simple SoC design-flow, Static Timing Analysis (STA) and verification at maximum performance	Energy savings less than run-slow/idle
Fits switched leakage management power domains	No forward progress until VDD fully ramped
Average power consumption is reduced	Processor executes at 100%-0% performance levels
-	PWM-style power request approach

The following subsections describe:

- *PWM frame period*
- *Frame duty cycle changes* on page 2-33
- *Example DVS emulation* on page 2-34
- *DVS emulation and IECMAXPERF assertion* on page 2-35.

PWM frame period

Each PWM frame is built from eight time slots. The time for each slot is programmable, and must be tuned by the software depending on the ramp and decay times for the voltage to go between maximum and minimum voltages. On reset, the time slots are set to 100µs. The length of each time slot is controlled by the value programmed in the IECDVSEMSTR Register and the **IECDVSEMCLKEN** input signal.

Figure 2-8 shows an example of three different PWM frames.

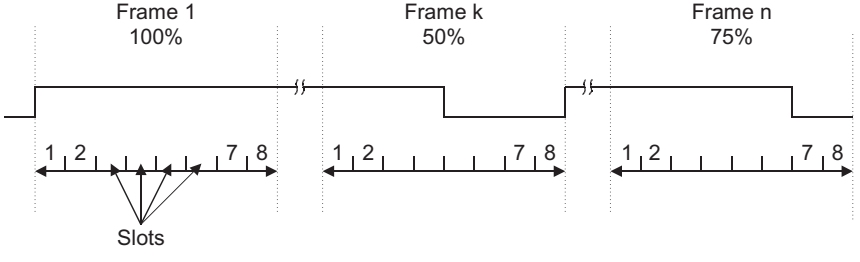


Figure 2-8 Example PWM frames

The sequence of events is as follows:

1. Frame 1 has a duty cycle of 100% because the performance requested at that time by the IEM software is 100%. All the slots in Frame 1 are HIGH, this is known as the mark period. None of the slots are LOW, this is known as the space period.
2. Frame k has a duty cycle of 50% because the performance requested at that time by the IEM software was 50%. In this case, half the slots of the frame are HIGH (the mark period) and half the slots are LOW (the space period).
3. Frame n has a duty cycle of 75% because the performance requested at that time by the IEM software was 75%. In this case, six slots of the frame are HIGH (the mark period) and two slots are LOW (the space period).

Note

Because eight slots are used in each frame, eight equally divided non-zero performance levels are supported in DVS emulation mode, each 12.5% higher than the previous level.

Frame duty cycle changes

The PWM duty cycle is changed when the IEM software programs a new performance level. At this point, the PWM frame is reset. Figure 2-9 shows an example of this happening.

In the figure you can see that frames 1 and 2 are both at 50%, the third frame starts at 50% as well but was not yet complete when the software programmed a new performance level (in slot3). The framing was reset at this point, and frame 4 started and the duty cycle was the new performance level of 75%. Frame 5 also had a 75% duty cycle.

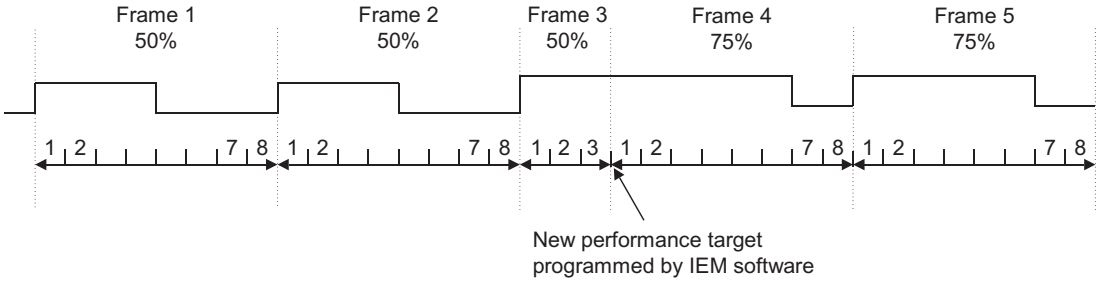


Figure 2-9 Duty cycles changes in frames

Example DVS emulation

Figure 2-10 on page 2-35 shows the sequence of events that occur when the software programs a new performance target and also how the IEC interacts with the processor to transition between sleep and wake-up. If for example the IEM software has requested a 50% performance level, in this situation, from Figure 2-10 on page 2-35, you can see that at the end of the mark period, the IEC asserts the **IECCPUSLPINT** interrupt.

When the processor sees this interrupt, it is expected to save state as appropriate, flush its write buffers, clear the **IECCPUSLPINT** interrupt and execute the WFI command when the **IECCPUSLPINT** output has been cleared. The execution of this command causes the **STANDBYWFI** output from the processor to be asserted.

The **STANDBYWFI** output from the ARM processor must be connected to the **IECCPUWFIACK** input of the IEC. When the IEC detects a high level on this input, it de-asserts the **IECPWRREQ** output.

Note

The **IECCPUSLPINT** interrupt can be cleared by writing a 1 to the corresponding bit in the IECICR Register.

At the end of the frame, the IEC asserts **IECCPUWUINT** to wake-up the processor and also requests power. The clock for the processor is not re-started by the DCG until a 1 is detected on **IECCRNTDVCIDX**, that is, until the power is available and stable. When the processor has woken up, **IECCPUWFIACK** is deasserted and it clears **IECCPUWUINT**.

Note

A new frame starts when the power is requested through the **IECPWRREQ** output. Therefore, some of the frame is used up while the voltage for the processor is ramping up to the maximum level.

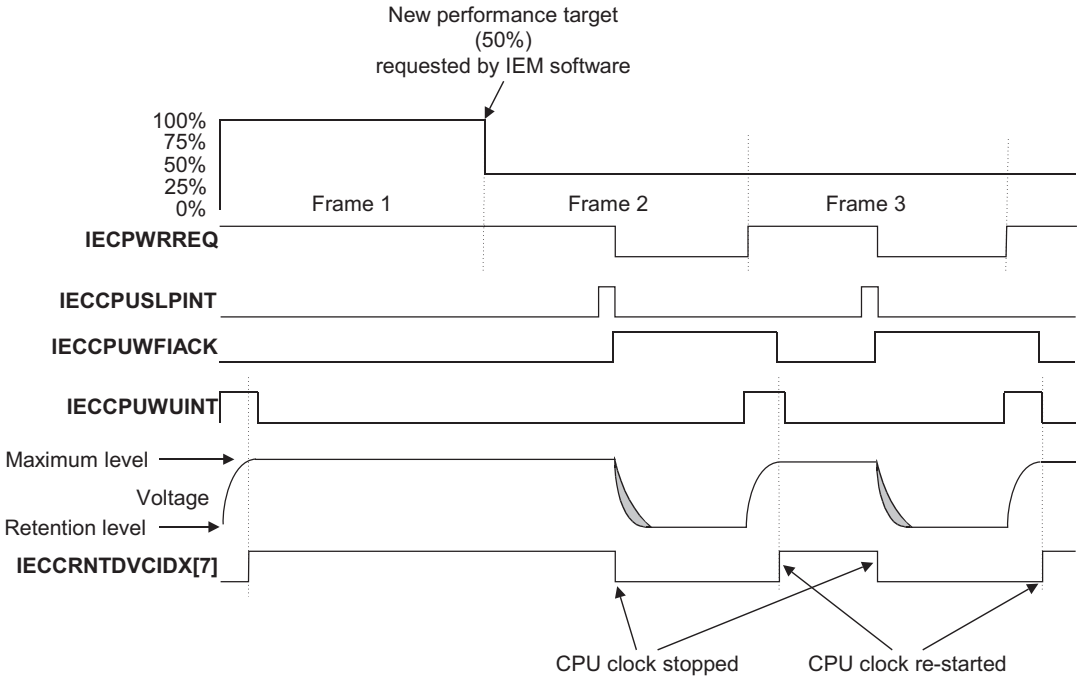


Figure 2-10 Simple example of DVS emulation

DVS emulation and IECMAXPERF assertion

Figure 2-11 on page 2-36 shows the sequence of events when a maximum performance condition is triggered and how the IEC behaves in this situation. This figure shows that when **IECMAXPERF** is asserted, the system is at 50% performance level but is in a sleep state. That is, the voltage to the processor is at the retention level, this is the minimum level to enable the system to function. The assertion of **IECMAXPERF** has the following effect:

- the current frame is terminated
- **IECPWRREQ** is asserted by the IEC
- the performance level is set to maximum
- **IECCPUWUINT** is asserted.

When the voltage reaches the full level **IECCRNTDVCIDX[7]** is 1, the processor clock is enabled by the DCG, the processor clears **IECCPUWUINT**, services the maximum performance condition and clears **IECMAXPERF** and carries on doing more work. When **IECMAXPERF** is cleared, the IEC starts a new frame at the previously programmed performance level of 50%.

