References


References


References

[140]


Validation, verification and certification of embedded systems. Technical report, NATO Research and Technology organization, 2005.

TTEthernet - A Powerful Network Solution for All Purposes, 2007. TTTech Computertechnik AG.


Index

A
Accuracy, 5
Always happen in parallel, 124

B
Behavioral domain, see Y-Chart

C
Combined conflict prediction table, 91
Computer aided design, 6
ConcurrenC model of computation, 32
Cooperative multithreading in systemC, 41
Creating parallel system models, 108
Current time advance table, 69

D
Data conflict table, 67
Data conflict table for predictions, 87
Discrete event simulation, 13, 39
  scheduler, 14
  simulation cycles, 13
  working threads, 13
Discrete event simulation data structures and operations, 39
Dynamic race condition checking, 115

E
Electronic system-level (ESL) design, 8
Embedded computer systems, 1
Embedded hardware system, 1, 4
Embedded operating systems, 2
Embedded software system, 2, 4
Estimation, 10

F
Event notification table, 68
Event notification table for predictions, 89
False conflicts, 75

H
Hardware description languages (HDLs), 10
Hardware-depend software (HdS), 2
Host-compiled simulation, 27

I
Implementation models, 10
In-order discrete event execution, 57
Instance isolation, 77
International technology roadmap for semiconductors (ITRS), 4

L
Levels of abstraction, 4

M
May happen in parallel, 120
Models, 4
Models of computation (MoCs), 30
  dataflow graph (DFG), 31
  dataflow process network (DFPN), 31
  finite state machine with datapath (FSMD), 31
  kahn process network (KPN), 31
  petri net, 31
  program state machine (PSM), 31
software/hardware integration medium (SHIM), 31
synchronous dataflow (SDF), 31
transaction-level modeling (TLM), 31
Modified segment adjacency matrix, 88
Moore’s law, 3
Multicore technology, 25
Multithreaded programming, 26

N
Next time advance table, 69

O
On-the-fly instance isolation, 80
Optimized scheduling using predictions, 85
data hazard prediction, 87
event hazard prediction, 89
time hazard prediction, 89
Optimized static code analysis, 81
Out-of-order parallel discrete event simulation, 59
dynamic conflict detection, 70
notations, 59
simulation event tuple, 59
simulation invariants, 60
simulation queues, 59
simulation state transition, 60
simulation states, 60
simulation time tuple, 59
scheduling algorithm, 62
static conflict analysis, 62

P
Parallel benchmark models, 96
parallel fibonacci calculation, 97
parallel fibonacci calculation with timing information, 97
parallel floating-point multiplications, 96
Parallel discrete event simulation (PDES), 27
conservative PDES, 27
distributed parallel simulation, 27
optimistic PDES, 27
Physical domain, see Y-Chart
Pre-emptive multithreading in specC, 42

R
Race condition diagnosis, 112
Real-time operating system (RTOS), 2
Recoding, 108
Refinement, 9
architecture refinement, 9
backend tasks, 10
communication refinement, 9
network refinement, 9
scheduling refinement, 9
Register-transfer level (RTL) design, 29

S
Segment, 62
Segment adjacency matrix, 87
Segment boundary, 62
Segment graph, 63
Shared variables and race conditions, 109, 114
Simulation, 8, 12
continuous simulation, 12
discrete event simulation, 12
monte carlo simulation, 12
SLDL multithreading semantics, 41
Software languages, 11
SpecC language, 17
behavioral hierarchy, 18
concurrent execution, 18
finite state machine (FSM) execution, 19
parallel execution, 19
pipelined execution, 19
sequential execution, 18, 19
communication, 20
exception handling, 20
abortion, 20
interrupt, 20
exporter, 22
importer, 22
library support, 21
persistent annotation, 21
refinement tools, 22
specC internal representation (SIR), 22
structural hierarchy, 17
synchronization, 20
timing, 20
exact timing, 20
timing constraints, 21
types, 18
Specification, 8
Static conflict analysis, 67
data hazards, 67
event hazards, 68
SLDL variables, 68
time hazards, 69
direct timing hazard, 69
indirect timing hazard, 69
Static segment aware detection, 120
Structural domain, see Y-Chart
Synchronous parallel discrete event simulation, 43
channel locking scheme, 45
code instrumentation for communication protection, 47
multicore simulation implementation optimization, 49
protecting communication, 45
protecting scheduling resources, 44
synchronization for multicore parallel simulation, 44
System design challenge, 3
System-level description languages (SDLs), 11
System-level design, 3, 4
System-level design methodologies, 6
  bottom-up design, 6
  meet-in-the-middle design, 7
top-down design, 6
System-level model of a DVD player, 71
System-level model of a H.264 video decoder, 35, 50, 73, 93, 99, 116, 128
System-level model of a H.264 video encoder, 95, 100, 117, 128
System-level model of a JPEG image encoder, 55, 72, 99, 128
System-level model of a mandelbrot graphics renderer, 128
System-level model of a MP3 audio decoder, 100, 128
System-level model of a video edge detector, 92, 99
System-level model of an MP3 audio decoder, 128
System-on-chip (SoC), 2
System-on-chip environment (SCE), 22
  architecture exploration, 23
  communication synthesis, 25
  network exploration, 24
  RTL synthesis, 25
  scheduling exploration, 23
  software synthesis, 25
SystemC language, 22

T
Temporal decoupling, 27
Time advance table for predictions, 89
Transaction-level modeling, 27

V
Validation, 8
Verification, 8

W
Well-defined system model, 107

Y
Y-Chart, 5