

Contents

Part I Introducing the Cell Broadband Engine

1	Introduction	3
1.1	About this book	3
1.2	Background of the CBE Architecture and Processor	5
1.3	Design of the Cell Architecture	9
1.4	The Cell as a supercomputer	11
1.5	Major Components of the Cell Broadband Engine	13
1.6	Flex I/O, DDR/XDR, Interrupt Controller	14
1.7	Conclusions	14
2	The Power Processing Element (PPE)	17
2.1	PPE: the Control Plane of the CBEA	17
2.2	PPE Problem State Registers	19
2.3	Memory arrays in the CBEA	21
2.3.1	PPE Caches	22
2.3.2	SPU Local Store	22
2.4	Viewing the PPE as a dual-core processor: EMT specifics	23
2.4.1	PPU Intrinsics for managing thread priorities and delays	25
2.5	PowerPC Instruction Set	26
2.6	SPE Memory Management and EA Aliasing	26
2.7	Power Processor Element (PPE) Cache Utilization	27
2.7.1	The <code>eiio</code> instruction	28
2.7.2	Instruction cache	28
2.7.3	Data cache miss	29
2.8	Understanding the PPE Pipeline	29
2.8.1	Discuss assembly language	33
2.9	Measuring and Profiling Application Performance	33
2.10	Conclusion	34

3	The Synergistic Processing Element	35
3.1	Introduction to Cell SPE	35
3.2	SPE Local Store	38
3.3	SPU LS Priority	39
3.4	GPRs in the SPE	40
3.5	SPE Intrinsics and Instruction	40
3.6	SPU Instruction Set: Pipelines, Execution Units and Latency	41
3.7	Load/Store Optimization on SPU	44
3.8	Branch Optimization on SPU	47
3.8.1	Branch elimination	48
3.9	Instruction Scheduling	49
3.10	Memory flow	49
3.11	Mailbox Facility	50
3.12	Use SPU Intrinsics with Mailboxes	54
3.13	Synergistic Processor Unit Channels and Memory Flow Controller	56
3.14	SPU Timer and Events	58
3.15	SPE Context on the PPE	58
3.16	Conclusion	58
3.17	What we have not discussed	59
4	Element Interconnect Bus	61
4.1	Introduction to the Element Interconnect Bus	61
4.1.1	EIB Possible Livelock Detection	65
4.2	Important things to remember about the EIB	65
4.3	Conclusions	66
5	Direct Memory Access (DMA)	67
5.1	Introduction to DMA in the Cell Broadband Engine	67
5.1.1	Difference between fence and barrier synchronization	68
5.2	Conclusions	71

Part II Programming the Cell Broadband Engine

6	Foundations for Program Development on CBE	75
6.1	Theory of Parallel Programming	75
6.1.1	Amdahl's Law	76
6.1.2	Gustafson's Law	77
6.2	The Class NC of complexity analysis	77
6.3	Parallel programming concepts	78
6.3.1	Parallel programming terminology	79
6.4	Introduction to pthreads	79
6.4.1	Understanding pthread programming model	80
6.4.2	Pthreads Keys: using thread specific data	81
6.4.3	Pthreads Summary	82
6.5	Multi-faceted parallelism in the CBE Architecture	84
6.6	Alignment using compiler attributes	84

6.7	OpenMP	86
6.7.1	SPE Programming	87
6.8	Double buffering example	88
6.9	Software Managed Cache	88
6.10	Physical Organization	89
6.11	Things to watch out for	89
6.12	Strategies and Paradigms for Identifying Parallelism	91
6.13	Conclusion	92
7	The development environment	93
7.1	How to setup an PS3 for Cell development	93
7.2	Alternatives to installing GNU/Linux on the PS3	94
7.3	The CESOF Format	96
7.4	Basic development environment	98
7.5	Debugging code on the SPU	99
7.6	Conclusions	101
8	Hello World	103
8.1	Writing simple programs on the Cell Broadband Engine	103
8.2	SPE Program Development	106
8.3	Using pthread with SPU tasks	108
8.4	Developing applications using libspe2	109
8.4.1	Major functions of the libspe2 library	109
8.5	Measuring Performance using High Resolution Timers	121
8.6	Synchronization	123
8.7	Conclusions	131
9	An Overview of the SDK	133
9.1	Major components of the Cell Broadband Engine SDK 3.0	133
9.2	Application programming libraries in the SDK	134
9.3	SPE Runtime Library	134
9.4	SPU Timer Library	135
9.5	Monte-Carlo Library	135
9.6	BLAS Library	136
9.7	LAPACK Library	137
9.8	SIMD Math API and Library	138
9.9	SPU Software Managed Cache	138
9.10	FFT Library	138
9.11	Game Math Library	138
9.12	Image Library	139
9.13	Large Matrix Library	139
9.14	Matrix Library	139
9.15	Synchronization Library	140
9.16	Vector Library	141
9.17	Multiprecision Math Library	141