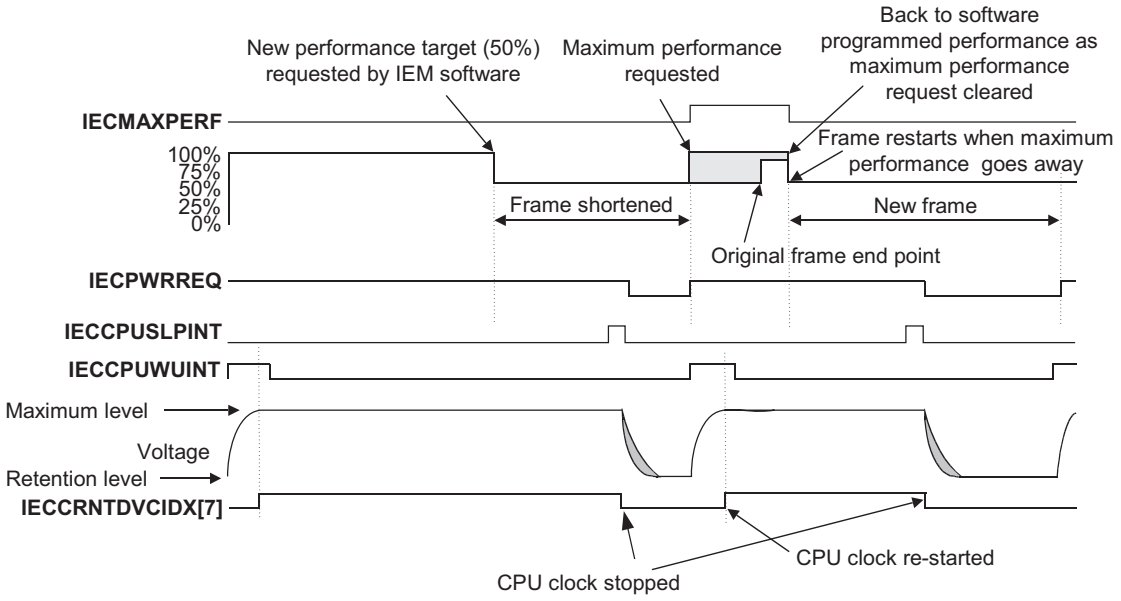


Figure 2-11 DVS emulation and IECMAXPERF

2.2.10 IEC and software development support

For software development and power measurement diagnostics support, a debug emulation mode is provided in the IEC. In the debug emulation mode, the **IECTGTDVCIDX** outputs are held at maximum value and only the **IECTGTDCGIDX** outputs are modulated. This feature enables you to debug your system and ensures that the OS programming model behaves correctly with full dynamic clock generation.

Note

The voltage settling times appear to be instantaneous when increasing target performance in this debug mode.

2.2.11 IEC and SoC DFT

When performing SoC DFT through scan based ATPG or other techniques, you might require the IEC outputs to be in a known static state. Consequently, a DFT interface is defined as part of the IEC specification. This DFT interface consists of two control signals and target performance index inputs. They are:

DFT control signal 1

The **IECDFTCLAMP** input is an active HIGH input and is used to indicate whether the target index outputs from the IEC, **IECTGTDVCIDX** and **IECTGTDCGIDX** must be held at static values. This feature can be used to ensure that during production test, the target index outputs from the IEC do not cause any unexpected state changes resulting in unpredictable test behavior.

The **IECTGTDVCIDX** and **IECTGTDCGIDX** outputs are held at maximum performance level, all 1s, when **IECDFTCLAMP** is asserted

The **IECTGTDVCIDX** and **IECTGTDCGIDX** outputs assume normal behavior when **IECDFTCLAMP** is not asserted.

DFT control signal 2

The **IECDFTPERFCNTL** input is an active HIGH input and it indicates whether the target index outputs are driven from the **IECDFTTGTDVCIDX** and **IECDFTTGTDCGIDX** inputs. Use this feature to ensure that the target performance index can be overridden easily by a hardware based DFT controller. This DFT controller is SoC specific and can use this feature to step through each of the performance levels in the DVC and the DCG.

The **IECTGTDVCIDX** and **IECTGTDCGIDX** outputs are driven with the same values of the **IECDFTTGTDVCIDX** and **IECDFTTGTDCGIDX** inputs respectively when **IECDFTPERFCNTL** is asserted.

The **IECTGTDVCIDX** and **IECTGTDCGIDX** outputs assume normal behavior when **IECDFTPERFCNTL** is not asserted.

IEC DFT target indexes

The **IECDFTTGTDVCIDX** and **IECDFTTGTDCGIDX** inputs use the same thermometer encoding as the **IECTGTDVCIDX** and **IECTGTDCGIDX** outputs. For more details about thermometer encoding, see *Target index and current index* on page 2-19.

The values presented on these inputs are only presented on the **IECTGTDVCIDX** and **IECTGTDCGIDX** outputs when **IECDFTPERFCNTL** is asserted.

Note

The two control signals **IECDFTCLAMP** and **IECDFPERCENTL** are mutually exclusive, they must not be asserted together.

Chapter 3
Programmer's Model

This chapter describes the registers of the IEC and provides details that you require if programming the device.

This section contains the following sections:

3.1 About the programmer's model

The following information is provided on register fields:

- The base address of the IEC is not fixed, but is determined by the APB address decoding and can be different for any particular system implementation.
- All reserved or unused address locations must not be accessed because this can result in unpredictable behavior of the device.
- All reserved or unused bits of writable registers must be written as zero, and ignored on read unless otherwise stated in the relevant text.
- All register bits are reset to logic 0 by a system reset unless otherwise stated in the relevant text.
- Unless otherwise stated in the relevant text, all registers support read and write accesses. A write updates the contents of the register and a read returns the contents of the register.
- All registers defined in this document can only be accessed using word reads and word writes, unless otherwise stated in the relevant text.

3.2 Summary of registers

Table 3-1 summarizes the IEC registers in base offset order.

Table 3-1 Summary of registers

Name	Base offset	Type	Reset nPOR	Description
IECDPCCR	0x000	R/W	0x1	See *DPC Control Register* on page 3-6
IECDVSEMSTR	0x004	R/W	0x63	See *DVS Emulation Slot Time Register* on page 3-7
IECDPCTGTPERF	0x008	WO	0x80	See *DPC Target Performance Register* on page 3-8
IECDPCCRNTPERF	0x00C	RO	System dependent	See *DPC Current Performance Register* on page 3-9
IECIMSC	0x010	R/W	0x3	See *Interrupt Mask Set and Clear Register* on page 3-9
IECRIS	0x014	RO	0x0	See *Raw Interrupt Status Register* on page 3-10
IECMIS	0x018	RO	0x0	See *Masked Interrupt Status Register* on page 3-11
IECICR	0x01C	WO	0x0	See *Interrupt Clear Register* on page 3-11
IECCFGCPUFREQ	0x020	RO	System dependent	See *Configured CPU Frequency Register* on page 3-12
IECDPMFREQ	0x024	RO	System dependent	See *DPM Frequency Register* on page 3-12
IECCFGDCGIDXMAP00	0x040	RO	System dependent	See *Configuration Fractional Index Map 00 Register* on page 3-13
IECCFGDCGIDXMAP32	0x044	RO	System dependent	See *Configuration Fractional Index Map 32 Register* on page 3-13
IECCFGDCGIDXMAP64	0x048	RO	System dependent	See *Configuration Fractional Index Map 64 Register* on page 3-14
IECCFGDVCIDXMAP	0x04C	RO	System dependent	See *Configuration DVC Index Map Register* on page 3-14
	0x050-0x05C	-	-	Reserved. read undefined, do not modify

Table 3-1 Summary of registers (continued)

Name	Base offset	Type	Reset nPOR	Description
IECCFGDCGPERFMAP0	0x060	RO	System dependent	See *Configuration Performance Map 0 Register* on page 3-15
IECCFGDCGPERFMAP4	0x064	RO	System dependent	See *Configuration Performance Map 4 Register* on page 3-15
	0x068-0x0FF	-	System dependent	Reserved. read undefined, do not modify
IECDPMCR	0x100	R/W	0x000	See *DPM Command Register* on page 3-15
	0x104	-	-	Reserved. read undefined, do not modify
IECDPM2RATE	0x108	R/W	0x80	See *DPM Channel 2 Rate Register* on page 3-17.
IECDPM3RATE	0x10C	R/W	0x80	See *DPM Channel 3 Rate Register* on page 3-17
	0x110-0x17F	-	-	Reserved, read undefined, do not modify
IECDPMILO	0x180	RO	0x00000000	See *DPM Channel 1 Low Register* on page 3-18
IECDPM1H1	0x184	RO	0x00000000	See *DPM Channel 1 High Register* on page 3-18
IECDPM2LO	0x188	RO	0x00000000	See *DPM Channel 2 Low Register* on page 3-19
IECDPM2HI	0x18C	RO	0x00000000	See *DPM Channel 2 High Register* on page 3-19
IECDPM3LO	0x190	RO	0x00000000	See *DPM Channel 3 Low Register* on page 3-20
IECDPM3HI	0x194	RO	0x00000000	See *DPM Channel 3 High Register* on page 3-20
	0x198-0xEFF	-	-	Reserved, read undefined, do not modify
Integration Test Registers	0xF00-0xF28			See *Test registers* on page 4-4
	0xF2C-0xFC8	-	-	Reserved, read undefined, do not modify
IECPeriphID4	0xFD0	RO	0x03	See *Peripheral Identification Register 4* on page 3-24

Table 3-1 Summary of registers (continued)

Name	Base offset	Type	Reset nPOR	Description
IECPeriphID5	0xFD4	RO	0x08	See *Peripheral Identification Register 5* on page 3-25
IECPeriphID6	0xFD8	RO	Reserved	See *Peripheral Identification Register 6* on page 3-25
IECPeriphID7	0xFDC	RO	Reserved	See *Peripheral Identification Register 7* on page 3-26
IECPeriphID0	0xFE0	RO	0x50	See *Peripheral Identification Register 0* on page 3-23
IECPeriphID1	0xFE4	RO	0x17	See *Peripheral Identification Register 1* on page 3-23
IECPeriphID2	0xFE8	RO	0x14	See *Peripheral Identification Register 2* on page 3-24
IECPeriphID3	0xFEC	RO	0x08	See *Peripheral Identification Register 3* on page 3-24
IECID0	0xFF0	RO	0x0D	See *IEC Identification Register 0* on page 3-27
IECID1	0xFF4	RO	0xF0	See *IEC Identification Register 1* on page 3-27
IECID2	0xFF8	RO	0x05	See *IEC Identification Register 2* on page 3-27
IECID3	0xFFC	RO	0xB1	See *IEC Identification Register 3* on page 3-28

3.3 Register descriptions

This section describes the IEC registers, with the exception of the test registers which are described in Chapter 4 *Programmer's Model for Test*. Table 3-1 on page 3-3 provides cross references to individual registers.

3.3.1 DPC Control Register

IECDPCCR is a read and write register. It enables software to control several different functions.

Figure 3-1 shows the register bit assignments.

