

Contents

Acknowledgments	xix
Preface	xxi
Foreword	xxiii
Chapter 1 Introduction	1
Concepts Leading to EPIC	3
<i>Pipelining</i>	3
<i>Load/Store Architecture</i>	4
<i>Cache</i>	6
<i>Branch Prediction</i>	8
<i>Branch Elimination—Predication</i>	9
<i>Superscalar</i>	10
<i>Dynamic Execution</i>	11
Intel® Itanium™ Architecture—New Features	12
<i>Speculation</i>	12
<i>Speculating More Than Loads</i>	14
<i>Predication</i>	15
<i>Parallel Compares</i>	15
<i>Rotating Register File</i>	16
Chapter 2 Data, Code, and Memory	19
Data Types, Sizes, and Formats	19
Application Memory Addressing Model	22
Data and Code Organization and Alignment	22
Byte Ordering	24
	v

vi Contents

Chapter 3 Register Resources	29
Application Register Set and Application Register State	29
Instruction Pointer	31
General Registers	31
Predicate Registers	36
Branch Registers	37
Current Frame Marker Register	38
Application Registers	40
<i>Kernel Registers</i>	42
<i>Register Stack Engine Support Registers</i>	42
<i>Compare and Exchange Value Register</i>	45
<i>User NaT Collection Register</i>	45
<i>Interval Time Counter Register</i>	45
<i>Loop Program Structure Support Registers</i>	45
<i>Previous Function State Register</i>	46
User Mask (UM) Register	47
Processor Identification Registers	48
Performance Monitor Data Registers	51
Chapter 4 Application Instruction Set	53
Instruction Set Architecture	53
Assembly Language Instruction Syntax	54
Instruction Groups	56
<i>Types of Dependencies</i>	56
<i>Identifying Instruction Groups</i>	58
Instruction Bundle Format and Templates	59
Syllable of Instructions	64
Chapter 5 Memory Access and Register Transfer Instructions	65
Memory Access Instructions	65
Load Instruction	66
<i>Load Types</i>	70
<i>Load Hints</i>	73
<i>Load Fill Operation</i>	74
Store Instruction	75
Speculation	78
<i>Control Dependencies and Control Speculation</i>	79
<i>Data Dependencies and Data Speculation</i>	83
Semaphore Instructions	87
<i>Exchange Instruction</i>	88
<i>Compare and Exchange Instruction</i>	89
<i>Fetch and Add Immediate Instruction</i>	90

Register Transfer Instructions	91
The Move Instructions	91
<i>General Register Move Instruction</i>	92
<i>Move Indirect Instruction</i>	93
<i>Move Instruction Pointer Instruction</i>	94
<i>Move Branch Register Instruction</i>	94
<i>Move Predicates Instruction</i>	95
<i>Move Application Register Instruction</i>	97
User Mask Instructions	98
<i>Move User Mask</i>	99
<i>Set User Mask Instruction</i>	99
<i>Reset User Mask Instruction</i>	100
Chapter 6 Integer Computation, Character String, and Population Count Instructions	101
Integer Computation Instruction Grouping	101
Integer Arithmetic Instructions	102
<i>Add Instructions</i>	102
<i>Subtract Instructions</i>	104
<i>Shift Left and Add Instruction</i>	105
Logical Instructions	105
<i>AND, OR, and Exclusive OR Instructions</i>	106
<i>AND Complement Instruction</i>	107
Large Constant Generating Instructions	107
32-bit Integer and 32-bit Address Pointer Instructions	108
<i>Sign-Extend and Zero-Extend Instructions</i>	108
<i>Add Pointer Instruction</i>	110
<i>Shift Left and Add Pointer Instruction</i>	111
Shift and Bit-Field Instructions	112
<i>Shift Left and Shift Right Instructions</i>	113
<i>Shift Right Pair Instruction</i>	114
<i>Extract and Deposit Instructions</i>	116
Character String and Population Count Instruction Grouping	119
<i>Character String Instruction</i>	119
<i>Population Count Instruction</i>	120
Chapter 7 Compare and Branch Instructions	123
Qualifying Instruction Execution with a Predicate	123
Compare Instruction Group	124
General Register Compare Instructions	125
<i>Compare Relationship Completers</i>	125
<i>Comparison Type Completer</i>	126
<i>Parallel Comparison Types</i>	129

