Guide to Cell Broadband Engine Programming Documentation

Version 1.0

August 2008

Sony Corporation
1-7-1 Konan, Minato-ku,
Tokyo, 108-0075 Japan

Sony Computer Entertainment Inc.
2-6-21 Minami-Aoyama, Minato-ku,
Tokyo, 107-0062 Japan

The Sony home page can be found at http://www.sony.net
The SCEI home page can be found at http://www.scei.co.jp
The Cell Broadband Engine home page can be found at http://cell.scei.co.jp

August 2008

Guide to Cell Broadband Engine Programming Documentation

Introduction

This Guide is designed to help you navigate the programming documentation for the Cell Broadband Engine™ (Cell/B.E.) processor. The documents are available on the public Sony Computer Entertainment website (http://cell.scei.co.jp/e_download.html) and the IBM website (http://www-128.ibm.com/developerworks/power/cell/). The Guide summarizes the documentation, explaining the relevance of each document to particular Cell/B.E. programming roles and objectives. Throughout this Guide and the documentation it summarizes, the Cell/B.E. is variously referred to as the Cell Processor, Cell Broadband Engine™, CBE, Cell BE, or BE.

Figure 1, "Cell/B.E. Block Diagram" and Figure 2, "Storage Domains" define the acronyms used in the Cell/B.E. documentation. For a more complete glossary, see the *CBE Programming Handbook*.

Figure 1 Cell/B.E. Block Diagram

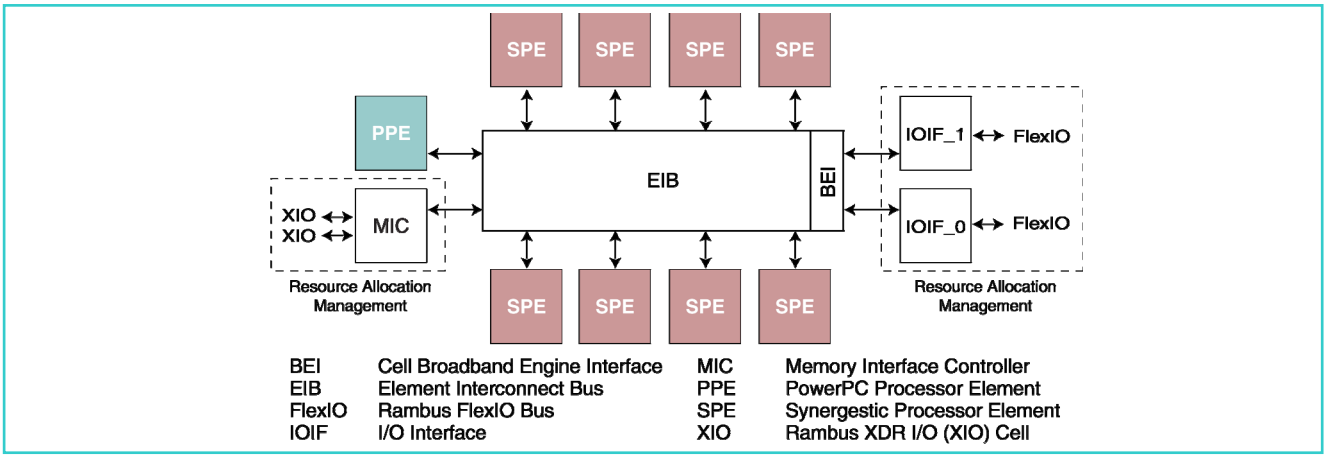
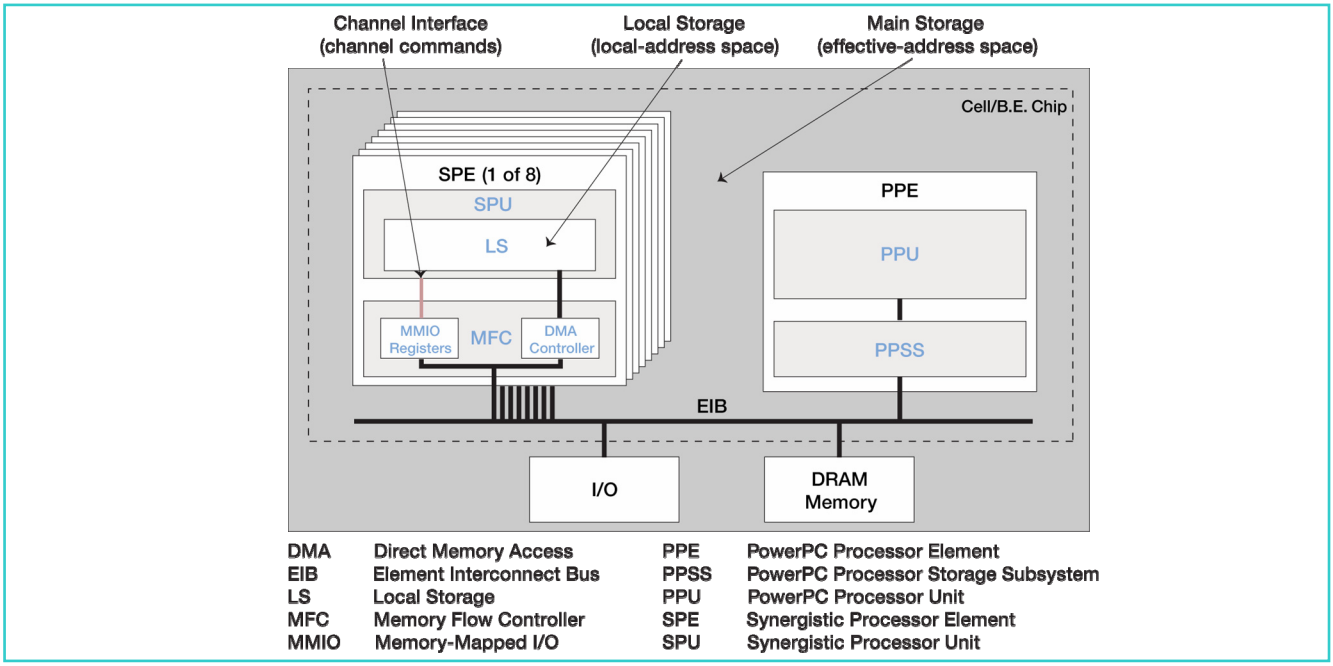