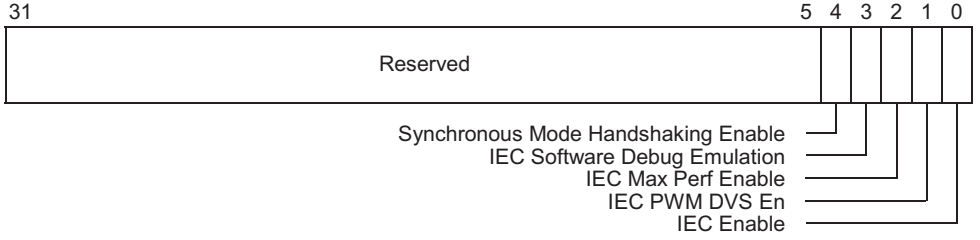


Figure 3-1 IECDPCCR Register bit assignments

Table 3-2 lists the register bit assignments.

Table 3-2 IECDPCCR Register bit assignments

Bits	Name	Description
[31:5]	Reserved	Reserved, read undefined, do not modify.
[4]	Synchronous Mode Handshaking Enable	Enable/disable the use of the synchronous mode handshaking control signals. Write 1 to enable. Write 0 to disable, also the reset value. When this bit is set, the synchronous mode handshaking signals are used to control entry and exit from the maximum performance mode. When this bit is cleared, the handshaking signals are not used.
[3]	IEC Software Debug Emulation	Control to debug performance scaling. Write 1 to enable. Write 0 to disable, also the reset value. When this bit is set[a], the performance level driven out of the **IECTGTDVCIDX** is set to maximum regardless of the software request. The performance level changes are only visible on **IECTGTDCGIDX**

Table 3-2 IECDPCCR Register bit assignments (continued)

Bits	Name	Description
[2]	IEC Max Perf Enable	Enable/disable maximum performance mode override Write 1 to enable. Write 0 to disable, also the reset value. When this bit is set, the maximum performance mode is enabled and therefore whenever **IECMAXPERF** goes high, the IEC requests maximum performance level regardless of the current software request.
[1]	IEC PWM DVS En	Enable/disable the IEC PWM DVS mode. Write 1 to enable. Write 0 to disable, also the reset value. When this bit is set, the IEC requests power through the **IECPWRREQ** output. The target performance index outputs are set to either maximum or minimum depending on the PWM state.
[0]	IEC Enable	Controls enabling and disabling of the IEC. Write 1 to enable, also the reset value. Write 0 to disable. When this bit is set, the IEC is enabled for performance scaling. When the bit is cleared, the IEC always requests maximum performance.

a. You must not use this during normal operation. You must only use this for software debug.

3.3.2 DVS Emulation Slot Time Register

IECDVSEMSTR is a read and write register. It enables software to program the length of each time slot in the PWM frame.

Figure 3-2 shows the register bit assignments.

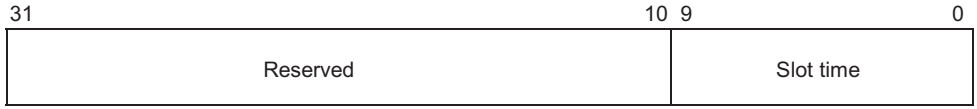


Figure 3-2 IECDVSEMSTR Register bit assignments

Table 3-3 lists the register bit assignments.

Table 3-3 IECDVSEMSTR Register bit assignments

Bits	Name	Description
[31:10]	Reserved	Reserved, read undefined, do not modify.
[9:0]	Slot time	The time in µs for each slot of a PWM frame. This is reset to `0x63`. For example, if you want each time slot to be 100µs in length, then the slot time bits must be programmed with `0x63`. Similarly, if you want each time slot to be 200µs in length, then the slot time bits must be programmed with `0xC7`[a].

a. This register must only be changed when DVS emulation mode is disabled. If written to during DVS emulation mode, it can cause unpredictable behavior.

Note

The slot time relies on the **IECDVSEMCLKEN** input signal being pulsed at a frequency of 1 MHz.

3.3.3 DPC Target Performance Register

IECDPCTGTPERF is a write-only register. A write to this register sets the new fractional performance level, that is, a fraction of the maximum performance level.

Figure 3-3 shows the register bit assignments.

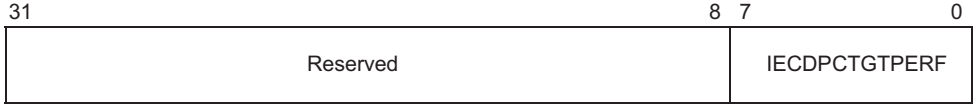


Figure 3-3 IECDPCTGTPERF Register bit assignments

Table 3-4 lists the register bit assignments.

Table 3-4 IECDPCTGTPERF Register bit assignments

Bits	Name	Description
[31:8]	Reserved	Reserved, read undefined, do not modify.
[7:0]	IECDPCTGTPERF	Sets the target fractional performance level. At system reset, the value `0x80` (100%).

3.3.4 DPC Current Performance Register

IECDPCCRNTPERF is a read only register. On a read, it returns the current fractional performance level of the system, that is, the decoded state of the **IECCRNTDCGIDX** input.

Figure 3-4 shows the register bit assignments.

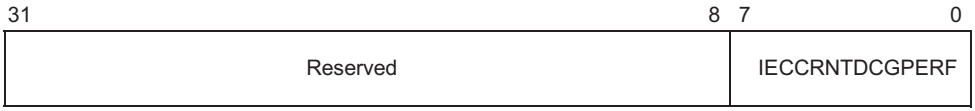


Figure 3-4 IECDPCCRNTPERF Register bit assignments

Table 3-5 shows the bit assignments for this register.

Table 3-5 IECDPCCRNTPERF Register bit assignments

Bits	Name	Description
[31:8]	Reserved	Reserved, read undefined, do not modify.
[7:0]	IECCRNTDCGPERF	Returns the current performance level as indicated to the IEC by the DCG on the **IECCRNTDCGIDX** inputs.

3.3.5 Interrupt Mask Set and Clear Register

IECIMSC is a read and write register. On a read, it returns the current state of the mask on the relevant interrupt. On a write of 1 to a particular bit, it sets the corresponding mask of that interrupt. A write of `0` clears the mask.

Figure 3-5 shows the register bit assignments.

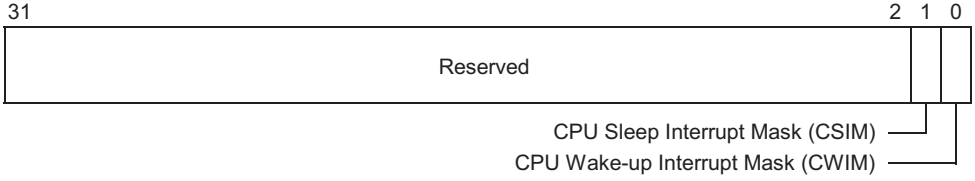


Figure 3-5 IECIMSC Register bit assignments

Table 3-6 lists the register bit assignments.

Table 3-6 IECIMSC Register bit assignments

Bits	Name	Description
[31:2]	Reserved	Reserved, read undefined, do not modify.
[1]	*CPU Sleep Interrupt Mask* (CSIM)	On a read, the current mask of the CSIM is returned. On a write of 1, the mask of CSIM interrupt is set. A write of 0 clears the mask. The reset value is 1.
[0]	*CPU Wake-up Interrupt Mask* (CWIM)	On a read, the current mask of the CWIM is returned. On a write of 1, the mask of CWIM interrupt is set. A write of 0 clears the mask. The reset value is 1.

3.3.6 Raw Interrupt Status Register

IECRIS is a read only register. On a read, it returns the current raw interrupt status of the corresponding interrupt.

Figure 3-6 shows the register bit assignments.

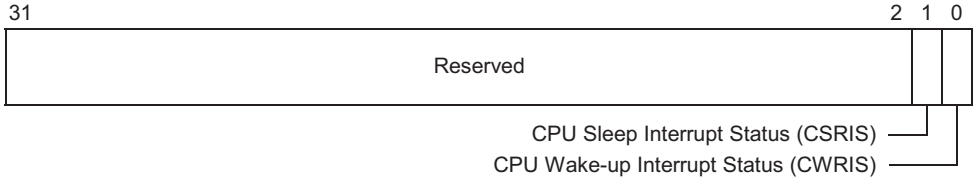


Figure 3-6 IECRIS Register bit assignments

Table 3-7 lists the register bit assignments.

Table 3-7 IECRIS Register bit assignments

Bits	Name	Description
[31:2]	Reserved	Reserved, read undefined, do not modify.
[1]	*CPU Sleep Interrupt Status* (CSRIS)	Returns the raw interrupt state prior to masking of the **IECCPUSLPINT** interrupt. The reset value is 0.
[0]	*CPU Wake-up Interrupt Status* (CWRIS)	Returns the raw interrupt state prior to masking of the **IECCPUWUINT** interrupt. The reset value is 0.

3.3.7 Masked Interrupt Status Register

IECMIS is a read-only register. On a read, it returns the current masked interrupt status of the corresponding interrupt.

Figure 3-7 shows the register bit assignments.

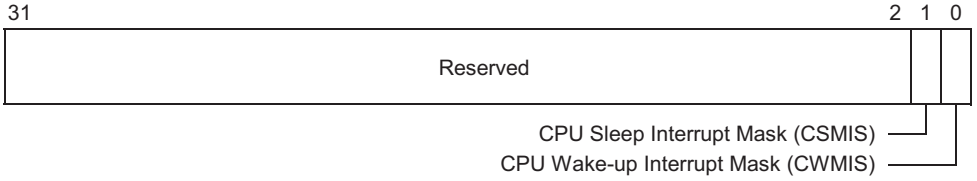


Figure 3-7 IECMIS Register bit assignments

Table 3-8 lists the register bit assignments.

Table 3-8 IECMIS Register bit assignments

Bits	Name	Description
[31:2]	Reserved	Reserved, read undefined, do not modify.
[1]	*CPU Sleep Masked Interrupt Status* (CSMIS)	Gives the masked interrupt state (after masking) of the **IECCPUSLPINT** interrupt. The reset value is `0`.
[0]	*CPU Wake-up Masked Interrupt Status* (CWMIS)	Gives the masked interrupt state (after masking) of the **IECCPUWUINT** interrupt. The reset value is `0`.

3.3.8 Interrupt Clear Register

IECICR is a write-only register. On a write of 1, the corresponding interrupt is cleared. A write of `0` has no effect.

Figure 3-8 shows the bit assignments for this register.

