

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Basics</b>	<b>7</b>
2.1	Hardware Model . . . . .	7
2.1.1	Components . . . . .	7
2.1.2	Cycle Times . . . . .	9
2.1.3	Hierarchical Designs . . . . .	10
2.1.4	Notations for Delay Formulae . . . . .	10
2.2	Number Representations and Basic Circuits . . . . .	12
2.2.1	Natural Numbers . . . . .	12
2.2.2	Integers . . . . .	14
2.3	Basic Circuits . . . . .	17
2.3.1	Trivial Constructions . . . . .	17
2.3.2	Testing for Zero or Equality . . . . .	19
2.3.3	Decoders . . . . .	19
2.3.4	Leading Zero Counter . . . . .	21
2.4	Arithmetic Circuits . . . . .	22
2.4.1	Carry Chain Adders . . . . .	22
2.4.2	Conditional Sum Adders . . . . .	24
2.4.3	Parallel Prefix Computation . . . . .	27
2.4.4	Carry Lookahead Adders . . . . .	28
2.4.5	Arithmetic Units . . . . .	30
2.4.6	Shifter . . . . .	31

## Table of contents

2.5	Multipliers . . . . .	34
2.5.1	School Method . . . . .	34
2.5.2	Carry Save Adders . . . . .	35
2.5.3	Multiplication Arrays . . . . .	36
2.5.4	4/2-Trees . . . . .	37
2.5.5	Multipliers with Booth Recoding . . . . .	42
2.5.6	Cost and Delay of the Booth Multiplier . . . . .	47
2.6	Control Automata . . . . .	50
2.6.1	Finite State Transducers . . . . .	50
2.6.2	Coding the State . . . . .	51
2.6.3	Generating the Outputs . . . . .	51
2.6.4	Computing the Next State . . . . .	52
2.6.5	Moore Automata . . . . .	54
2.6.6	Precomputing the Control Signals . . . . .	55
2.6.7	Mealy Automata . . . . .	56
2.6.8	Interaction with the Data Paths . . . . .	58
2.7	Selected References and Further Reading . . . . .	61
2.8	Exercises . . . . .	61
<b>3</b>	<b>A Sequential DLX Design</b>	<b>63</b>
3.1	Instruction Set Architecture . . . . .	63
3.1.1	Instruction Formats . . . . .	64
3.1.2	Instruction Set Coding . . . . .	64
3.1.3	Memory Organization . . . . .	68
3.2	High Level Data Paths . . . . .	69
3.3	Environments . . . . .	71
3.3.1	General Purpose Register File . . . . .	71
3.3.2	Instruction Register Environment . . . . .	73
3.3.3	PC Environment . . . . .	74
3.3.4	ALU Environment . . . . .	75
3.3.5	Memory Environment . . . . .	78
3.3.6	Shifter Environment SHenv . . . . .	81
3.3.7	Shifter Environment SH4Lenv . . . . .	85
3.4	Sequential Control . . . . .	88
3.4.1	Sequential Control without Stalling . . . . .	88
3.4.2	Parameters of the Control Automaton . . . . .	95
3.4.3	A Simple Stall Engine . . . . .	97
3.5	Hardware Cost and Cycle Time . . . . .	99
3.5.1	Hardware Cost . . . . .	99
3.5.2	Cycle Time . . . . .	100
3.6	Selected References and Further Reading . . . . .	104