Figure 2 Storage Domains



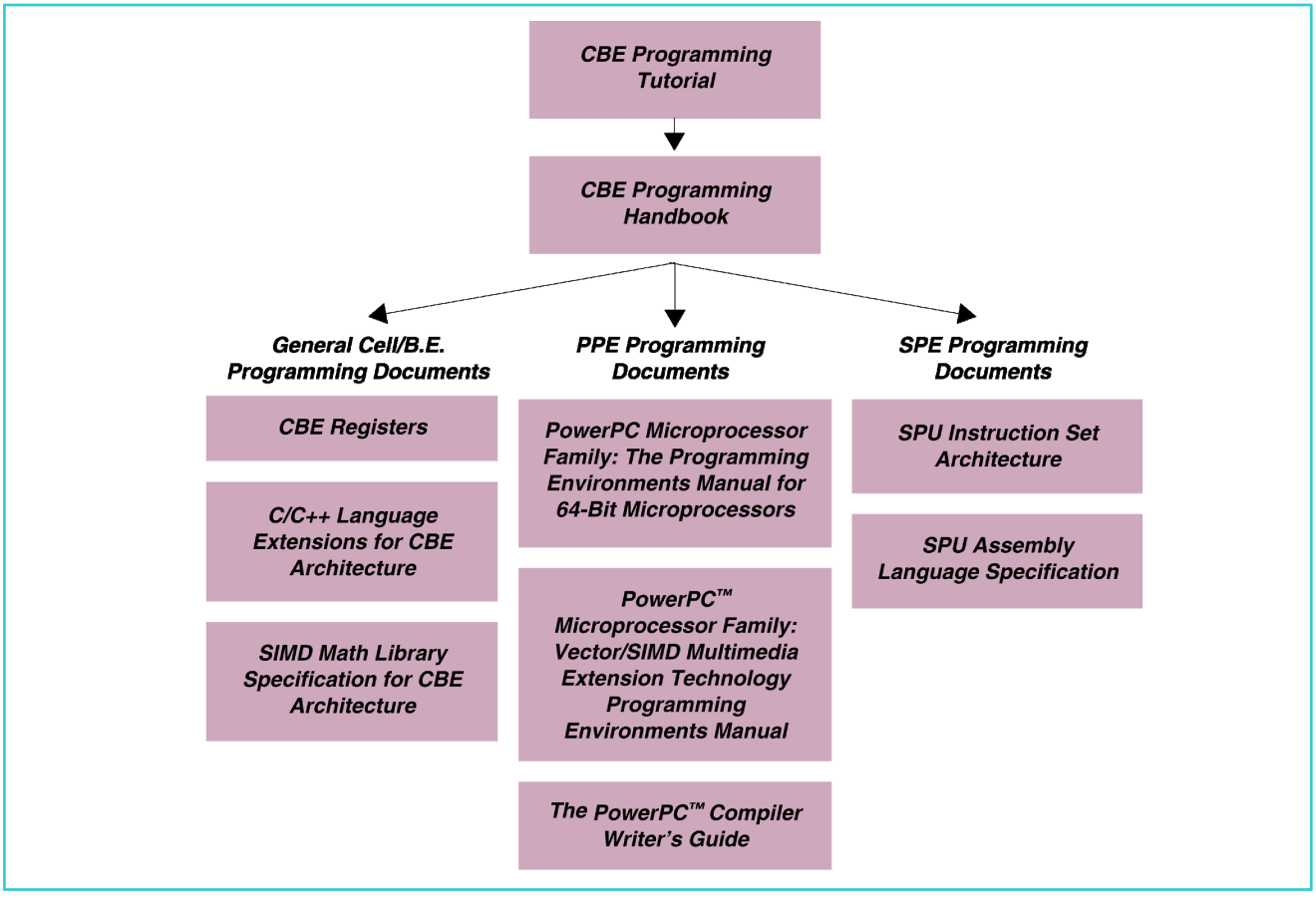
The PPE (Figure 1 and Figure 2) is a general-purpose, 64-bit PowerPC RISC processor that runs the operating system(s) for all applications and external interfaces on the Cell/B.E. It operates on scalars or vectors containing 8- to 64-bit fixed-point or single- or double-precision floating-point data types.

Each of the eight SPEs (Figure 1 and Figure 2) is a 128-bit RISC processor that executes the SPU instruction set, which is designed for data-rich single-instruction, multiple data (SIMD) and scalar applications. It operates on scalars or vectors containing 8- to 128-bit fixed-point or single- or double-precision floating-point data types. An SPE consists of two main units – the SPU, which executes instructions, and the MFC, which controls DMA transfers and related functions. In general, the term "SPU" refers to the instruction-execution unit, and the term "SPE" refers to either or both of the SPU and MFC. However, some documents use the terms "SPU" and "SPE" interchangeably.

Recommended Order of Reading Cell/B.E. Documents

Figure 3, "Recommended Order of Reading" illustrates the documents described in this guide and the recommended reading order. An appendix lists some public websites for Cell/B.E. and PLAYSTATION®3 (PS3) programmers. Programmers writing code only for the PPE do not need to read the SPE documents, and programmers writing code only for the SPEs do not need to read the PPE documents.

Figure 3 Recommended Order of Reading



General Conventions of Microprocessor and Cell/B.E. Documentation

The Cell/B.E. documentation set, in general, observes industry-wide conventions with respect to microprocessor documentation. Specifically, document titles have the following content implications:

- *Architecture* – An architecture document is a specification of requirements for a microprocessor or computer system. It does not specify details of how the microprocessor or computer system must be implemented; instead it provides a template for a family of compatible implementations. The target audience may include application programmers, system programmers, and hardware developers, depending on the actual content of the document.
- *Microarchitecture*—A microarchitecture document describes the internal functions of microprocessor hardware, and it normally illustrates these functions as general block diagrams, Boolean-logic diagrams, and/or circuit diagrams. The target audience may be application and/or system programmers, depending on the actual content of the document.
- *Instruction Set Architecture (ISA)* – An ISA document describes a microprocessor model that is seen by compiled software (machine or object code), including instructions, address and data formats, registers, memory organization, and state variables on which the instructions operate. The target audience is application and system programmers.
- *Application Binary Interface (ABI)* – An ABI document describes the interface between compiled software (machine or object code) and a particular pairing of system software (operating system, library, or other software service) and microprocessor hardware. The target audience is system programmers.
- *Application Programming Interface (API)* – An API document describes the interface between uncompiled software (source code) and an operating system, library, or other software service, independent of any particular microprocessor hardware. The target audience is application and system programmers.
- *Language Specification* – A language specification describes a programming language, including grammar, lexicon, register usage, and related details. The target audience is application and system programmers.
- *Registers* – A Registers document describes register addresses and how register contents affect microprocessor operations. The fields of the registers are illustrated, bit-by-bit, and their functions are described. All registers, including both problem-state (user) and privilege-state (supervisor) registers, are typically described. The target audience is application and system programmers.

Special Terms Used in Cell/B.E. Documentation

- *Privilege State (Supervisor)* – The state in which system software has system-wide control over all other software processes.
- *Problem State (User)* – The state in which application software runs under system software.
- *Intrinsics* – C/C++ function calls mapped to one or more Cell/B.E. assembly instructions.
- *SIMD* – Single-instruction, multiple-data.

Introductory Documents

CBE Programming Tutorial

Cell Broadband Engine Public Information and Downloads

http://cell.scei.co.jp/e_download.html

Length

190 pages.

Audience

C/C++ programmers interested in developing applications or libraries for the Cell/B.E. The document is not intended for programmers developing device drivers, compilers, or operating systems.

Content

A short overview of the Cell/B.E. hardware organization, programming methods, programming examples, and process of writing and debugging programs. The document assumes access to the Cell/B.E.-specific IBM software development kit (SDK), 64-bit Linux® OS and standard toolset, and Full System Simulator. The examples in the tutorial highlight the general principals required for Cell/B.E. programming, so that a programmer can apply this knowledge to other Cell/B.E. development environments.

Recommended Use

This is a beginning tutorial, useful during initial attempts to understand the programming process.

Document Sections

The tutorial has the following sections:

1. *Overview of the Cell Broadband Engine*: A short summary of the Cell/B.E.'s hardware organization and programming methods.
2. *The PPE and the Programming Process*: A summary of the PowerPC Processor Element (PPE), including registers, instruction sets, communication and interoperability with SPEs, and examples of code development and debugging on the simulator.
3. *Programming the SPEs*: A summary of the Synergistic Processor Elements (SPEs), including registers, memory flow controller (MFC), channels, instruction and MFC-command sets, coding and code-porting examples, performance analysis, and general programming tips.
4. *Programming Models*: Summaries of various programming models, including function offload, device extension, computation acceleration, streaming, shared-memory multiprocessor, asymmetric-thread runtime, and user-mode thread models. Also a short summary of application frameworks and SPE overlays.
5. *The Simulator*: A summary of the IBM Full System Simulator for the Cell Broadband Engine, including basic functions, command-line and GUI interfaces, and performance monitoring.

Recommended Prerequisites

None.

Further Details

See:

- *CBE Programming Handbook*
- *Programming the Cell Broadband Engine: Examples and Best Practices*

CBE Programming Handbook

Cell Broadband Engine Public Information and Downloads

http://cell.scei.co.jp/e_download.html

Length

877 pages.

Audience

C/C++ and assembly-language programmers interested in developing applications, libraries, middleware, drivers, compilers, or operating systems for the Cell/B.E.

Content

A general programming user guide and reference for the Cell/B.E., with content for programmers interested in developing almost any kind of application or system software. It describes all of the Cell/B.E. facilities needed to develop such programs. It is system-independent; it makes no assumptions about development-tool or operating-system environments.

Recommended Use

Use this document throughout Cell/B.E. software development. Use it together with the CBE Registers document and other PPE and SPE reference documents.