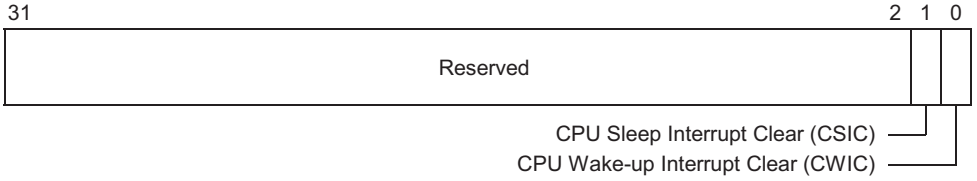


Figure 3-8 IECICR Register bit assignments

Table 3-9 lists the register bit assignments.

Table 3-9 IECICR Register bit assignments

Bits	Name	Description
[31:2]	Reserved	Reserved, read undefined, do not modify
[1]	*CPU Sleep Interrupt Clear* (CSIC)	Clears the **IECCPUSLPINT** interrupt. The reset value is `0`.
[0]	*CPU Wake-up Interrupt Clear* (CWIC)	Clears the **IECCPUWUINT** interrupt. The reset value is `0`.

3.3.9 Configured CPU Frequency Register

IECCFGCPUFREQ is a read-only register and returns the frequency, in kHz, as indicated by the **IECCFGCPUFREQ** inputs of the IEC.

Figure 3-9 shows the bit assignments for this register.

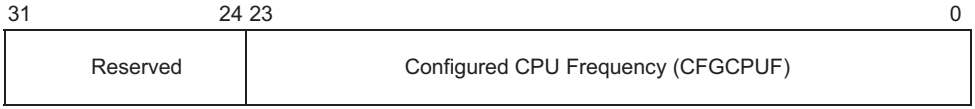


Figure 3-9 IECCFGCPUFREQ Register bit assignments

Table 3-10 lists the register bit assignments.

Table 3-10 IECCFGCPUFREQ Register bit assignments

Bits	Name	Description
[31:24]	Reserved	Reserved, read undefined, do not modify.
[23:0]	*Configured CPU Frequency* (CFGCPUF)	The configured CPU frequency in kHz.

3.3.10 DPM Frequency Register

IECDPMFREQ is a read-only register and returns the frequency that the **IECDPMCLKEN** input of the IEC is being clocked at in kHz. This frequency is as indicated by the **IECCFGFREQDPM** inputs.

Figure 3-10 on page 3-13 lists the register bit assignments.

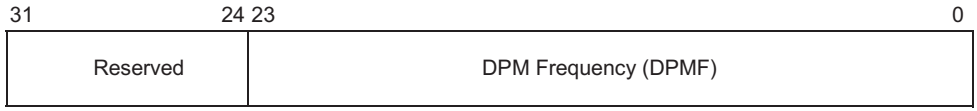


Figure 3-10 IECDPMFREQ Register bit assignments

Table 3-11 shows the bit assignments for this register.

Table 3-11 IECDPMFREQ Register bit assignments

Bits	Name	Description
[31:24]	Reserved	Reserved, read undefined, do not modify.
[23:0]	*DPM Frequency* (DPMF)	The DPM frequency in kHz.

3.3.11 Configuration Fractional Index Map 00 Register

IECCFGDCGIDXMAP00 is a read-only register and is provided for diagnostic support. It is a read-only register and is provided for diagnostic support. When read it returns the state of the **IECCFGDCGIDXMAP[31:0]** inputs. This information is used to determine the index level in the DCG that maps to the corresponding fractional performance level.

Table 3-12 shows the bit assignments for this register.

Table 3-12 IECCFGDCGIDXMAP00 Register bit assignments

Bits	Name	Description
[31:0]	IECCFGDCGIDXMAP00	State of **IECCFGDCGIDXMAP [31:0]**

3.3.12 Configuration Fractional Index Map 32 Register

IECCFGDCGIDXMAP32 is a read-only register and is provided for diagnostic support. When read it returns the state of the **IECCFGDCGIDXMAP [63:32]** inputs. This information is used to determine the index level in the DCG that maps to the corresponding fractional performance level.

Table 3-13 shows details of the bit assignments for this register.

Table 3-13 IECCFGDCGIDXMAP32 Register bit assignments

Bits	Name	Description
[31:0]	IECCFGDCGIDXMAP32	State of **IECCFGDCGIDXMAP [63:32]**

3.3.13 Configuration Fractional Index Map 64 Register

IECCFGDCGIDXMAP64 is a read-only register and is provided for diagnostic support. When read it returns the state of the **IECCFGDCGIDXMAP [95:64]** inputs. This information is used to determine the index level in the DCG that maps to the corresponding fractional performance level.

Table 3-14 shows the bit assignments for this register.

Table 3-14 IECCFGDCGIDXMAP64 Register bit assignments

Bits	Name	Description
[31:0]	IECCFGDCGIDXMAP64	State of **IECCFGDCGIDXMAP [95:64]**

3.3.14 Configuration DVC Index Map Register

IECCFGDVCIDXMAP is a read-only register and is provided for diagnostic support. When read it returns the state of the **IECCFGDVCIDXMAP[23:0]** inputs. This information is used to determine the relationship between the DVC index levels and the corresponding DCG index levels.

Figure 3-11 shows the bit assignments for this register.

Figure 3-11 IECCFGDVCIDXMAP Register bit assignments

Table 3-15 lists the register bit assignments.

Table 3-15 IECCFGDVCIDXMAP Register bit assignments

Bits	Name	Description
[31:24]	Reserved	Reserved, read undefined, do not modify.
[23:0]	IECCFGDVCIDXMAP	State of **IECCFGDVCIDXMAP [23:0]**

3.3.15 Configuration Performance Map 0 Register

IECCFCDGPERFMAP0 is a read-only register and is provided for diagnostic support. When read it returns the state of the **IECCFGDCGPERFMAP[31:0]** inputs. This information is used to determine the quantized fractional performance for each corresponding index level supported by the DCG.

Table 3-16 shows the bit assignments for this register.

Table 3-16 IECCFGDCGPERFMAP0 Register bit assignments

Bits	Name	Description
[31:0]	IECCFGDCGPERFMAP0	State of **IECCFGDCGPERFMAP** [31:0]

3.3.16 Configuration Performance Map 4 Register

IECCFGDCGPERFMAP4 is a read-only register and is provided for diagnostic support. When read it returns the state of the **IECCFGDCGPERFMAP[63:32]** inputs. This information is used to determine the quantized fractional performance for each corresponding index level supported by the DCG.

Table 3-17 shows the bit assignments for this register.

Table 3-17 IECFGDCGPERFMAP4 Register bit assignments

Bits	Name	Description
[31:0]	IECCFGDCGPERFMAP4	State of **IECCFGDCGPERFMAP** [63:32]

3.3.17 DPM Command Register

IECDPMCR is a read and write register. A number of commands can be issued for each of the channels by the IEM software.

Figure 3-12 shows the bit assignments.

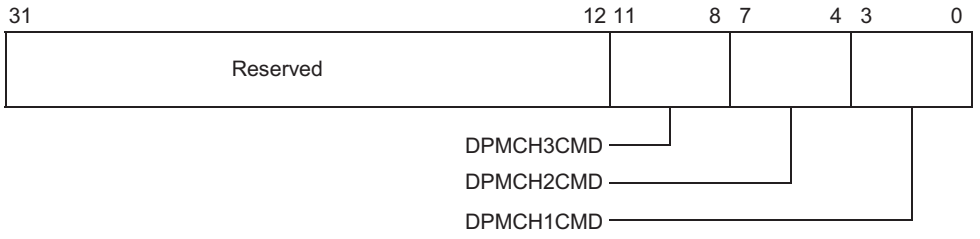


Figure 3-12 IECDPMCR Register bit assignments

Table 3-18 lists the register bit assignments.

Table 3-18 IECDPMCR Register bit assignments

Bits	Name	Description
[31:12]	Reserved	Reserved, read undefined, do not modify.
[11:8]	DPMCH3CMD	DPM Channel 3 command.
[7:4]	DPMCH2CMD	DPM Channel 2 command.
[3:0]	DPMCH1CMD	DPM Channel 1 command.

The commands and their encoding for each of the channels are shown in Table 3-19.

Table 3-19 DPMCHxCMD encoding bit assignments

DPMCHxCMD[a]	Command	Description
b'0000	Freeze	The channel is frozen and stops accumulating. This is also the reset value.
b'0001	Reset	The channel is reset to zero.
b'0010	Accumulate	The channel starts accumulating.
b'0011	Reserved	-
b'01xx[b]	Reserved	-
b'1xxx[b]	Reserved	-

a. The x indicates that the value applies to DPMCH1CMD, DPMCH2CMD, or DPMCH3CMD.
b. x is an ignored bit

3.3.18 DPM Channel 2 Rate Register

The IECDPM2RATE Register controls the rate that channel 2 of the DPM accumulates. It is a read and write register. The rate is an 8-bit fractional value, that is, it is a fraction of the maximum performance. The format of the rate is identical to the format of the IECDPCTGTPERF Register, that is, it has an inherent binary point.

Figure 3-13 shows the bit assignments for this register.

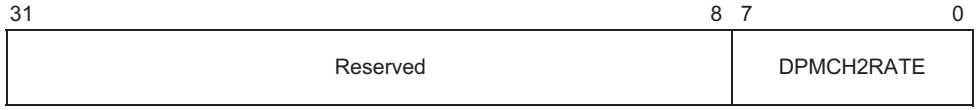


Figure 3-13 IECDPM2RATE Register bit assignments

Table 3-20 shows the bit assignments for this register.

Table 3-20 IECDPM2RATE Register bit assignments

Bits	Name	Description
[31:8]	Reserved	Reserved, read undefined, do not modify.
[7:0]	DPMCH2RATE	The fractional rate that DPM channel 2 counts. The reset value of this register is `0x80`, that is, 100%.

3.3.19 DPM Channel 3 Rate Register

The IECDPM3RATE Register controls the rate that channel 3 of the DPM accumulates. It is a read and write register. The rate is an 8-bit fractional value, that is, it is a fraction of the maximum performance. The format of the rate is identical to the format of the IECDPCTGTPERF Register, that is, it has an inherent binary point.

