Introduction to Discrete Event Simulation II

John Colley

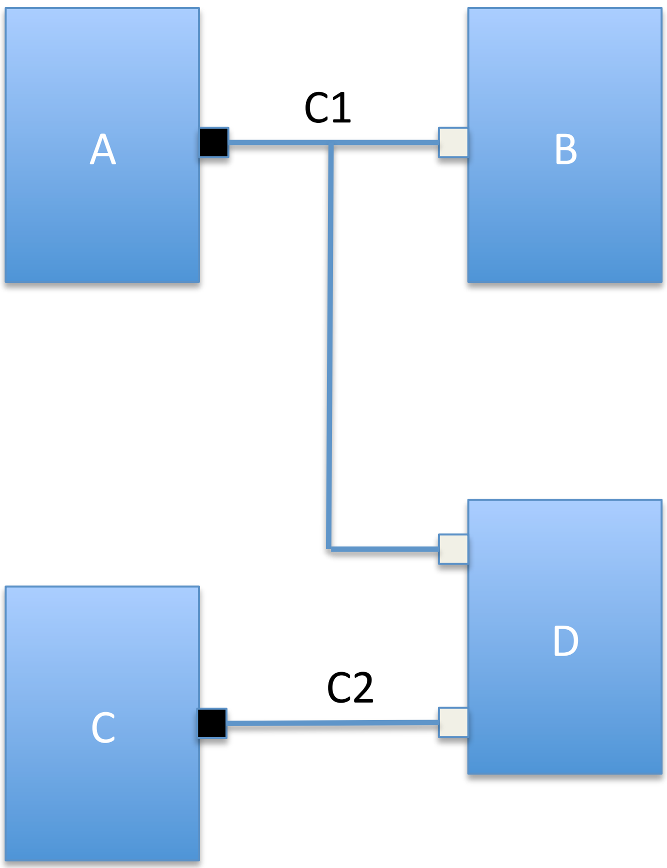
13th February 2012

Southampton

Introduction

- Discrete Event Simulation Principles – Review
- The Testbench
- Simulation-based Verification
- Sign Off
- Summary

Discrete Event Simulation



COMPONENT VIEW

Components: A, B, C, D (processes)

Connections: C1, C2 (unidirectional)

Ports: IN ☐ OUT ■

SIMULATOR API

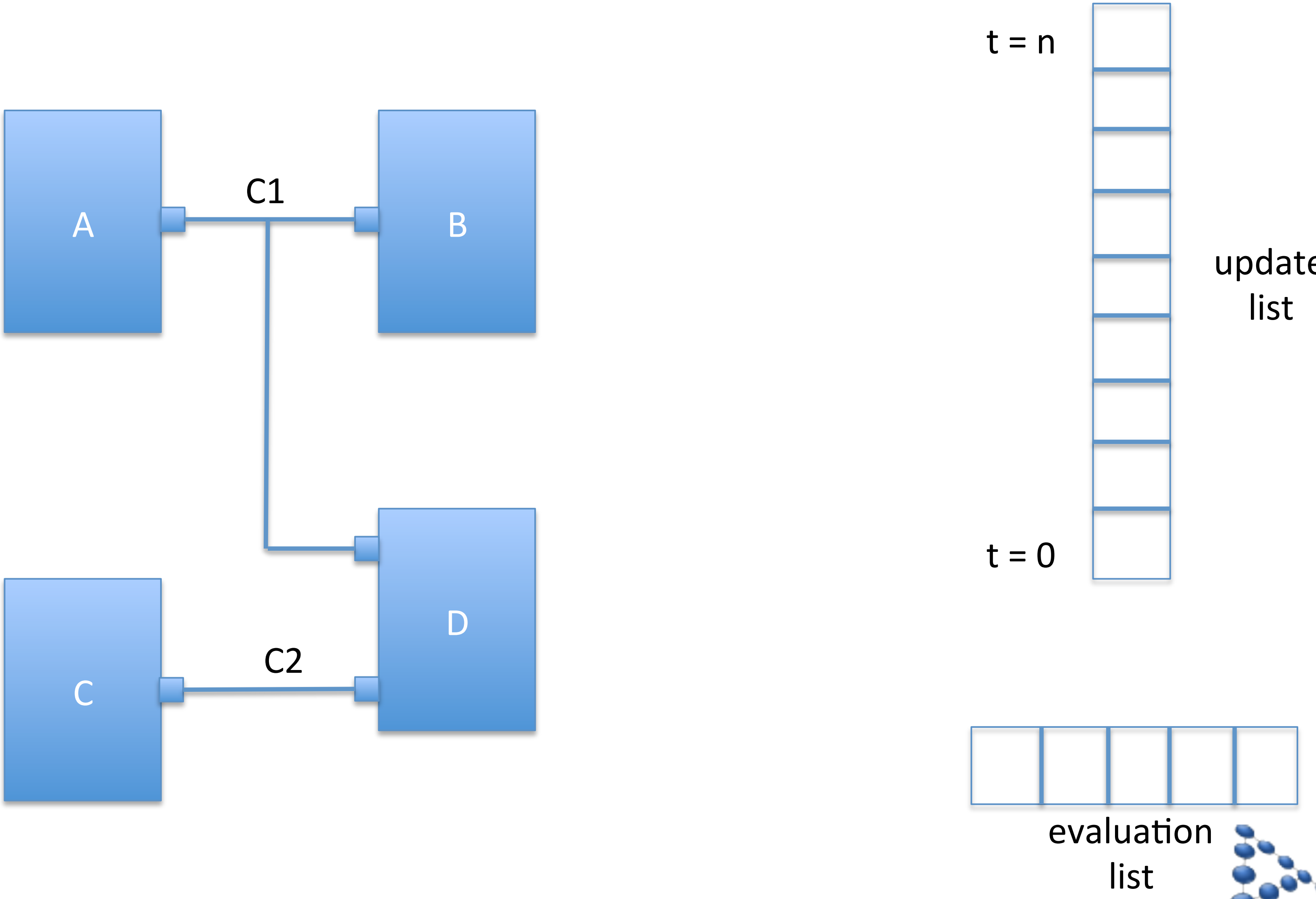
GetValue(port)