Document Sections

The handbook has the following sections:

- *Sections 1 to 3 (for application and system programmers):* General hardware and programming-environment overviews of the Cell/B.E., its PPE, and its SPEs.
- *Sections 4 to 16 (primarily for system programmers):* Additional hardware and programming topics for the privilege-state (supervisor) programming environment. There are chapters describing the virtual storage environment, memory and cache management, I/O architecture, resource allocation, PPE interrupts, PPE multithreading, logical partitions and hypervisors, SPE context switching, the time base and decrementers, software objects and executables, SPE loading, power and thermal management, and performance monitoring. Of these, only the decrementers topic is typically needed by application programmers.
- *Sections 17 to 24 and Appendixes (for application and system programmers):* More problem-state (user) programming topics, including SPE channels and related memory-mapped I/O (MMIO) interface, SPE events, DMA transfers and interprocessor communication, shared-storage synchronization, parallel programming, SIMD programming, relationship between PPE vector/SIMD multimedia extension programming and SPE programming, and SPE programming tips. The appendices summarize the PPE instruction set and intrinsics, SPU instruction set and intrinsics, and performance-monitor signals.
- *Glossary:* The best available glossary for Cell/B.E. terminology.

Recommended Prerequisites

None.

Further Details

See:

- IBM Cell Broadband Engine Resource Center
- *Programming the Cell Broadband Engine: Examples and Best Practices* (Parts 2, 3, and 5).
- Several additional references are given throughout the document itself.

General Cell/B.E. Programming Documents

CBE Registers

Cell Broadband Engine Public Information and Downloads

http://cell.scei.co.jp/e_download.html

Length

358 pages.

Audience

Assembly-language programmers writing application or system programs.

Content

Hardware and software details of how the problem-state (user) and privilege-state (supervisor) registers operate. The register fields are illustrated, bit-by-bit, and their functions are described. All registers are considered to be memory-mapped I/O (MMIO) registers, whether or not the registers are associated with an I/O device. An MMIO register is any internal or external register that is accessed through the main-storage space with load and/or store instructions.

Recommended Use

Use this document throughout software development, together with the *CBE Programming Handbook* and other PPE and SPE reference documents.

Document Sections

The document has the following sections:

1. *Cell Broadband Engine Memory-Mapped I/O Registers:* The base addresses and offset ranges of all CBE registers.
2. *PowerPC Processor Element (PPE) MMIO Registers:* Registers used by for the PPE.
3. *Synergistic Processor Element MMIO Registers:* Registers used by the SPEs.
4. *BEI I/O Command (IOC) MMIO Registers:* Registers used by the I/O interface controller for commands.
5. *IOC Address Translation MMIO Registers:* Registers used by the I/O interface controller for address translation.
6. *Internal Interrupt Controller (IIC) MMIO Registers:* Registers used by the internal interrupt controller.
7. *Memory Interface Controller (MIC) MMIO Registers:* Registers used by the memory interface controller.
8. *Token Manager (TKM) MMIO Registers:* Registers used by the token manager.
9. *CBE Distribution (BED) of I/O MMIO Registers:* Registers used by the CBE distribution bus.
10. *Element Interconnect Bus (EIB) MMIO Registers:* Registers used by the element interconnect bus, which handles communication between the PPE, SPEs, memory, and I/O devices.
11. *Pervasive MMIO Registers:* Registers used by the pervasive logic, which performs power management, thermal management, clock control, software-performance monitoring, and trace analysis.
12. *PowerPC Processor Element Special Purpose Registers:* Special purpose registers (SPRs) used in privileged state by the PPE and read or written using special PowerPC instructions.

Recommended Prerequisites

Read or browse:

- *CBE Programming Handbook*

Further Details

See:

- *PowerPC Microprocessor Family: The Programming Environments Manual for 64-Bit Microprocessors*
- *Cell Broadband Engine Architecture*
- *PowerPC Architecture Books I, II, and III*

C/C++ Language Extensions for CBE Architecture

Cell Broadband Engine Public Information and Downloads

http://cell.scei.co.jp/e_download.html

Length

150 pages.

Audience

C/C++ application and system programmers.

