

# Index

## A

Analog abstraction, 94  
Analog circuits, 89–103  
Analog faults, 90–92, 94, 95, 98, 101, 103  
Annotation, 68, 113, 114, 123, 125, 126  
Attributes, 16, 109, 112–115, 123–127  
Automatic code generation, 18  
Automatic update, 6

## B

Behavioral level, 90, 96, 99  
Breadth-first search, 72, 78

## C

C++, 2, 6, 7, 11, 13, 15–17, 19, 26, 33, 84, 85, 95, 98, 101–103  
Circuit level, 90, 94  
Compilation, 50, 53, 58–65, 67–69  
Concurrency, 47, 86, 90  
Consistency, 107–127  
Constrained random (CR), 27, 28, 39, 42  
Constrained random generator (CRG), 28, 30, 33–34  
Correlation, 92, 109, 112–117  
Co-simulation, 2, 3, 5, 6, 13, 15, 19, 20, 26, 30, 42  
CR, *see* Constrained random (CR)  
CRG, *see* Constrained random generator (CRG)  
Cyber-physical systems simulation, 2, 3, 21

## D

Dataflow languages, 45–69  
DBM, *see* Difference bound matrices (DBM)  
Design automation, 108, 127  
Design flow, 3, 26, 108, 109, 117, 126  
Design restructuring, 125, 126  
Difference bound matrices (DBM), 50–54, 58, 60, 63, 67, 69  
Domain-specific language, 33, 73, 85  
Drug discovery, 74, 76, 77, 82–85  
Duty cycle, 36–38

## E

Electronic Design Automation (EDA), 108, 111, 112, 125  
Electronic System Level (ESL), 26–28  
Event-driven

## F

Fault analysis, 89–103  
Fault injection, 91, 98, 99, 103  
Fault models, 91, 92, 94–98, 103  
Fault taxonomy, 94–98  
Field programmable gate arrays (FPGA), 72, 73, 77–81, 83–86  
File exchange, 7, 8  
Firmware, 25–42  
FPGA, *see* Field programmable gate arrays (FPGA)  
FMI, *see* Functional Mockup Interface (FMI)  
Functional level, 90, 92, 98, 103  
Functional Mockup Interface (FMI), 1–21

Functional programming, 73, 75, 77  
 Functional safety, 90  
 Functor, 75, 77

**G**

Graph, 7, 19, 72–82, 85, 86, 98, 99, 109, 110, 118  
 Graph processing, 71–86  
 Graph transformation, 72–77, 82, 86  
 Graph traversal, 78–80

**H**

Hardware acceleration, 71–86  
 Hardware Description Languages (HDLs), 1–21, 90, 108–110, 112–115, 117, 122, 124–127  
 Hardware synthesis, 73  
 Hybrid systems, 67, 68

**I**

Instruction blocks (IBs), 30–33  
 Interconnect, 78, 84, 85, 119  
 ISO 26262, 90

**L**

Language manipulation, 89–103  
 Low-power, 26, 110, 126  
 Lustre, 45, 46, 49, 67, 68

**M**

Mathworks Simulink, 3, 6, 7, 20  
 Memory bottleneck, 77, 86  
 Methodology, 1–3, 5, 7–20, 27, 73, 90, 91, 95, 98, 103, 107–127  
 Monad, 75, 77

**N**

Network science, 72

**P**

Performance, 2, 3, 19–21, 26, 28–30, 36, 41, 42, 78, 83–85, 99, 103  
 Power-aware, 26, 27, 30, 42, 107–127  
 Power consumption, 26, 28–30, 34, 37, 40–42, 84, 108  
 Power-intent, 108–112, 114, 115, 117, 118, 122–127  
 Power management, 25–42, 108, 112, 119  
 Power modeling, 26, 27, 34, 35

Protein-interaction networks, 73, 75, 76, 79, 82, 84, 85

**R**

Register Transfer Level (RTL), 2, 7, 8, 13, 18, 26, 114, 117, 119, 126, 127

**S**

Safety, 68, 90  
 SETs, *see* Single Event Transients (SETs)  
 Shortest path, 72, 77–81, 83, 85  
 Signal-flow graph, 99  
 Simulation, 2, 3, 5–7, 9, 10, 17, 19–21, 26, 27, 34, 35, 41, 42, 45–69, 90–92, 97–99, 101–103, 108  
 Single Event Transients (SETs), 97  
 Smart systems, 2, 18, 86  
 SMT, 34, 67  
 SoCRocket, 27, 34–42  
 SOCs, *see* System-on-Chips (SoCs)  
 Structural, 92, 108–113, 115, 116, 120, 122  
 Symbolic simulation, 45–69  
 Synchronous languages, 45–69  
 SystemC, 26, 27, 33, 41, 42  
 System-on-Chips (SoCs), 26, 27, 107, 108, 114, 119, 120, 122

**T**

Timed automata, 46, 49, 67  
 Timers, 45–69  
 Transaction level modeling (TLM), 26, 27  
 Type system, 49, 50, 54, 62, 69

**U**

Unified power format (UPF), 27, 108, 109, 111–113, 115, 117–119, 122–127

**V**

Validation, 3, 25–42, 108, 109, 117–125  
 Verification, 26, 27, 46, 67, 68, 108  
 Verilog-AMS, 90, 91, 94–96, 98, 102, 103  
 Virtual platforms, 20, 21  
 Virtual prototype (VP), 25–42

**W**

Workload, 26–33, 39, 41, 42

**Z**

Zélus, 46, 47, 53–55, 58, 60, 63–65, 67–69