viii Contents

<i>Compare Instructions Implemented as a Pseudo-operation</i>	130
<i>Compare Word Instruction</i>	131
Test Bit and Test NaT Bit Instructions	132
Branch Instruction Group	134
<i>Types of Branch Operations and Instructions</i>	134
<i>The Conditional Branch Instruction</i>	136
<i>Branch Hints</i>	137
Eliminating Conditional Branches with Predication	139
Optimizing Code with the Parallel Compare Operation	142
Register Stack and the Register Stack Engine	144
<i>Organization of the Register Stack</i>	144
<i>Parameter Passing</i>	146
<i>Register Stack Engine</i>	148
Conditional Branch Function Support Instructions	150
<i>Structure of a Function Call</i>	150
<i>Function Call Process</i>	153
Loop Program Structures and Loop Support Branch Instructions	156
<i>Nonpipelined While Loop</i>	157
<i>Nonpipelined Counted Loop</i>	158
<i>Pipelined Implementation of Loops</i>	160
Chapter 8 Multimedia Instructions	169
Multimedia Data Structures	169
Multimedia Instruction Group	172
Parallel Arithmetic Instructions	172
<i>Parallel Addition Instructions</i>	172
<i>Parallel Subtraction Instructions</i>	180
<i>Parallel Compare Instruction</i>	182
<i>Parallel Multiplication Instructions</i>	184
<i>Parallel Sum of Absolute Difference Instruction</i>	187
<i>Parallel Minimum and Maximum Instructions</i>	188
Parallel Shift Instructions	190
Parallel Data Arrangement Instructions	193
<i>Multiplex Instruction</i>	193
<i>Mix Instruction</i>	197
<i>Pack Instruction</i>	199
<i>Unpack Instruction</i>	201
Chapter 9 Floating-point Architecture	203
Floating-point Data	203
<i>IEEE Real Numbers</i>	204
<i>IEEE Real Number Memory Formats</i>	206

Floating-point Application Programming Model	207
<i>Floating-point Register File</i>	207
<i>Floating-point Status Register</i>	209
<i>Floating-point Status Register Status Field Instructions</i>	213
<i>Floating-point Fields of the User Mask</i>	215
Floating-point Register Format and Data Types Encoding	216
<i>Floating-point Register Format</i>	216
<i>IEEE Real Numbers</i>	218
<i>Special Numbers and Other Data Types</i>	219
<i>Parallel Floating-point Data</i>	220
Floating-point Memory Access, Register Transfer, and Data Conversion Instructions	221
<i>Floating-point Load, Store, and Load-pair Instructions</i>	221
<i>Floating-point Register Transfer Instructions</i>	229
<i>Integer/Floating-point Conversion Instructions</i>	232
Floating-point Arithmetic Instructions	234
<i>Arithmetic Computations Involving One Source Operand Register</i>	234
<i>Arithmetic Computations Involving Two Source Operand Registers</i>	237
<i>Arithmetic Computations Involving Three Source Operand Registers</i>	240
<i>Division and Square Root Computations</i>	242
<i>Single-precision Floating-point Divide, Latency Optimized</i>	244
<i>Double-precision Floating-point Square Root, Latency Optimized</i>	248
<i>Optimizing a Floating-point Computation Using the Fused Multiply-add Operation</i>	253
Integer Multiplication	254
Integer Divide and Integer Remainder Computations	256
Floating-point Logic Instructions	256
Floating-point Data Arrangement Instructions	258
<i>Floating-point Merge Operation</i>	259
<i>Floating-point Parallel Mix Operation</i>	262
<i>Floating-point Pack Operation</i>	263
<i>Floating-point Swap Operation</i>	264
<i>Floating-point Sign Extend Operation</i>	265
Floating-point Compare and Class Instructions	267
<i>Floating-point Compare Instructions</i>	267
<i>Floating-point Class Instruction</i>	270
Chapter 10 IA-32 Compatibility and Application Execution	273
Operating Environments for Itanium-based Applications	273
Invoking IA-32 Applications	275
<i>Instruction Set Transition Model</i>	275
<i>Invoking the IA-32 Instruction Set</i>	276
<i>Returning to the Itanium Instruction Set</i>	277
IA-32 Application Register Set	278

x Contents

Chapter 11 Compiler Technology for Itanium-based Software	283
Architecture of the Intel Compiler	283
Profile of an Application Program	284
Dynamic Profile-guided Compilation	285
Interprocedural Analysis and Optimization	286
Memory Optimization, Parallelization, and Vectorization	287
Global Scalar Optimization	288
Code Generation and Scheduling	288
<i>Software Pipeliner</i>	290
<i>Instruction Predication</i>	291
<i>Global Code Scheduler</i>	292
<i>Compensation Code</i>	292
<i>Speculation and Predication</i>	293
<i>Downward Code Motion</i>	294
<i>Global Register Allocation</i>	295
Index	297

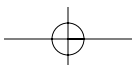
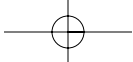
List of Figures