<b>4</b>	<b>Basic Pipelining</b>	<b>105</b>
4.1	Delayed Branch and Delayed PC . . . . .	107
4.2	Prepared Sequential Machines . . . . .	111
4.2.1	Prepared DLX Data Paths . . . . .	114
4.2.2	FSD for the Prepared Data Paths . . . . .	120
4.2.3	Precomputed Control . . . . .	122
4.2.4	A Basic Observation . . . . .	128
4.3	Pipelining as a Transformation . . . . .	130
4.3.1	Correctness . . . . .	131
4.3.2	Hardware Cost and Cycle Time . . . . .	139
4.4	Result Forwarding . . . . .	143
4.4.1	Valid Flags . . . . .	144
4.4.2	3-Stage Forwarding . . . . .	145
4.4.3	Correctness . . . . .	148
4.5	Hardware Interlock . . . . .	151
4.5.1	Stall Engine . . . . .	151
4.5.2	Scheduling Function . . . . .	154
4.5.3	Simulation Theorem . . . . .	157
4.6	Cost Performance Analysis . . . . .	159
4.6.1	Hardware Cost and Cycle Time . . . . .	159
4.6.2	Performance Model . . . . .	160
4.6.3	Delay Slots of Branch/Jump Instructions . . . . .	162
4.6.4	CPI Ratio of the DLX Designs . . . . .	163
4.6.5	Design Evaluation . . . . .	166
4.7	Selected References and Further Reading . . . . .	168
4.8	Exercises . . . . .	169
<b>5</b>	<b>Interrupt Handling</b>	<b>171</b>
5.1	Attempting a Rigorous Treatment of Interrupts . . . . .	171
5.2	Extended Instruction Set Architecture . . . . .	174
5.3	Interrupt Service Routines For Nested Interrupts . . . . .	177
5.4	Admissible Interrupt Service Routines . . . . .	180
5.4.1	Set of Constraints . . . . .	180
5.4.2	Bracket Structures . . . . .	181
5.4.3	Properties of Admissible Interrupt Service Routines . . . . .	182
5.5	Interrupt Hardware . . . . .	190
5.5.1	Environment PCenv . . . . .	191
5.5.2	Circuit Daddr . . . . .	193
5.5.3	Register File Environment RFenv . . . . .	194
5.5.4	Modified Data Paths . . . . .	198
5.5.5	Cause Environment CAenv . . . . .	202
5.5.6	Control Unit . . . . .	204

## Table of contents

5.6	Pipelined Interrupt Hardware . . . . .	214
5.6.1	PC Environment . . . . .	214
5.6.2	Forwarding and Interlocking . . . . .	216
5.6.3	Stall Engine . . . . .	220
5.6.4	Cost and Delay of the $DLX_{II}$ Hardware . . . . .	225
5.7	Correctness of the Interrupt Hardware . . . . .	227
5.8	Selected References and Further Reading . . . . .	235
5.9	Exercises . . . . .	236
<b>6</b>	<b>Memory System Design</b>	<b>239</b>
6.1	A Monolithic Memory Design . . . . .	239
6.1.1	The Limits of On-chip RAM . . . . .	240
6.1.2	A Synchronous Bus Protocol . . . . .	241
6.1.3	Sequential DLX with Off-Chip Main Memory . . . . .	245
6.2	The Memory Hierarchy . . . . .	253
6.2.1	The Principle of Locality . . . . .	254
6.2.2	The Principles of Caches . . . . .	255
6.2.3	Execution of Memory Transactions . . . . .	263
6.3	A Cache Design . . . . .	265
6.3.1	Design of a Direct Mapped Cache . . . . .	266
6.3.2	Design of a Set Associative Cache . . . . .	268
6.3.3	Design of a Cache Interface . . . . .	276
6.4	Sequential DLX with Cache Memory . . . . .	280
6.4.1	Changes in the DLX Design . . . . .	280
6.4.2	Variations of the Cache Design . . . . .	290
6.5	Pipelined DLX with Cache Memory . . . . .	299
6.5.1	Changes in the DLX Data Paths . . . . .	300
6.5.2	Memory Control . . . . .	304
6.5.3	Design Evaluation . . . . .	309
6.6	Selected References and Further Reading . . . . .	314
6.7	Exercises . . . . .	314
<b>7</b>	<b>IEEE Floating Point Standard and Theory of Rounding</b>	<b>317</b>
7.1	Number Formats . . . . .	317
7.1.1	Binary Fractions . . . . .	317
7.1.2	Two's Complement Fractions . . . . .	318
7.1.3	Biased Integer Format . . . . .	318
7.1.4	IEEE Floating Point Numbers . . . . .	320
7.1.5	Geometry of Representable Numbers . . . . .	321
7.1.6	Convention on Notation . . . . .	322
7.2	Rounding . . . . .	323
7.2.1	Rounding Modes . . . . .	323