Figure 3-14 shows the bit assignments for this register.

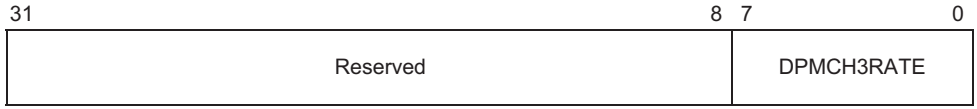


Figure 3-14 IECDPM3RATE Register bit assignments

Table 3-21 lists the register bit assignments.

Table 3-21 IECDPM3RATE Register bit assignments

Bits	Name	Description
[31:8]	Reserved	Reserved, read undefined, do not modify.
[7:0]	DPMCH3RATE	The fractional rate that DPM channel 3 counts. The reset value of this register is `0x80`, that is, 100%.

3.3.20 DPM Channel 1 Low Register

The IECDPM1LO Register enables you to read the current count in the low 32 bits of channel 1 of the DPM. It is a read-only register. When read it returns the current count of the low 32 bits of channel 1 of the DPM. The read also causes the high 32 bits of channel 1 to be captured. These high 32 bits are read through the IECDPM1HI Register.

Note

To read all 64 bits of channel 1 of the DPM, you must read the IECDPM1LO Register followed by a read of the IECDPM1HI Register. If you fail to follow this sequence then this results in an incorrect count for channel 1 when read.

Table 3-22 lists the register bit assignments.

Table 3-22 IECDPM1LO Register bit assignments

Bits	Name	Description
[31:0]	IECDPM1LO	Low 32-bits of DPM channel 1. The reset value is `0x00000000`.

3.3.21 DPM Channel 1 High Register

The IECDPM1HI Register enables you to read the current count in the high 32-bits of channel 1 of the DPM. It is a read only register. When read it returns the previously captured count of the high 32 bits of channel 1 of the DPM. This count is captured when you read the low 32 bits of channel 1.

Note

To read all 64 bits of channel 1 of the DPM, you must read the IECDPM1LO Register followed by a read of the IECDPM1HI Register. If you fail to follow this sequence then this results in an incorrect count for channel 1 when read.

Table 3-23 lists the register bit assignments.

Table 3-23 IECDPM1HI Register bit assignments

Bits	Name	Description
[31:0]	IECDPM1HI	High 32-bits of DPM channel 1. The reset value is `0x00000000`.

3.3.22 DPM Channel 2 Low Register

The IECDPM2LO Register enables you to read the current count in the low 32-bits of channel 2 of the DPM. It is a read-only register. When read it returns the current count of the low 32 bits of channel 2 of the DPM. The read also causes the high 32-bits of channel 2 to be captured. These high 32 bits are read through the IECDPM2HI Register.

Note

To read all 64-bits of channel 2 of the DPM, you must read the IECDPM2LO Register followed by a read of the IECDPM2HI Register. If you fail to follow this sequence then this results in an incorrect count for channel 2 when read.

Table 3-24 lists the register bit assignments.

Table 3-24 IECDPM2LO Register bit assignments

Bits	Name	Description
[31:0]	IECDPM2LO	Low 32-bits of DPM channel 2 The reset value is `0x00000000`.

3.3.23 DPM Channel 2 High Register

The IECDPM2HI Register enables you to read the current count in the high 32-bits of channel 2 of the DPM. It is a read-only register. When read it returns the previously captured count of the high 32 bits of channel 2 of the DPM. This count is captured when you read the low 32 bits of channel 2.

Note

To read all 64 bits of channel 2 of the DPM, you must read the IECDPM2LO Register followed by a read of the IECDPM2HI Register. If you fail to follow this sequence then this results in an incorrect count for channel 2 when read.

Table 3-25 lists the register bit assignments.

Table 3-25 IECDPM2HI Register bit assignments

Bits	Name	Type	Description
[31:0]	IECDPM2HI	Read	High 32-bits of DPM channel 2. The reset value is `0x00000000`.

3.3.24 DPM Channel 3 Low Register

The IECDPM3LO Register enables you to read the current count in the low 32 bits of channel 3 of the DPM. It is a read only register. When read it returns the current count of the low 32 bits of channel 3 of the DPM. The read also causes the high 32 bits of channel 3 to be captured. These high 32 bits are read through the IECDPM3HI Register.

Note

To read all 64 bits of channel 3 of the DPM, you must read the IECDPM3LO Register followed by a read of the IECDPM3HI Register. If you fail to follow this sequence then this results in an incorrect count for channel 3 when read.

Table 3-26 lists the register bit assignments.

Table 3-26 IECDPM3LO Register bit assignments

Bits	Name	Description
[31:0]	IECDPM3LO	Low 32-bits of DPM channel 3. The reset value is `0x00000000`.

3.3.25 DPM Channel 3 High Register

The IECDPM3HI Register enables you to read the current count in the high 32-bits of channel 3 of the DPM. It is a read only register. When read it returns the previously captured count of the high 32-bits of channel 3 of the DPM. This count is captured when you read the low 32-bits of channel 3.

Note

To read all 64-bits of channel 3 of the DPM, you must read the IECDPM3LO Register followed by a read of the IECDPM3HI Register. If you fail to follow this sequence then this results in an incorrect count for channel 3 when read.

Table 3-27 lists the register bit assignments.

Table 3-27 IECDPM3HI Register bit assignments

Bits	Name	Description
[31:0]	IECDPM3HI	High 32-bits of DPM channel 3. The reset value is `0x00000000`.

3.3.26 Peripheral Identification Registers

The Peripheral Identification Registers are eight, 8-bit read-only registers. They span two address locations:

- IECPeriphID0-3 Registers span address locations `0xFE0-0xFEC`
- IECPeriphID4-7 Registers span address locations `0xFD0-0xFDC`

Each of these blocks of registers 0-3 and 4-7 can conceptually be treated as one 32-bit read-only register. The IECPeriphID0-3 Registers provide the peripheral options listed in Table 3-28.

Table 3-28 Peripheral Identification Register options, IECPeriphID0-3

Bits	Description
Configuration 1[31:24]	The configuration option of the peripheral.
Revision number[23:20]	The revision number of the peripheral. The revision number starts from 0.
Designer[19:12]	The designer identification. ARM Limited is `0x41`, ASCII A.
Part number[11:0]	The peripheral, using the three digit product code. The IEC code is `0x750`.

Note

When you design a systems memory map then you must remember that the register has a 4KB-memory footprint. All memory accesses to the peripheral identification registers must be 32-bit, using the `LDR` and `STR` instructions.

Figure 3-15 on page 3-22 shows the bit assignments for the IECPeriphID0-3 Registers.

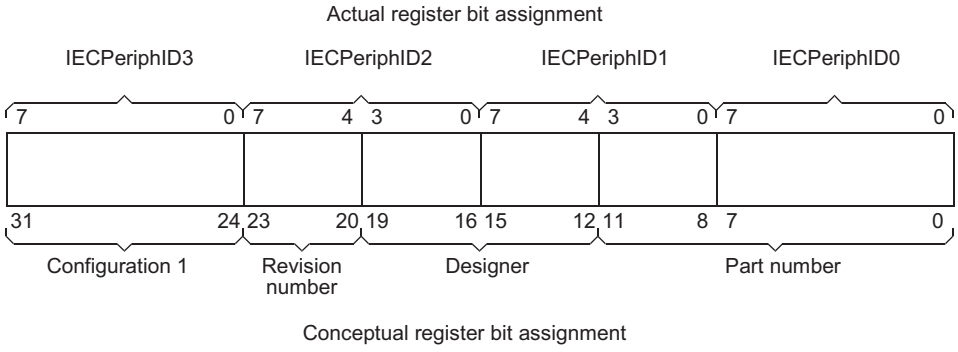


Figure 3-15 Peripheral Identification Register bit assignments, IECPeriphID0-3

The IECPeriphID4-7 Registers provide the peripheral options listed in Table 3-29.

Table 3-29 Peripheral Identification Register options, IECPeriphID4-7

Bits	Description
Configuration 5[31:24]	Reserved, read undefined
Configuration 4[23:16]	Reserved, read undefined
Configuration 3[15:8]	The configuration option of the peripheral
Configuration 2[7:0]	The configuration option of the peripheral

Figure 3-16 shows the bit assignments for the IECPeriphID4-7 Registers.

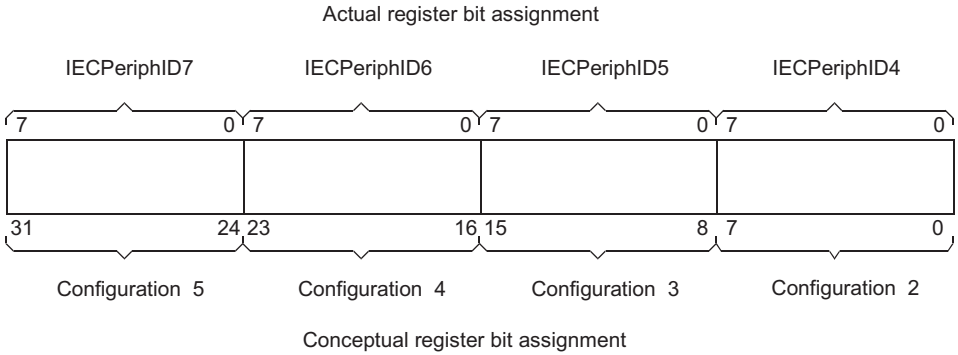


Figure 3-16 Peripheral Identification Register bit assignments, IECPeriphID4-7

The eight, 8-bit peripheral identification registers are described in the following subsections:

- *Peripheral Identification Register 0*
- *Peripheral Identification Register 1*
- *Peripheral Identification Register 2* on page 3-24
- *Peripheral Identification Register 3* on page 3-24
- *Peripheral Identification Register 4* on page 3-24
- *Peripheral Identification Register 5* on page 3-25
- *Peripheral Identification Register 6* on page 3-25
- *Peripheral Identification Register 7* on page 3-26.

Peripheral Identification Register 0

The IECPeriphID0 Register is read-only. It is hard coded and the fields in the register determine the reset value. Table 3-30 lists the register bit assignments.