Content

Extensions (intrinsics) to the C and C++ languages that allow software developers to access hardware features not accessible (or not easily accessible) from C or C++. The extensions allow SIMD and other applications to obtain PPE or SPE performance comparable to assembly-language programs. The document also includes function specifications to facilitate communication between SPEs and the PPE, and it lists a minimal set of standard library functions that must be provided as part of a standard SPE programming environment.

Recommended Use

Browse during first readings of the *CBE Programming Handbook*, and refer to it thereafter as needed during coding of applications for the PPE or SPEs.

Document Sections

The document has the following sections:

1. *Data Types and Programming Directives:* PPU vector/SIMD multimedia extension and SPU vector data types, operations on these data types, programming directives, and predefined macro target definitions.
2. *SPU Low-Level Specific and Generic Intrinsics:* The basic intrinsics and built-ins that make the underlying SPU ISA and SPE hardware accessible from the C programming language.
3. *Composite Intrinsics:* Intrinsics that can be constructed from a series of low-level (generic or specific) intrinsics.
4. *Programming Support for MFC Input and Output:* Utility functions implemented either as macro definitions or built-in functions that support MFC input or output.
5. *SPU and Vector Multimedia Extension Intrinsics:* Mapping between PPU vector/SIMD multimedia extension intrinsics and SPU intrinsics.
6. *PPU VMX Intrinsics:* Intrinsics that make the underlying PPU vector/SIMD multimedia extension instruction set accessible from C and C++.
7. *PPU Intrinsics:* A minimal set of specific intrinsics to make the underlying PPU instruction set accessible from C.
8. *SPU C and C++ Standard Libraries and Language Support:* Differences between C and C++ standard libraries on the SPU and the corresponding IS0/IEC standards, and common language features that are not supported on the SPU.
9. *Floating-Point Arithmetic on the SPU:* Differences between Annex F of the C99 language floating-point standard (ISO/IEC 9899) and the SPU implementations that apply to SPU compilers and libraries.

Recommended Prerequisites

Read or browse:

- *CBE Programming Handbook*

Further Details

See:

- *SIMD Math Library Specification for CBE Architecture*

SIMD Math Library Specification for CBE Architecture

Cell Broadband Engine Public Information and Downloads

http://cell.scei.co.jp/e_download.html

Length

40 pages.

Audience

C/C++ application programmers interested in developing single-instruction, multiple-data (SIMD) applications.

Content

A math library that takes advantage of the SIMD instructions supported by the PPE and SPEs.

Recommended Use

Browse during first readings of the *CBE Programming Handbook*, and refer to it thereafter as needed during coding of SIMD applications.

Document Sections

The document has the following sections:

1. *Overview of the SIMD Math Library:* Library and header files, function overviews, and special cases.
2. *SIMD Function Specifications:* Type definitions and detailed function descriptions.

Recommended Prerequisites

Read or browse:

- *CBE Programming Handbook*
- *C/C++ Language Extensions for CBE Architecture*

Further Details

None.

PPE Programming Documents

PowerPC Microprocessor Family: The Programming Environments Manual for 64-Bit Microprocessors

Cell Broadband Engine Public Information and Downloads

http://cell.scei.co.jp/e_download.html

Length

657 pages.

Audience

Assembly-language programmers writing application or system programs for the PPE.

Content

An in-depth description of the 64-bit PowerPC Instruction Set Architecture (ISA), which is implemented by the PPE. The manual describes all of the user-level and supervisor-level instructions and software-visible resources.

Recommended Use

Browse after readings of the *CBE Programming Handbook* and refer to it thereafter as needed during coding of PPE applications.

Document Sections

The document has the following sections:

1. *Overview:* Summarizes the PowerPC ISA.
2. *PowerPC Register Set:* The software-related details about how registers operate.
3. *Operand Conventions:* Conventions for storing values in registers and memory, accessing registers, representing data in registers, floating-point data formats, and exception conditions.
4. *Addressing Modes and Instruction Set Summary:* A summary of instructions and addressing modes for integer, floating-point, load-store, flow control, processor control, memory synchronization, and external control operations.
5. *Cache Model and Memory Coherency:* The cache model, built-in architectural controls for maintaining memory coherency, cache-control instructions, and special concerns for memory coherency in single-processor and multiprocessor systems.
6. *Exceptions:* The way in which the PPE implements exceptions (also called “interrupts”).
7. *Memory Management:* Memory management unit (MMU) functions.
8. *Instruction Set:* Detailed descriptions of the 64-bit PowerPC instructions.