7.2.2	Two Central Concepts . . . . .	325
7.2.3	Factorings and Normalization Shifts . . . . .	325
7.2.4	Algebra of Rounding and Sticky Bits . . . . .	326
7.2.5	Rounding with Unlimited Exponent Range . . . . .	330
7.2.6	Decomposition Theorem for Rounding . . . . .	330
7.2.7	Rounding Algorithms . . . . .	335
7.3	Exceptions . . . . .	335
7.3.1	Overflow . . . . .	336
7.3.2	Underflow . . . . .	336
7.3.3	Wrapped Exponents . . . . .	338
7.3.4	Inexact Result . . . . .	341
7.4	Arithmetic on Special Operands . . . . .	341
7.4.1	Operations with NaNs . . . . .	342
7.4.2	Addition and Subtraction . . . . .	343
7.4.3	Multiplication . . . . .	344
7.4.4	Division . . . . .	344
7.4.5	Comparison . . . . .	345
7.4.6	Format Conversions . . . . .	347
7.5	Selected References and Further Reading . . . . .	349
7.6	Exercises . . . . .	349
<b>8</b>	<b>Floating Point Algorithms and Data Paths</b>	<b>351</b>
8.1	Unpacking . . . . .	354
8.2	Addition and Subtraction . . . . .	359
8.2.1	Addition Algorithm . . . . .	359
8.2.2	Adder Circuitry . . . . .	360
8.3	Multiplication and Division . . . . .	372
8.3.1	Newton-Raphson Iteration . . . . .	373
8.3.2	Initial Approximation . . . . .	375
8.3.3	Newton-Raphson Iteration with Finite Precision . . . . .	377
8.3.4	Table Size versus Number of Iterations . . . . .	379
8.3.5	Computing the Representative of the Quotient . . . . .	380
8.3.6	Multiplier and Divider Circuits . . . . .	381
8.4	Floating Point Rounder . . . . .	390
8.4.1	Specification and Overview . . . . .	391
8.4.2	Normalization Shift . . . . .	394
8.4.3	Selection of the Representative . . . . .	405
8.4.4	Significand Rounding . . . . .	406
8.4.5	Post Normalization . . . . .	407
8.4.6	Exponent Adjustment . . . . .	408
8.4.7	Exponent Rounding . . . . .	409
8.4.8	Circuit SPECFPRND . . . . .	410

## Table of contents

8.5	Circuit FCon . . . . .	412
8.5.1	Floating Point Condition Test . . . . .	414
8.5.2	Absolute Value and Negation . . . . .	417
8.5.3	IEEE Floating Point Exceptions . . . . .	418
8.6	Format Conversion . . . . .	418
8.6.1	Specification of the Conversions . . . . .	419
8.6.2	Implementation of the Conversions . . . . .	423
8.7	Evaluation of the FPU Design . . . . .	432
8.8	Selected References and Further Reading . . . . .	435
8.9	Exercises . . . . .	436
<b>9</b>	<b>Pipelined DLX Machine with Floating Point Core</b>	<b>439</b>
9.1	Extended Instruction Set Architecture . . . . .	441
9.1.1	FPU Register Set . . . . .	441
9.1.2	Interrupt Causes . . . . .	443
9.1.3	FPU Instruction Set . . . . .	444
9.2	Data Paths without Forwarding . . . . .	445
9.2.1	Instruction Decode . . . . .	448
9.2.2	Memory Stage . . . . .	451
9.2.3	Write Back Stage . . . . .	455
9.2.4	Execute Stage . . . . .	461
9.3	Control of the Prepared Sequential Design . . . . .	470
9.3.1	Precomputed Control without Division . . . . .	474
9.3.2	Supporting Divisions . . . . .	479
9.4	Pipelined DLX Design with FPU . . . . .	485
9.4.1	PC Environment . . . . .	485
9.4.2	Forwarding and Interlocking . . . . .	486
9.4.3	Stall Engine . . . . .	498
9.4.4	Cost and Delay of the Control . . . . .	503
9.4.5	Simulation Theorem . . . . .	507
9.5	Evaluation . . . . .	508
9.5.1	Hardware Cost and Cycle Time . . . . .	508
9.5.2	Variation of the Cache Size . . . . .	511
9.6	Exercises . . . . .	516
<b>A</b>	<b>DLX Instruction Set Architecture</b>	<b>519</b>
A.1	DLX Fixed-Point Core: FXU . . . . .	519
A.1.1	Instruction Formats . . . . .	520
A.1.2	Instruction Set Coding . . . . .	521
A.2	Floating-Point Extension . . . . .	521
A.2.1	FPU Register Set . . . . .	521
A.2.2	FPU Instruction Set . . . . .	522

<b>B Specification of the FDLX Design</b>	<b>527</b>
B.1 RTL Instructions of the FDLX . . . . .	527
B.1.1 Stage IF . . . . .	527
B.1.2 Stage ID . . . . .	527
B.1.3 Stage EX . . . . .	529
B.1.4 Stage M . . . . .	532
B.1.5 Stage WB . . . . .	534
B.2 Control Automata of the FDLX Design . . . . .	534
B.2.1 Automaton Controlling Stage ID . . . . .	535
B.2.2 Precomputed Control . . . . .	536
 <b>Bibliography</b>	 <b>543</b>
 <b>Index</b>	 <b>549</b>