Table 3-30 IECPeriphID0 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved, read undefined
[7:0]	Partnumber0	These bits read back as `0x50`

Peripheral Identification Register 1

The IECPeriphID1 Register is read-only. It is hard coded and the fields in the register determine the reset value. Table 3-31 lists the register bit assignments.

Table 3-31 IECPeriphID1 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved, read undefined
[7:4]	Designer0	These bits read back as `0x1`
[3:0]	Partnumber1	These bits read back as `0x7`

Peripheral Identification Register 2

The IECPeriphID2 Register is read-only. It is hard coded and the fields in the register determine the reset value. Table 3-32 lists the register bit assignments.

Table 3-32 IECPeripID2 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved, read undefined
[7:4]	Revision	These bits read back as `0x1`
[3:0]	Designer1	These bits read back as `0x4`

Peripheral Identification Register 3

The IECPeriphID3 Register is read-only. It is hard coded and the fields in the register determine the reset value. Table 3-33 lists the register bit assignments.

Table 3-33 IECPeriphID3 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved, read undefined.
[7:0]	Configuration1	Number of DPC levels. These bits read back as `0x08`.

The configuration field is used to indicate the Number of DPC levels. This is the maximum number of levels supported in the hardware.

Peripheral Identification Register 4

The IECPeriphID4 Register is read-only. It is hard coded and the fields in the register determine the reset value. Table 3-34 lists the register bit assignments.

Table 3-34 IECPeriphID4 Register bit assignments

Bit	Name	Description
[31:8]	-	Reserved, read undefined.
[7:3]	Reserved	
[2:0]	Configuration 2	Number of DPM channels. These bits read back as `0x3`.

The configuration field is used to indicate the number of DPM channels. This is the number of Dynamic Performance channels supported in the hardware. The control attributes for channels 1 to DPMCHANNELS (inclusive) are supported, all other channels are unimplemented and reserved.

Peripheral Identification Register 5

The IECPeriphID5 Register is read-only. It is hard coded and the fields in the register determine the reset value. Table 3-35 lists the register bit assignments

Table 3-35 IECPeriphID5 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved, read undefined.
[7:0]	Configuration 3	Number of DVS slots in a frame. These bits read back as `0x08`.

The configuration field is used to indicate the number of slots in a DVS emulation frame. This determines the number of supported performance levels available in the DVS emulation mode.

Peripheral Identification Register 6

The IECPeriphID6 Register is read-only. It is hard coded and the fields in the register determine the reset value. Table 3-36 lists the register bit assignments

Table 3-36 IECPeriphID6 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved, read undefined
[7:0]	Configuration 4	These bits are all reserved

Peripheral Identification Register 7

The IECPeriphID7 Register is read-only. It is hard coded and the fields in the register determine the reset value. Table 3-37 lists the register bit assignments

Table 3-37 IECPeriphID7 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved, read undefined
[7:0]	Configuration 5	These bits are all reserved

3.3.27 IEC Identification Registers

The IECID0-3 Registers are four, 8-bit read-only registers that span address locations `0xFF0-0xFFC`. The registers can conceptually be treated as one 32-bit register. These are used as a standard cross-peripheral identification system. The IECID Register is set to `0xB105F00D`. Figure 3-17 shows the bit assignments for the IECID-03 registers.

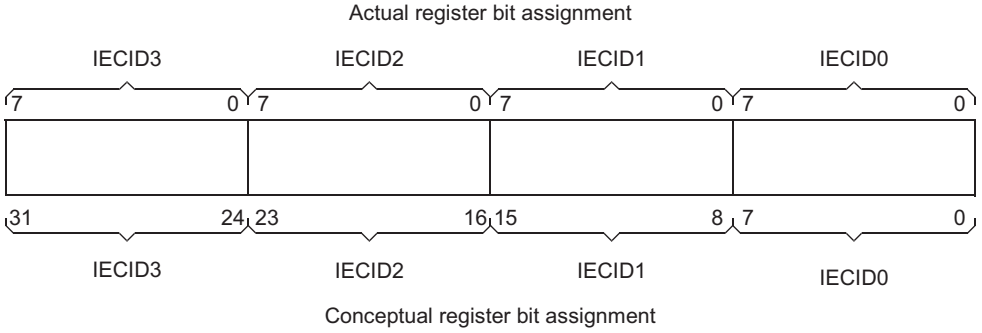


Figure 3-17 Identification Register bit assignments

Note

When you design a systems memory map then you must remember that the register has a 4KB-memory footprint. All memory accesses to the identification registers must be 32-bit, using the `LDR` and `STR` instructions.

The four, 8-bit IEC identification registers are described in:

- *IEC Identification Register 0* on page 3-27
- *IEC Identification Register 1* on page 3-27
- *IEC Identification Register 2* on page 3-27

- *IEC Identification Register 3* on page 3-28.

IEC Identification Register 0

The IECID0 Register is hard coded. The fields in the register determine the reset value. Table 3-38 lists the register bit assignments.

Table 3-38 IECID0 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved, read undefined
[7:0]	IECID0	These bits read back as `0x0D`

IEC Identification Register 1

The IECID1 Register is hard coded. The fields in the register determine the reset value. Table 3-39 lists the register bit assignments.

Table 3-39 IECID1 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved, read undefined
[7:0]	IECID1	These bits read back as `0xF0`

IEC Identification Register 2

The IECID2 Register is hard coded. The fields in the register determine the reset value. Table 3-40 lists the register bit assignments.

Table 3-40 IECID2 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved, read undefined
[7:0]	IECID2	These bits read back as `0x05`

IEC Identification Register 3

The IECID3 Register is hard coded. The fields in the register determine the reset value. Table 3-41 lists the register bit assignments.

Table 3-41 IECID3 Register bit assignments

Bits	Name	Description
[31:8]	-	
[7:0]	IECID3	These bits read back as `0xB1`

Chapter 4
Programmer's Model for Test

This chapter describes the additional logic for functional verification and production test. It contains the following sections:

4.1 IEC test harness overview

The additional logic for functional verification and integration vectors enables:

- capture of input signals to the block
- stimulation of the output signals.

The integration vectors provide a way of verifying that the IEC is correctly wired into a system. This is done by separately testing three groups of signals:

AMBA signals

These are tested by checking the connections of all the address and data bits.

Primary input and output signals

These are tested using a trickbox that can demonstrate the correct connection of the input and output signals to external pads.

Intra-chip signals (such as interrupt sources)

The tests for these signals are system-specific and enable you to write the necessary tests. Additional logic is implemented that enable you to read and write to each intra-chip input and output signal.

These test features are controlled by test registers. These enable you to test the IEC in isolation from the rest of the system using only transfers from the AMBA APB.

4.2 Scan testing

The IEC is designed to be compatible with scan insertion and the use of *Automatic Test Pattern Generation* (ATPG) techniques for manufacturing testing.

4.3 Test registers

The IEC test registers are memory mapped as shown in Table 4-1 summarizes the IEC test registers in base offset order.

Table 4-1 Integration test registers

Name	Base offset	Type	Reset nPOR	Description
IECITCR	0xF00	R/W	0x0	*Integration Test Control Register*.
	0xF04-0xF0F	-	-	Reserved, read undefined, do not modify.
IECITIP1	0xF10	R/W	0x00	*Integration Test Input Read or Set Register 1* on page 4-5.
IECITIP2	0xF14	R/W	0x0	*Integration Test Input Read or Set Register 2* on page 4-6.
IECITIP3	0xF18	R/W	0x00	*Integration Test Input Read or Set Register 3* on page 4-7.
	0xF14-0xF1F	-	0x00	Reserved, read undefined, do not modify.
IECITOP1	0xF20	R/W	0x0	*Integration Test Output Read or Set Register 1* on page 4-8.
IECITOP2	0xF24	R/W	0x00	*Integration Test Output Read or Set Register 2* on page 4-8.
IECITOP3	0xF28	R/W	0x00	*Integration Test Output Read or Set Register 3* on page 4-9.

Note

Test registers must not be accessed during normal mode of operation.

4.3.1 Integration Test Control Register

IECITCR is a read and write register. This general test register controls the operation of the IEC under test conditions. Figure 4-1 on page 4-5 shows the register bit assignments.

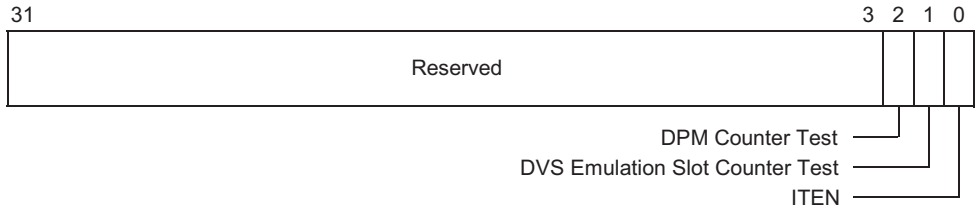


Figure 4-1 IECITCR Register bit assignments

Table 4-2 lists the register bit assignments.

Table 4-2 IECITCR Register bit assignments

Bit	Name	Description
[31:3]	-	Reserved. Unpredictable when read. Should be written as zero.
[2]	DPM Counter Test	Enable or disable test mode for all DPM counters. 0=DPM counter test mode disabled, also the reset value. 1=DPM counter test mode enabled. When this bit is set, the 64-bit DPM counters are split up into eight separate 8-bit counters, each accumulate by the CPU or programmed rate. This reduces the testing time required to ensure that all bits of the counters toggle correctly.
[1]	DVS Emulation Slot Counter Test	Enable or disable test mode for the bus V slotcounter. 0=DVS emulation slot counter test mode disabled, also reset value. 1=DVS emulation slot counter test mode enabled. When this bit is set, the 10-bit DVS emulation slot timing counter is split up into two 5-bit counters, each decrement separately. This reduces the testing time required to ensure that all bits of the counters toggle correctly.
[0]	ITEN	Integration test enable. When this bit is set to 1, the IEC is put into integration test mode. When 0, the IEC is in normal operating mode. The reset value is `0`.

4.3.2 Integration Test Input Read or Set Register 1

IECITIP1 is a read and write register. In integration test mode, it enables inputs to be both written to and read from.