1.1	Pipelined Execution in a Load/Store Architecture	5
2.1	Integer Number Formats	21
2.2	Application Memory Address Space	23
2.3	Little-endian Data	25
2.4	Big-endian Data	26
3.1	Application Register Set	30
3.2	General Register Set	32
3.3	Static and Stacked General Registers	34
3.4	Organization of the Predicate Registers	36
3.5	Branch Registers	38
3.6	Frame Marker Format	39
3.7	RSC Register Format	42
3.8	BSP Register Format	44
3.9	BSPSTORE Register Format	44
3.10	RNAT Register Format	44
3.11	Epilog Count Register Format	46
3.12	PFS Format	47
3.13	User Mask Format	48
3.14	CPUID Register File and Fields	49
3.15	Performance Registers	50
4.1	Identifying Instruction Groups	58
4.2	Instruction Code Bundle Format and Example	60
4.3	Bundle and Instruction Group Boundaries	63
4.4	Instruction Groups and Corresponding Bundles	64
4.5	Syllable of Code	64
5.1	Architectural View of Memory	74
5.2	Branch Barrier Removal with Control Speculation	80

xii *List of Figures*

5.3	Store Barrier Removal with Data Speculation	84
5.4	Data Speculation Routine Involving Speculation Check and Recovery	87
5.5	Relationship Between Predicate Registers and Bit Locations	95
6.1	Add Pointer Operation	110
6.2	Shift Left and Add Pointer Operation	111
6.3	128-bit-input Funnel Shift Operation	114
6.4	Extract Operation	116
6.5	Deposit Operation	117
7.1	Branch Elimination through Predication	139
7.2	Optimized and Nonoptimized Predication for IF-THEN-ELSE Program Structure	141
7.3	Optimizing Code with the Parallel Compare Operation	143
7.4	Organization of the Register Stack	145
7.5	Treatment of Functions	147
7.6	Register Stack Engine Operation	149
7.7	Structure of a Function	151
7.8	Conditional Execution of Function Calls	153
7.9	Nested Functions A and B	154
7.10	Stack Frame Transitions for Call and Return of Function B	155
7.11	Nonpipelined While Loop Program Structure	157
7.12	Search for a Matching Quad Word in a Block of Memory	158
7.13	Nonpipelined Counted Loop Program Structure	159
7.14	Clearing a Block of Memory	160
7.15	Nonpipelined versus Pipelined Loop Execution	161
7.16	Three Treatments of the Counted Loop	162
8.1	Little-endian Ordered Multimedia Parallel Data in Memory and Registers	170
8.2	4×16 Parallel Add Operation	174
8.3	4×16 Parallel Average Operations with Rounding	177
8.4	8×8 Parallel Subtraction Operation	181
8.5	4×16 Parallel Average Subtraction Operation	181
8.6	4×16 Parallel Comparison Operation	182
8.7	Example of an 8×8 Parallel Comparison	184
8.8	Right and Left Parallel Multiplication Operations	185
8.9	Parallel Multiplication and Shift Right Operation	186
8.10	Parallel Sum of Absolute Difference Operation	187
8.11	4×16 Parallel Minimum Operation	189
8.12	4×16 Parallel Maximum Operation	189
8.13	4×16 Parallel Shift Left Operation	192
8.14	8×8 mux Operations	195
8.15	4×16 mux2 Operations	196
8.16	4×16 mix Operations	198
8.17	Operation of the mix1.r Instruction	199
8.18	4×16 Pack Operation	200
8.19	Operation of the pack4.sss Instruction	201

8.20	4×16 Unpack Operations	201
8.21	Operation of the unpack4.l Instruction	202
9.1	Floating-point Register File	208
9.2	Floating-point Status Register Format	209
9.3	Floating-point Status Field Format	210
9.4	User Mask Format	215
9.5	Format of a Floating-point Register	216
9.6	Saving and Loading a Single-precision Floating-point Number from Memory	224
9.7	Saving and Loading a Double-precision Floating-point Number from Memory	226
9.8	Saving and Loading a Double-extended Precision Floating-point Number from Memory	226
9.9	Spilling the Content of a Floating-point Register to Memory	227
9.10	Saving and Loading an Integer Number in Floating-point Memory Format into a Floating-point Register	227
9.11	Optimizing a Recurrence Floating-point Computation	254
9.12	Floating-point Merge Operations	260
9.13	Parallel Floating-point Merge Operations	261
9.14	Floating-point Parallel Mix Operations	262
9.15	Floating-point Pack Operation	263
9.16	Floating-point Swap Operations	265
9.17	Floating-point Sign Extend Operations	266
10.1	Compatibility with IA-32 Applications	274
10.2	Instruction Set Transition Model	276
11.1	Compilation Process	284
11.2	Dynamic Profile-guided Compilation	286
11.3	Code Generator Compilation Order	289
11.4	Impact of Predicated Execution on Branch Prediction	292