9.18	ALF Library and Framework	142
9.19	DACS Library and Framework	143
9.20	Conclusions	143
10	Basic Algorithms	145
10.1	Some number theoretic problems	145
10.1.1	Collatz conjecture	145
10.1.2	The Takeuchi Benchmark	147
10.2	Merit Factor of Sequences and Auto-correlation	148
10.3	Computing Greatest Common Divisor (GCD)	155
10.3.1	Suggested enhancements	157
10.4	Sorting Algorithms	157
10.4.1	Implementation of SIMD Sort	158
10.4.2	The <code>spu_shuffle</code> function	158
10.4.3	The <code>spu_rlqbyte</code> function	161
10.5	Table lookup	163
10.6	(Perfect) Hash Function	164
10.7	Minimal Perfect Hashing	166
10.8	Implementing minimal perfect hash in the SPU	168
10.9	Using perfect hash functions with large graph clusters	169
10.10	Heap Data Structure	169
10.11	Disjoint Set Union-Find Algorithms	170
10.12	Computing Zech logarithm using SPE in batch mode	172
10.12.1	How to calculate $Z(x)$	173
10.13	Conclusions	175
11	Graph Theory on the CBEA	177
11.1	Introduction	177
11.2	Definitions	177
11.3	Graph representation	178
11.3.1	Min-cut	179
11.4	Minimum Spanning Tree using Kruskal's Method	181
11.5	Revisiting min-cut using MST and Karger's Method	188
11.6	Dijkstra's Algorithms for Shortest Paths	189
11.7	Floyd-Warshall All-pairs shortest-path	192
11.8	Formal definition of APSP (all-pairs shortest-path)	192
11.9	Maximal Independent Set	194
11.10	Approximation algorithm for maximal matching	194
11.11	Partitioning using Simulated Annealing	197
11.11.1	k -way Partitioning : Problem definition	198
11.11.2	The cost function	199
11.11.3	Terminal Propagation	200
11.12	Simulated Annealing	200
11.12.1	Implementation of Simulated Annealing on the SPU	202
11.13	Partitioning using Spectral Methods	207

11.13.1	Theoretical formulation of spectral partitioning	208
11.13.2	Application to graph partitioning and finite-element meshes	209
11.14	On-line streaming algorithm for graph properties	212
11.14.1	Discussion of implementation	218
11.15	Conclusion	221
12	Alternative methods for parallel programming on SPE	223
12.1	Introduction	223
12.2	NESL: Nested Data Parallel Language	223
12.2.1	The Parallel Apply-to-each construct	224
12.2.2	Nested parallelism	224
12.3	CVL and VCODE: Vector Libraries	226
12.3.1	CBE/SPE specific changes to VCODE	228
12.4	A Simplified SPU ISA Simulator	228
12.5	VCODE Compiler	235
12.6	PowerList: Data structure for parallel programming	237
12.6.1	Functions on Powerlists	238
12.6.2	Using powerlists for polynomial evaluation	239
12.7	ALF: Accelerated Library Framework	239
12.7.1	ALF workload division	240
12.8	Tale of Dwarfs	241
12.9	Introduction to Cilk++	242
12.10	Introduction to Intel Ct	243
12.11	Introduction to RapidMind	243
12.11.1	Conclusion	244
13	Computational Mathematics on the CBEA	245
13.1	Introduction	245
13.2	Calculating PI	245
13.3	Line fitting	247
13.4	Representing Monomials and Polynomials	251
13.4.1	Application of algebraic geometry in robotics	251
13.4.2	Application of algebraic geometry in CAD	251
13.4.3	Implicitization of Curves and Parametric Forms	252
13.5	Evaluating polynomials	255
13.6	SIMD Polynomial evaluation with fixed step	256
13.7	Parallel Polynomial Evaluation	256
13.8	Evaluating Boolean Functions	259
13.9	Parallel Matrix Functions	259
13.10	Solving Equations in a Single Complex Variable	259
13.10.1	Performance Comparisons	275
13.11	Conclusions	285