Figure 4-2 on page 4-6 shows the bit assignments for the register.

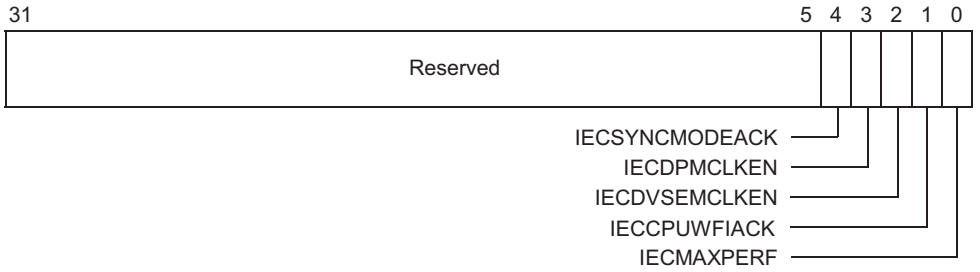


Figure 4-2 IECITIP1 Register 1 bit assignments

Table 4-3 lists the register bit assignments.

Table 4-3 IECITIP1 Register 1 bit assignments

Bit	Name	Description
[31:5]	-	Reserved. Unpredictable when read. Should be written as zero.
[4]	IECSYNCMODEACK	Intra-chip input. Writes to this bit, set the value to be driven onto the input **IECSYNCMODEACK**, in the integration test mode. Reads return the value of the **IECSYNCMODEACK** input at the output of the test multiplexer. The reset value is `0`.
[3]	IECDPMCLKEN	Intra-chip input. Writes to this bit, set the value to be driven onto the input **IECDPMCLKEN**, in the integration test mode. Reads return the value of the **IECDPMCLKEN** input at the output of the test multiplexer. The reset value is `0`.
[2]	IECDVSEMCLKEN	Intra-chip input. Writes to this bit set the value to be driven onto the input **IECDVSEMCLKEN**, in the integration test mode. Reads return the value of the **IECDVSEMCLKEN** input at the output of the test multiplexer. The reset value is `0`.
[1]	IECCPUWFIACK	Intra-chip input. Writes to this bit set the value to be driven onto the input **IECCPUWFIACK**, in the integration test mode. Reads return the value of the **IECCPUWFIACK** input at the output of the test multiplexer. The reset value is `0`.
[0]	IECMAXPERF	Intra-chip input. Writes to this bit set the value to be driven onto the input **IECMAXPERF,** in the integration test mode. Reads return the value of the **IECMAXPERF** input at the output of the test multiplexer. The reset value is `0`.

4.3.3 Integration Test Input Read or Set Register 2

IECITIP2 is a read and write register. In integration test mode, it enables inputs to be both written to and read from.

Figure 4-3 on page 4-7 shows the bit assignments for the register.

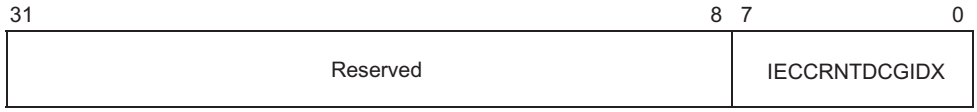


Figure 4-3 IECITIP2 Register 2 bit assignments

Table 4-4 lists the register bit assignments.

Table 4-4 IECITIP2 Register 2 bit assignments

Bit	Name	Description
[31:8]	-	Reserved, read undefined, do not modify.
[7:0]	IECCRNTDCGIDX	Intra-chip input. Writes to these bits set the value to be driven onto the inputs **IECCRNTDCGIDX[7:0]**, in the integration test mode. Reads return the value of the **IECCRNTDCGIDX[7:0]** inputs at the output of the test multiplexer. The reset value is `0x00`.

4.3.4 Integration Test Input Read or Set Register 3

IECITIP3 is a read and write register. In integration test mode, it enables inputs to be both written to and read from.

Figure 4-4 shows the bit assignments for the register.

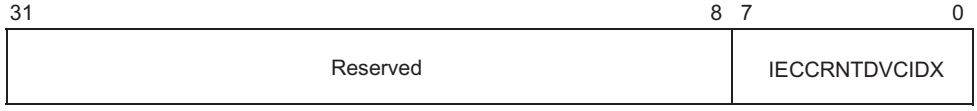


Figure 4-4 IECITIP3 Register 3 bit assignments

Table 4-5 lists the register bit assignments.

Table 4-5 IECITIP3 Register 3 bit assignments

Bit	Name	Function
[31:8]	-	Reserved, read undefined, do not modify.
[7:0]	IECCRNTDVCIDX	Intra-chip input. Writes to these bits set the value to be driven onto the inputs **IECCRNTDVCIDX[7:0]**, in the integration test mode. Reads return the value of the**IECCRNTDVCIDX[7:0]** inputs at the output of the test multiplexer. The reset value is `0x00`.

4.3.5 Integration Test Output Read or Set Register 1

IECITOP1 is a read and write register. In integration test mode, it enables outputs to be both written to and read from.

Figure 4-5 shows the bit assignments for the register.

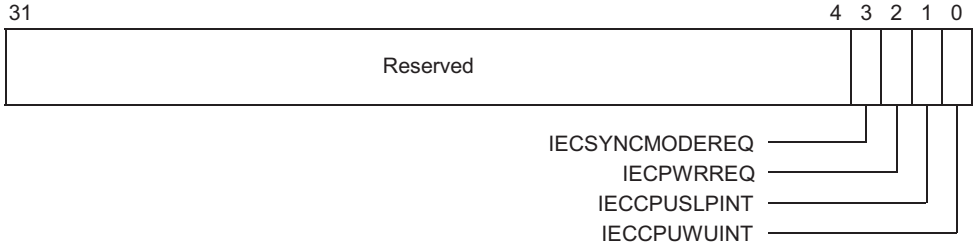


Figure 4-5 IECITOP1 Register 1 bit assignments

Table 4-6 lists the register bit assignments.

Table 4-6 IECITOP1 Register 1 bit assignments

Bit	Name	Description
[31:4]	-	Reserved, read undefined, do not modify.
[3]	IECSYNCMODEREQ	Intra-chip output. Writes to this bit set the value to be driven onto the **IECSYNCMODEREQ** output in integration test mode. Reads return the value of **IECSYNCMODEREQ** at the output of the test multiplexer. The reset value is `0`.
[2]	IECPWRREQ	Intra-chip output. Writes to this bit set the value to be driven onto the **IECPWRREQ** output in integration test mode. Reads return the value of **IECPWRREQ** at the output of the test multiplexer. The reset value is `0`.
[1]	IECCPUSLPINT	Intra-chip output. Writes to this bit set the value to be driven onto the **IECCPUSLPINT** output in integration test mode. Reads return the value of **IECCPUSLPINT** at the output of the test multiplexer. The reset value is `0`.
[0]	IECCPUWUINT	Intra-chip output. Writes to this bit set the value to be driven onto the **IECCPUWUINT** output in integration test mode. Reads return the value of **IECCPUWUINT** at the output of the test multiplexer. The reset value is `0`.

4.3.6 Integration Test Output Read or Set Register 2

IECITOP2 is a read and write register. In integration test mode, it enables outputs to be both written to and read from.

Figure 4-6 shows the bit assignments for the register.

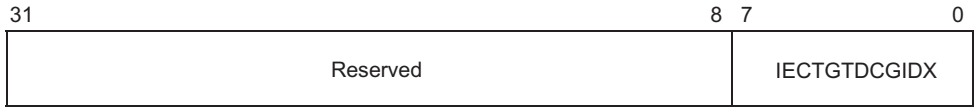


Figure 4-6 IECITOP2 Register 2 bit assignments

Table 4-7 lists the register bit assignments.

Table 4-7 IECITOP2 Register 2 bit assignments

Bit	Name	Description
[31:8]	-	Reserved, read undefined, do not modify.
[7:0]	IECTGTDCGIDX	Intra-chip outputs. Writes to these bits set the value to be driven onto the **IECTGTDCGIDX [7:0]** outputs in integration test mode. Reads return the value of **IECTGTDCGIDX[7:0]** at the output of the test multiplexer. The reset value is `0x00`.

4.3.7 Integration Test Output Read or Set Register 3

IECITOP3 is a read and write register. In integration test mode, it enables outputs to be both written to and read from.

Figure 4-7 shows the bit assignments for the register.

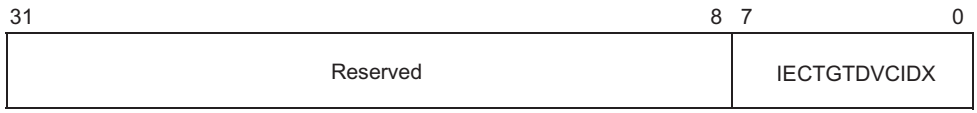


Figure 4-7 IECITOP3 Register 3 bit assignments

Table 4-8 lists the register bit assignments.

Table 4-8 IECITOP3 Register bit assignments

Bit	Name	Description
[31:8]	-	Reserved, read undefined, do not modify.
[7:0]	IECTGTDVCIDX	Intra-chip outputs. Writes to these bits set the value to be driven onto the **IECTGTDVCIDX[7:0]** outputs in integration test mode. Reads return the value of **IECTGTDVCIDX[7:0]** at the output of the test multiplexer. The reset value is `0x00`.

Appendix A
Signal Descriptions

This appendix describes the signals. It contains the following section:

- *IEC signals* on page A-2.

A.1 IEC signals

The IEC signals are describes in the following sections:

- *Clock and resets*
- *APB Slave interface*
- *Interrupts* on page A-3
- *Request and acknowledge* on page A-3
- *DVC control and status* on page A-4
- *DCG control and status* on page A-4
- *DPM channel enable* on page A-4
- *Configuration* on page A-5
- *DVS emulation* on page A-5
- *Synchronization mode handshaking* on page A-6
- *DFT interface* on page A-7.

A.1.1 Clock and resets

Table A-1 shows the clocks and reset signals.

