

Contents

1	Introduction and Motivation	1
1.1	The Growing Reliance on FPGAs	1
1.1.1	FPGAs for Aerospace	2
1.1.2	FPGAs for Supercomputing	4
1.1.3	FPGAs for Video Analysis	5
1.1.4	FPGAs for High-Throughput Cryptography	5
1.1.5	FPGAs for Intrusion Detection and Prevention	6
1.2	FPGA Architectures	6
1.2.1	The Attractiveness of Reconfigurable Hardware	7
1.2.2	The Internals of an FPGA	8
1.2.3	Design Flow	13
1.3	The Many Facets of FPGA Security	16
1.3.1	Security Is Hard	17
1.3.2	Complexity and Abstraction	18
1.3.3	Baked in Versus Tacked on	19
1.3.4	Separation of FPGA Cores	20
1.4	Organization of This Book	21
	References	22
2	High Assurance Software Lessons and Techniques	27
2.1	Background	27
2.2	Malicious Software	27
2.2.1	Trojan Horses	28
2.2.2	Subversion	29
2.3	Assurance	30
2.4	Commensurate Protection	31
2.4.1	Threat Model	32
2.5	Security Policy Enforcement	34
2.5.1	Types of Policies	34
2.5.2	Policy Enforcement Mechanisms	39
2.5.3	Composition of Trusted Components	50

2.6	Assurance of Policy Enforcement	51
2.6.1	Life Cycle Support	52
2.6.2	Configuration Management	55
2.6.3	Independent Assessment	56
2.6.4	Dynamic Program Analysis	58
2.6.5	Trusted Distribution	60
2.6.6	Trusted Recovery	61
2.6.7	Static Analysis of Program Specifications	62
	References	65
3	Hardware Security Challenges	71
3.1	Malicious Hardware	71
3.1.1	Categories of Malicious Hardware	71
3.1.2	Foundry Trust	72
3.1.3	Physical Attacks	74
3.2	Covert Channel Definition	75
3.2.1	The Process Abstraction	76
3.2.2	Equivalence Classes	76
3.2.3	Formal Definition	76
3.2.4	Synchronization	77
3.2.5	Shared Resources	77
3.2.6	Requirements	77
3.2.7	Bypass	78
3.3	Existing Approaches to Limiting Covert and Side Channel Attacks	78
3.3.1	Shared Resource Matrix Methodology	78
3.3.2	Cache Interference	79
3.3.3	FPGA Masking Schemes	79
3.4	Detecting and Mitigating Covert Channels on FPGAs	80
3.4.1	Design Flows	80
3.4.2	Spatial Isolation	80
3.4.3	Memory Protection	81
3.5	Policy State as a Covert Storage Channel	81
3.5.1	Stateful Policies	81
3.5.2	Covert Channel Mechanism	81
3.5.3	Encoding Schemes	82
3.5.4	Covert Storage Channel Detection	83
3.5.5	Covert Channel Mitigation	83
	References	84
4	FPGA Updates and Programmability	87
4.1	Introduction	87
4.2	Bitstream Encryption and Authentication	87
4.2.1	Key Management	88
4.2.2	Defeating Bitstream Encryption	89
4.3	Remote Updates	90