List of Tables

2.1	Data Types and Sizes	20
2.2	Range for Unsigned and Signed Integer Numbers	20
3.1	Frame Marker Field Descriptions	39
3.2	Description of the Application Registers	41
3.3	RSC Field Descriptions	43
3.4	PFS Field Descriptions	47
3.5	User Mask Field Descriptions	48
3.6	CPUID Register 3 Field Descriptions	49
4.1	Register File Notation for Assembly Language Statements	55
4.2	Types of Data Dependencies	57
4.3	Relationship between Instruction Type and Execution Unit Type	61
4.4	Template Field Encoding and Instruction Slot Mapping	62
5.1	Load Instruction Formats	66
5.2	sz Completers	66
5.3	Load Type Completers	68
5.4	Supported Memory Load Instructions	69
5.5	Load and Store Hint Completers	69
5.6	Store Instruction Formats	76
5.7	Store Type Completers	76
5.8	Supported Memory Store Instructions	77
5.9	Speculation Check Instruction Formats	82
5.10	Semaphore Instruction Subgroup	88
5.11	Semaphore Type Completers	89
5.12	Register Transfer Instructions	92
5.13	Indirect Register File Mnemonics	93
5.14	Execution Unit Required to Access Application Registers	98
6.1	Integer Arithmetic Instructions	102

xvi *List of Tables*

6.2	Logical Instructions	106
6.3	Large Constant Generation Instructions	107
6.4	32-bit Address Pointer and 32-bit Integer Instructions	109
6.5	Available Values for the <i>xsz</i> Part of Mnemonic	109
6.6	Shift and Bit Field Instructions	112
6.7	Character String and Population Count Instructions	119
6.8	Result Ranges for <i>czx</i>	120
7.1	Compare Instruction Formats	124
7.2	Compare Relationship Completers for Normal and Unconditional Instruction Types	126
7.3	Comparison Type Completers	127
7.4	Compare Relationship Completers for Parallel Types	129
7.5	Compare Relationships Implemented as Pseudo-operations for Normal and Unconditional Types	130
7.6	Test Bit and Test NaT Bit Relationships for Normal and Unconditional Type Completers	133
7.7	Branch Instruction Formats	135
7.8	Branch Types Completers	135
7.9	Branch Whether Hint Completers	137
7.10	Sequential Prefetch Hint Completers	138
7.11	Branch Cache Deallocation Hint Completers	138
7.12	Register Stack Instructions	146
7.13	Phases of Execution	164
7.14	Pipelined Execution of the Loop Program	165
8.1	Parallel Arithmetic Instruction Formats	173
8.2	Saturation Completers	174
8.3	Parallel Saturation Limits	175
8.4	Parallel Average Completers	176
8.5	Parallel Comparison Relationships	182
8.6	Multiply Completers	185
8.7	Parallel Multiplication and Shift Right Count Options	186
8.8	Parallel Shift Instruction Formats	191
8.9	Data Arrangement Instruction Formats	194
8.10	mux Permutations for 8-Bit Elements	194
9.1	IEEE Real-data Type Ranges	204
9.2	IEEE Real-data Type Memory Formats	207
9.3	Trap Fields of the Floating-point Status Register	210
9.4	Status Flags of the FPSR Status Field	211
9.5	Control Fields of the FPSR Status Field	211
9.6	Floating-point Rounding Control Definitions	212
9.7	Floating-point Computational Model Control Definitions	212
9.8	Floating-point Status Register Instruction Formats	213
9.9	Status Field Completers	214
9.10	User Mask Floating-point Field Description	215

9.11	Floating-point Register Encoding	217
9.12	Floating-point Memory Access, Register Transfer, and Data Conversion Instruction Formats	223
9.13	Supported Floating-point Load and Store Instructions	224
9.14	fsz Completers	224
9.15	Floating-point Arithmetic Instructions that Employ One Source Operand	235
9.16	Floating-point Arithmetic Instructions that Employ Two Source Operands	237
9.17	Precision Control Completer	239
9.18	Floating-point Arithmetic Instructions that Employ Three Source Operands	241
9.19	Floating-point Arithmetic Instructions that Perform Approximations	244
9.20	Integer Multiply Instruction Formats	255
9.21	Floating-point Logic Instructions	257
9.22	Floating-point Data Arrangement Instruction Formats	259
9.23	Floating-point Compare Instructions	268
9.24	Floating-point Compare Relationships	269
9.25	Floating-point Comparison Types	269
9.26	Floating-point Class Relationships	270
9.27	Floating-point Classes (fclass9) Options	271
10.1	Itanium Processor Operating Environments	275
10.2	Invoke IA-32 Branch Instruction Format	277
10.3	Jump to Itanium Instruction Set	278
10.4	IA-32 Register Mapping	279