There are also appendices describing multiple-precision shifts, floating-point models, synchronization programming examples, and simplified mnemonics.

Recommended Prerequisites

Read or browse:

- *CBE Programming Handbook*

Further Details

See:

- *PowerPC™ Microprocessor Family: Vector/SIMD Multimedia Extension Technology Programming Environments Manual*
- *SIMD Math Library Specification for CBE Architecture*
- *The PowerPC™ Compiler Writer’s Guide*
- *PowerPC Architecture Books I, II, and III*

PowerPC™ Microprocessor Family: Vector/SIMD Multimedia Extension Technology Programming Environments Manual

Cell Broadband Engine Public Information and Downloads

http://cell.scei.co.jp/e_download.html

Length

317 pages.

Audience

Assembly-language programmers writing application programs for the PPE.

Content

The PowerPC Vector/SIMD Multimedia Extension instruction set architecture (ISA), and how it relates to the 64-bit PowerPC ISA.

Recommended Use

Browse after readings of the *CBE Programming Handbook* and refer to it thereafter as needed during coding of PPE applications.

Document Sections

The document has the following sections:

1. *Overview:* Summarizes the instructions set architecture.
2. *Vector Register Set:* The software-related details about how registers operate.
3. *Operand Conventions:* Conventions for storing values in registers and memory, accessing registers, representing data in registers, floating-point data formats, and exception conditions.
4. *Addressing Modes and Instruction Set Summary:* A summary of instructions and addressing modes for vector integer arithmetic, vector floating-point arithmetic, vector load and store, vector permutation and formatting, processor control, and memory control operations.
5. *Cache, Exceptions, and Memory Management:* The cache, exceptions (interrupts), and memory-management models.
6. *Vector Processing Instructions:* Detailed descriptions of the instructions.

Recommended Prerequisites

Read or browse:

- *CBE Programming Handbook*
- *PowerPC Microprocessor Family: The Programming Environments Manual for 64-Bit Microprocessors*

Further Details

See:

- *SIMD Math Library Specification for CBE Architecture*

The PowerPC™ Compiler Writer's Guide

Cell Broadband Engine Public Information and Downloads

http://cell.scei.co.jp/e_download.html

Length

264 pages.

Audience

Although the title indicates an audience of compiler writers, the document is useful to all assembly-language programmers who wish to optimize performance of application or system programs for the PPE.

Content

This guide describes, mainly by coding examples from IBM's most experienced compiler developers, the code patterns that perform well on PowerPC processors. It will be particularly helpful to compiler developers and application-code specialists who are already familiar with optimizing compiler technology and are looking for ways to exploit the PowerPC architecture. It will also be helpful to application programmers who need to understand and tune the output of PowerPC compilers. The guide focuses only on a complier's generation of efficient PowerPC back-end code; it does not attempt to teach how to write an entire compiler or the accompanying library routines.

Recommended Use

Browse after browsing the *PowerPC Microprocessor Family: The Programming Environments Manual for 64-Bit Microprocessors* and refer to it thereafter as needed during coding of PPE software.

Document Sections

The document has the following sections:

1. *Introduction:* Summarizes RISC architectures, optimizing compilers, and assumptions.
2. *Overview of the PowerPC Architecture:* Summarizes the PowerPC application environment and instruction set.
3. *Code Selection:* How to choose optimal code for control-flow, integer and string operations, and floating-point operations.
4. *Implementation Issues:* Hardware implementations, hazards, scheduling, and alignment.
5. *Clever Examples:* A collection of common algorithms implemented in efficient assembly code.

There are also appendices describing application binary interface (ABI) considerations, early PowerPC 6xx hardware implementations, PowerPC instruction-usage statistics, and optimal code sequences.