14	Vector Graphics on SPU	287
14.1	Introduction	287
14.2	Basic Computational Geometry on SPU	287
14.2.1	Reading Polygon Data	288
14.2.2	Checking Polygon Convexity	289
14.2.3	Computing Convex Hull of 2d Data	291
14.2.4	Partitioning point set data to do parallel convex hull	292
14.2.5	Computing the bounding box of a polygon	294
14.2.6	Computing Polygon Signed Area using SIMD	295
14.2.7	Implementation of parallel line-clipper	296
14.2.8	Implementation of parallel polygon clipper	298
14.3	Conclusion	299
15	Optimizing SPU Programs	301
15.1	Introduction to Program Optimization for Cell	301
15.2	GNU gprof	301
15.3	Branch probability extraction	302
15.3.1	Using -fprofile-generate	303
15.3.2	Using -fprofile-use	303
15.4	SPU Assembly Analysis	303
15.4.1	SPU-Timing tool	304
15.4.2	IBM ASM Visualizer	305
15.5	FDPRPRO: Optimization Tool	306
15.6	Conclusion	308

Part III Case Studies

16	Line-of-sight Computation	311
16.1	Introduction to the line-of-sight computation	311
16.2	Basic equation for LOS calculation	314
16.3	Watershed analysis using LOS	316
16.4	LOS formulation using Bresenham's line-drawing for interpolation	316
16.5	SIMD Implementation of line-of-sight algorithm	319
16.6	Watershed Analysis	320
16.6.1	Watershed analysis on multiple SPUs	321
16.7	SPU Implementation	321
16.8	Conclusion	327
17	Structure Determination using PDF	329
17.1	Problem of nano-structure determination	329
17.2	Implementation of the <i>ab-initio</i> PDF method	330
17.3	Algorithms and data structures required for the problem	332
17.4	Integrate visualization using OpenGL	332
17.5	Conclusions	347

18	Polytope Enumeration	349
18.1	What is a polytope ?	349
18.2	Examples of polytopes	352
18.3	Algorithm	352
18.4	Experimental Results	363
18.5	Future Work	365
18.6	Current Results	365
18.7	Conclusion	366
19	Synthetic Aperture Radar	367
19.1	Introduction	367
19.2	Fundamental principal behind Synthetic Aperture Radar	368
19.2.1	Operational Parameters	369
19.3	Problem Statement for SAR Image Processing	369
19.4	Block Adaptive Quantization	370
19.5	Signal processing functions for SAR	373
19.6	Implementation Details	374
19.7	Experimental Results	377
19.8	Conclusions	379
20	VLSI Design Automation	381
20.1	CAD of Very Large Scale Integrated Circuits	381
20.2	Algorithmic Design and HDL Capture	382
20.2.1	HDL Capture	385
20.2.2	BLIF Format in a nutshell	386
20.3	Equation Processing	389
20.4	Microword Optimization using Local Search on SPU	395
20.4.1	Problem formulation	396
20.4.2	Heuristic techniques & local search	397
20.5	Conclusion	399
21	Floorplanning: VLSI and other Applications	401
21.1	What is a VLSI floorplan	401
21.2	The sequence-pair data-structure	401
21.3	Use of a floorplan in non-VLSI applications	403
21.4	Implementation of a Sequence-pair based Floorplanner	404
21.5	Conclusions	407
22	VLSI Placement	409
22.1	Introduction to VLSI Standard Cell Placement	409
22.1.1	Input to the problem	410
22.2	Reading gate-level BLIF files and making a graph	412
22.3	SPU basic cell placer, using Simulated Annealing	412
22.4	What is the VLSI global routing problem	434
22.5	Conclusion	436

23 Power Estimation for VLSI 437

23.1 Introduction 437

23.2 Static Probabilities 438

23.2.1 Transition Probabilities 439

23.3 Spatial Correlations 440

23.4 SAE 441

23.4.1 The Parker-McCluskey’s Approximation 441

23.5 Bit-vector implementation of probability polynomials 444

23.5.1 Approximation Scheme 446

23.5.2 Virtual inputs at scannable FFs 447

23.6 Applications of PRBC 447

23.7 A Nested Data-Parallel Implementation of the Parker-McCluskey
Method 448

23.7.1 The Parallel Implementation 449

23.8 Analysis of our method 450

23.9 PRBC Implementation 451

23.10 Probability Polynomial Evaluation on SPU 458

23.11 Experiments 459

23.12 Conclusion 460

24 Concluding Remarks 461

24.1 Progress in small steps 461

24.2 Use of Common Lisp 461

24.3 Future Work 462

24.4 Conclusions 464

24.5 Website Information 464

A Introduction to Common Lisp 465

A.1 Introduction to Common Lisp 465

A.2 Functions in Common Lisp 466

A.3 File I/O 467

A.4 Conclusion 467

Index 469

References 480



<http://www.springer.com/978-1-4419-0307-5>

Practical Computing on the Cell Broadband Engine

Koranne, S.

2009, XXXIV, 485 p. 20 illus., Hardcover

ISBN: 978-1-4419-0307-5