- 4.3.1 Authentication 90
- 4.3.2 Trusted Recovery 91
- 4.4 Partial Reconfiguration 91
 - 4.4.1 Applications of Partial Reconfiguration 91
 - 4.4.2 Hot-Swappable vs. Stop-the-World 92
 - 4.4.3 Internal Configuration Access Port 92
 - 4.4.4 Dynamic Security and Complexity 92
 - 4.4.5 Object Reuse 93
 - 4.4.6 Integrity Verification 94
 - References 95
- 5 Memory Protection on FPGAs 97**
 - 5.1 Overview 97
 - 5.2 Memory Protection on FPGAs 98
 - 5.3 Policy Description and Synthesis 99
 - 5.3.1 Memory Access Policy 99
 - 5.3.2 Hardware Synthesis 102
 - 5.4 A Higher-Level Specification Language 104
 - 5.5 Example Policies 106
 - 5.5.1 Controlled Sharing 106
 - 5.5.2 Access List 108
 - 5.5.3 Chinese Wall 109
 - 5.5.4 Bell and LaPadula Confidentiality Model 110
 - 5.5.5 High Water Mark 111
 - 5.5.6 Biba Integrity Model 112
 - 5.5.7 Redaction 113
 - 5.6 System Architecture 116
 - 5.7 Evaluation 116
 - 5.8 Using the Policy Compiler 117
 - 5.9 Constructing Mathematically Precise Policies 120
 - 5.9.1 Cross Product Method 120
 - 5.9.2 Examples 121
 - 5.9.3 Monotonic Policy Changes 123
 - 5.9.4 Formal Aspects of Hybrid Policies 124
 - 5.10 Summary 125
 - References 125
- 6 Spatial Separation with Moats 127**
 - 6.1 Overview 127
 - 6.2 Separation 128
 - 6.3 Physical Isolation with Moats 128
 - 6.4 Constructing Moats 128
 - 6.4.1 The Gap Method 129
 - 6.4.2 The Inspection Method 130
 - 6.4.3 Comparing the Gap and Inspection Methods 130

- 6.5 Secure Interconnect with Drawbridges 132
 - 6.5.1 Drawbridges for Direct Connections 132
 - 6.5.2 Route Tracing with Partial Reconfiguration 135
 - 6.5.3 Drawbridges for Shared Bus Architectures 135
- 6.6 Protecting the Reference Monitor with Moats 137
 - References 138
- 7 Putting It All Together: A Design Example 139**
 - 7.1 A Multi-Core Reconfigurable Embedded System 139
 - 7.2 On-Chip Peripheral Bus 140
 - 7.3 AES core 141
 - 7.4 Logical Isolation Compartments 141
 - 7.5 Reference Monitor 141
 - 7.6 Stateful Policy 142
 - 7.7 Secure Interconnect Scalability 145
 - 7.8 Covert Channels 145
 - 7.9 Incorporating Moats and Drawbridges 146
 - 7.10 Implementation and Evaluation 147
 - 7.11 Software Interface 148
 - 7.12 Security Usability 148
 - 7.13 More Example Security Architectures 148
 - 7.13.1 Classes of Designs 148
 - 7.13.2 Topologies 150
 - 7.14 Summary 151
 - References 152
- 8 Forward-Looking Problems 153**
 - 8.1 Trustworthy Tools 153
 - 8.2 Formal Verification of Secure Systems 154
 - 8.3 Security Usability 155
 - 8.4 Hardware Trust 155
 - 8.5 Languages 155
 - 8.6 Configuration Management 156
 - 8.7 Securing the Supply Chain 156
 - 8.8 Physical Attacks on FPGAs 157
 - 8.9 Design Theft and Failure Analysis 157
 - 8.10 Partial Reconfiguration and Dynamic Security 158
 - 8.11 Concluding Remarks 158
 - References 160
- A Computer Architecture Fundamentals 161**
 - A.1 What Do Computer Architects Do All Day? 161
 - A.2 Tradeoffs Between CPUs, FPGAs, and ASICs 162
 - A.3 Computer Architecture and Computer Science 163
 - A.4 Program Analysis 164
 - A.4.1 The Science of Processor Simulation 164

- A.4.2 On-Chip Profiling Engines 165
- A.4.3 Binary Instrumentation 166
- A.4.4 Phase Classification 167
- A.5 Novel Computer Architectures 168
 - A.5.1 The DIVA Architecture 168
 - A.5.2 The Raw Microprocessor 169
 - A.5.3 The WaveScalar Architecture 169
 - A.5.4 Architectures for Medicine 169
- A.6 Memory 170
- A.7 Superscalar Processors 173
- A.8 Multithreading 174
- References 175