Recommended Prerequisites

Read or browse:

- *CBE Programming Handbook*
- *PowerPC Microprocessor Family: The Programming Environments Manual for 64-Bit Microprocessors*
- *PowerPC™ Microprocessor Family: Vector/SIMD Multimedia Extension Technology Programming Environments Manual*

Further Details

See:

- *PowerPC Architecture Books I, II, and III*

SPE Programming Documents

SPU Instruction Set Architecture

Cell Broadband Engine Public Information and Downloads

http://cell.scei.co.jp/e_download.html

Length

278 pages.

Audience

Assembly-language programmers writing application or system programs for the SPE.

Content

The instruction set architecture (ISA) implemented by the SPEs, including descriptions of instruction functions, instruction and data formats, interrupts, and storage-access synchronization.

Recommended Use

Browse after readings of the *CBE Programming Handbook* and refer to it thereafter as needed during coding of SPE applications.

Document Sections

The document has the following sections:

1. *Introduction:* Summarizes the SPU ISA.
2. *SPU Architectural Overview:* Descriptions of the instruction and data formats.
3. *Memory – Load/Store Instructions:* Descriptions of the load, store, and generate-controls instructions.
4. *Constant-Formation Instructions:* Descriptions of the immediate-load instructions.
5. *Integer and Logical Instructions:* Descriptions of the integer, logical, count, form-select, gather, select-bits, shuffle-bytes, and related instructions.
6. *Shift and Rotate Instructions:* Descriptions of the shift and rotate instructions.
7. *Compare, Branch, and Halt Instructions:* Descriptions of the compare, branch, and halt instructions.
8. *Hint-for-Branch Instructions:* Descriptions of the hint-for-branch instructions.
9. *Floating-Point Instructions:* Descriptions of the floating-point instructions.
10. *Control Instructions:* Descriptions of the control instructions, including stop-and-signal, synchronize, and move to/from special-purpose register (SPR).
11. *Channel Instructions:* Descriptions of the channel read and write instructions.
12. *SPU Interrupt Facility:* Descriptions of the interrupt facility.
13. *Synchronization and Ordering:* Descriptions of the storage-access synchronization and ordering facilities.

Recommended Prerequisites

Read or browse:

- *CBE Programming Handbook*
- *C/C++ Language Extensions for CBE Architecture*

Further Details

See:

- *SPU Assembly Language Specification*

SPU Assembly Language Specification

Cell Broadband Engine Public Information and Downloads

http://cell.scei.co.jp/e_download.html

Length

31 pages.

Audience

Assembly-language programmers writing application or system programs for the SPE.

Content

The grammar, lexicon, register usage, and related details for an SPE assembly language.

Recommended Use

Browse after readings of the *CBE Programming Handbook* and refer to it thereafter as needed during assembly-language coding of SPE applications.

Document Sections

The document has the following sections:

1. *Introduction:* Summarizes the contents of the document.
2. *Instruction Set and Instruction Syntax:* Descriptions of notation and data formats, the instruction set, register and instruction aliases, channel mnemonics, encoding of immediate values, and errors and warnings.

Recommended Prerequisites

Read or browse:

- *CBE Programming Handbook*
- *SPU Instruction Set Architecture*

Further Details

None.

Appendix – Programming Links

This appendix provides links to public websites containing information about programming the Sony Computer Entertainment PLAYSTATION®3 (PS3) platform and the Cell/B.E.:

- *PLAYSTATION®3 Websites:*
  - Sony Computer Entertainment (SCE) PLAYSTATION®3 Developer Network: http://www.scedev.net
  - Game Developers Conference (GDC): http://devevents.scedev.net/gdc
  - Sony Computer Entertainment Europe (SCEE): http://www.technology.scee.net
  - Insomniac Games: http://www.insomniacgames.com/tech/techpage.php
- *Cell/B.E. Websites:*
  - Sony Computer Entertainment, Inc. (SCEI): http://cell.scei.co.jp/index_e.html
  - Power.org: http://www.power.org/resources/devcorner/cellcorner
  - IBM: http://www-128.ibm.com/developerworks/power/cell/
  - CellPerformance: http://www.cellperformance.com
  - Beyond 3D Forum: http://forum.beyond3d.com/forumdisplay.php?f=57