Table A-1 Clock and reset signals

Signal	Type	Source or destination	Description
nPOR	Input	Reset generator	Power-on-reset input. This input can be asynchronously asserted but must be de-asserted synchronously to **PCLK**.
PCLK	Input	Clock generator	APB clock input

A.1.2 APB Slave interface

Table A-2 shows the APB slave interface signals.

Table A-2 APB interface signal

Signal	Type	Source or Destination	Description
PADDR[11:2]	Input	APB bridge	APB address bus
PENABLE	Input	APB bridge	APB enable
PRDATA[31:0]	Output	APB bridge	APB read data bus

Table A-2 APB interface signal (continued)

Signal	Type	Source or Destination	Description
PSEL	Input	APB bridge	APB peripheral select
PWDATA[11:0]	Input	APB bridge	APB write data bus
PWRITE	Input	APB bridge	APB read/write control

A.1.3 Interrupts

Table A-3 shows the IEC interrupt signals.

Table A-3 Interrupt signals

Signal	Type	Source or Destination	Description
IECCPUSLPINT	Output	Interrupt controller	CPU sleep interrupt
IECCPUWUINT	Output	Interrupt controller	CPU wake-up interrupt
IECINT	Output	Interrupt controller	Combined interrupt

A.1.4 Request and acknowledge

Table A-4 shows the request and acknowledge signals.

Table A-4 Request and acknowledge signals

Signal	Type	Source or Destination	Description
IECMAXPERF	Input	Interrupt controller	Request to go to maximum performance level
IECWFIACK	Input	CPU	Acknowledge indicating CPU has entered WFI mode. This signal must be connected to the **STANDBYWFI** output of an ARM core.

A.1.5 DVC control and status

Table A-5 shows the DVC control and status signals.

Table A-5 DVC control and status signals

Signal	Type	Source or destination	Description
IECCRNTDVCIDX[7:0]	Input	DVC	Current performance index of the DVC
IECTGTDVCIDX[7:0]	Output	DVC	Target performance index for the DVC

A.1.6 DCG control and status

Table A-6 shows the DCG control and status signals.

Table A-6 DCG control and status signals

Signal	Type	Source or destination	Description
IECCRNTDCGIDX[7:0]	Input	DCG	Current performance index of the DCG
IECTGTDCGIDX[7:0]	Output	DCG	Target performance index for the DCG

A.1.7 DPM channel enable

Table A-7 shows the DPM channel clock enable signal.

Table A-7 DPM channel enable signal

Signal	Type	Source or destination	Description
IECDPMCLKEN	Input	Clock generator	The enable for the DPM channel clock. All the DPM channels only increment on a rising edge of PCLK when **IECDPMCLKEN** is HIGH.

A.1.8 Configuration

Table A-8 shows the configuration interface signals.

Table A-8 Configuration interface signals

Signal	Type	Source or destination	Description
IECCFGDCGIDXMAP[95:0]	Input	Tie-off	The DCG index level corresponding to each performance level
IECCFGDCGPERFMAP[63:0]	Input	Tie-off	The DCG fractional performance for each index level
IECCFGDVCIDXMAP[23:0]	Input	Tie-off	The DVC index level corresponding to each performance level
IECCFGFREQCPU[23:0]	Input	Tie-off	Maximum frequency of CPU in kHz
IECCFGFREQDPM[23:0]	Input	Tie-off	Frequency that the DPM accumulators in the IEC are clocked in kHz

A.1.9 DVS emulation

Table A-9 shows the DVS emulation interface signals.

Table A-9 DVS emulation signals

Signal	Type	Source or destination	Description
IECDVSEMCLKEN	Input	Clock generator	The enable for advancing the PWM frame time slots when in DVS emulation mode. This signal must be pulsed at a frequency of 1MHz.
IECPWRREQ	Output	DVC	Request for maximum/minimum voltage when in DVS emulation mode.

A.1.10 Synchronization mode handshaking

Table A-10 shows the synchronization mode handshaking signals.

Table A-10 Synchronization mode handshaking signals

Signal	Type	Source or destination	Description
IECSYNCMODEACK	Input	Synchronous or asynchronous bridges	Acknowledge to show entry to or exit from synchronous mode has completed.
IECSYNCMODEREQ	Output	Synchronous or asynchronous bridges	Request for entry or exit from synchronous mode.

A.1.11 DFT interface

Table A-11 shows the DFT interface signals.

Table A-11 DFT interface signals

Signal	Type	Source or destination	Description
IECDFTCLAMP	Input	Test controller	This signal can be used during scan test of the SoC, to ensure that all target index outputs from the IEC are static. • When **IECDFTCLAMP** is 1: — **IECTGTDVCIDX[7:0]** and **IECTGTDCGIDX[7:0]** outputs are set to 100%, that is, set to all 1's. • When **IECDFTCLAMP** is 0: — **IECTGTDVCIDX[7:0]** and **IECTGTDCGIDX[7:0]** outputs are set to normal behavior.
IECDFTPERFCNTL	Input	Test controller	Performance control override for DFT. • When **IECDFTPERFCNTL** is 1: — **IECTGTDVCIDX[7:0]** and **IECTGTDCGIDX[7:0]** outputs are set to the values as presented on the **IECDFTTGTDVCIDX[7:0]** and **IECDFTTGTDCGIDX[7:0]** inputs respectively. • When **IECDFTPERFCNTL** is 0: — **IECTGTDVCIDX[7:0]** and **IECTGTDCGIDX[7:0]** outputs are set to normal behavior and the state of the **IECDFTTGTDVCIDX[7:0]** and **IECDFTTGTDCGIDX[7:0]** inputs is irrelevant.
IECDFTTGTDCGIDX[7:0]	Input	Test controller	Target index for DCG when **IECDFTPERFCNTL** is asserted. Uses the same encoding as the **IECTGTDCGIDX[7:0]** outputs.
IECDFTTGTDVCIDX[7:0]	Input	Test controller	Target index for DVC when **IECDFTPERFCNTL** is asserted. Uses the same encoding as the **IECTGTDVCIDX[7:0]** outputs.

Appendix B
Revisions

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

Table B-1 Differences between issue B and issue C

Change	Location
IECCRNTDCGIDX[7:0] signal changed direction	Figure 2-2 on page 2-10
Fractional performance map updated: Value b'00000xxx changed to b'000001xx Value b'0.0000xxx changed to b'0.00001xx	Table 2-1 on page 2-7
Disable and reset values for bit[4], bit[3] and bit[2] changed to 0	Table 3-2 on page 3-6

Glossary

This glossary describes some of the terms used in this manual. Where terms can have several meanings, the meaning presented here is intended.

Advanced Microcontroller Bus Architecture(AMBA)
AMBA is the ARM open standard for multi-master on-chip buses, capable of running with multiple masters and slaves. It is an on-chip bus specification that details a strategy for the interconnection and management of functional blocks that make up a *System-on-Chip* (SoC). It aids in the development of embedded processors with one or more CPUs or signal processors and multiple peripherals. AMBA complements a reusable design methodology by defining a common backbone for SoC modules. APB conforms to this standard.

Advanced Peripheral Bus (APB)
The AMBA Advanced Peripheral Bus is a simpler bus protocol than AHB. It is designed for use with ancillary or general-purpose peripherals such as timers, interrupt controllers, UARTs, and I/O ports. Connection to the main system bus is through a system-to-peripheral bus bridge that helps to reduce system power consumption.

AMBA *See* Advanced Microcontroller Bus Architecture.

APB *See* Advanced Peripheral Bus.

Advanced Power Controller (APC)
The NSC on-chip IP block that interfaces between the IEC, the HPM, the DCG and external power supplies conforming to the PowerWise™ Interface.

Architecture
The organization of hardware and/or software that characterizes a processor and its attached components, and enables devices with similar characteristics to be grouped together when describing their behavior, for example, Harvard architecture, instruction set architecture, ARMv6 architecture.

Central Processing Unit (CPU)
The part of a processor that contains the ALU, the registers, and the instruction decode logic and control circuitry. Also commonly known as the processor core.

CPU
See Central Processing Unit.

Dynamic Clock Generator (DCG)
A system-specific clock generator that has an IEC-compliant performance setting and monitoring interface. It also interfaces to the *Advanced Power Controller* (APC) and the *Hardware Performance Monitor* (HPM).

Dynamic Performance Controller (DPC)
A functional part of the IEC. The DPC translates the target performance level into index values for the DCG and DVC interfaces. It also performs reverse mapping when the IEM software reads the current performance level

Dynamic Performance Monitor (DPM)
A functional part of the IEC. The DPM provides hardware support for the IEM software to monitor the work done for various tasks. Three channels are available. Each channel is in the form of a 64-bit accumulator that counts the fractional performance every PCLK clock cycle when **IECDPMCLKEN** is HIGH.

Dynamic Voltage Controller (DVC)
A system specific block that controls an off-chip power supply unit and interfaces to the IEC and DCG blocks

Dynamic Voltage Scaling (DVS)
Support for controlling the multi-level power supply to a subsystem to reduce voltage when less than 100% of the performance is required by the subsystem. This is achieved using dynamic software control.

Energy Management Unit (EMU)
The NSC implementation of a PSU that works closely with the APC to provide a closed loop voltage scaling solution.

Intelligent Energy Controller (IEC)
The component that provides a standard API interface for the IEM software to set performance levels regardless of any system specific blocks such as the DVC and the DCG. It also provides monitoring functions through the DPM.

Intelligent energy management
A process to enable prolonged battery life of a device by dynamically controlling the power level.

Intelligent Energy Manager (IEM)
An energy manager solution consisting of both software and hardware components that function together to prolong battery life in a device.

Implementation- defined
A feature that is not architecturally defined, and can vary between implementations. The feature is defined and documented for each individual implementation.

Processor A contraction of microprocessor. A processor includes the CPU or core, plus additional components such as memory, and interfaces. These are combined as a single macrocell, that can be fabricated on an integrated circuit.

Register A temporary storage location used to hold binary data until it is ready to be used.

Reserved A field in a control register or instruction format is reserved if the field is to be defined by the implementation, or produces Unpredictable results if the contents of the field are not zero. These fields are reserved for use in future extensions of the architecture or are implementation-specific. All reserved bits not used by the implementation must be written as zero and are read as zero.

Index

E

F

H

I

M

N

O

P

R

S

T

W