HasChanged(port)

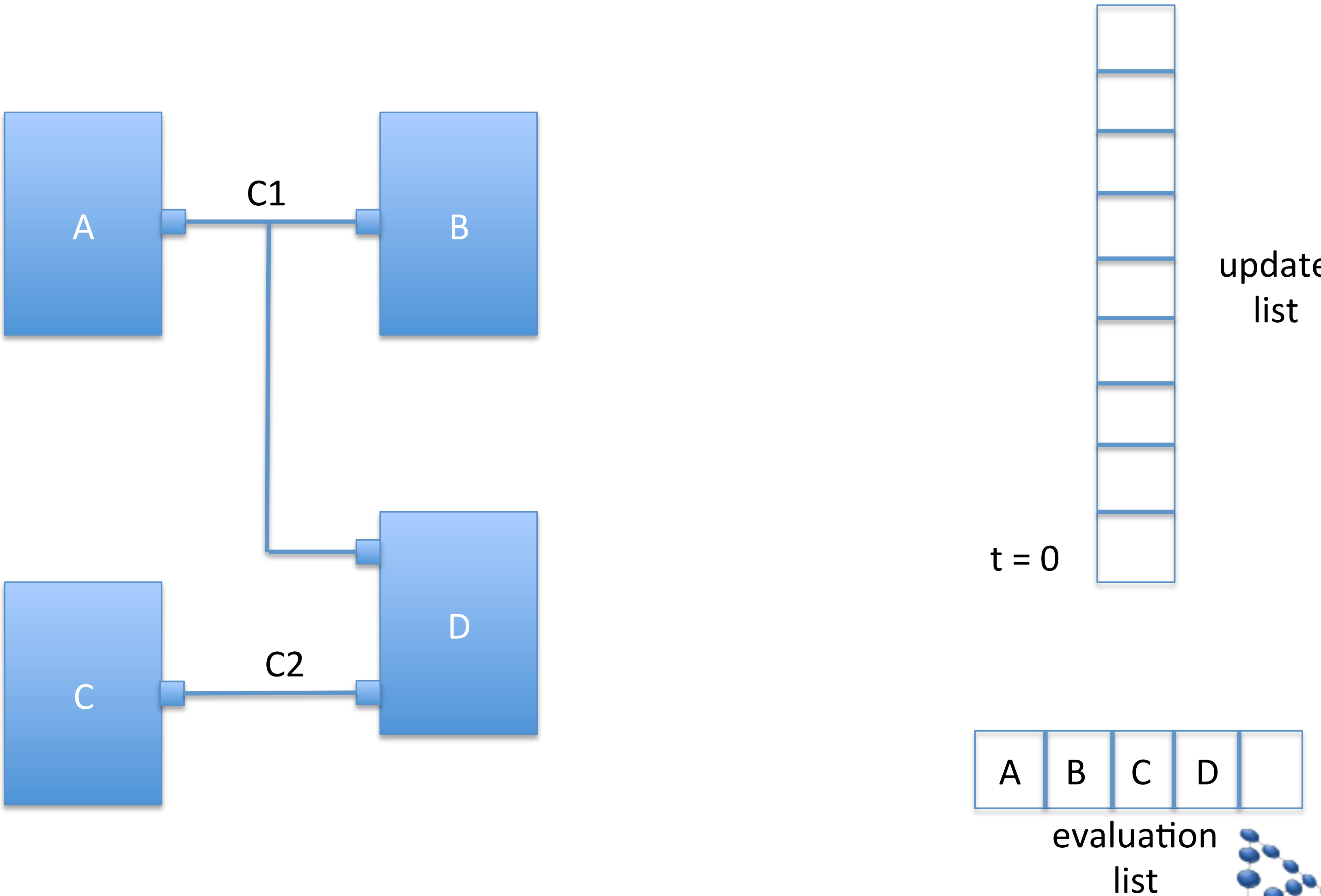
SetValue(OUT port, val, delay)

ScheduleEval(component, delay)

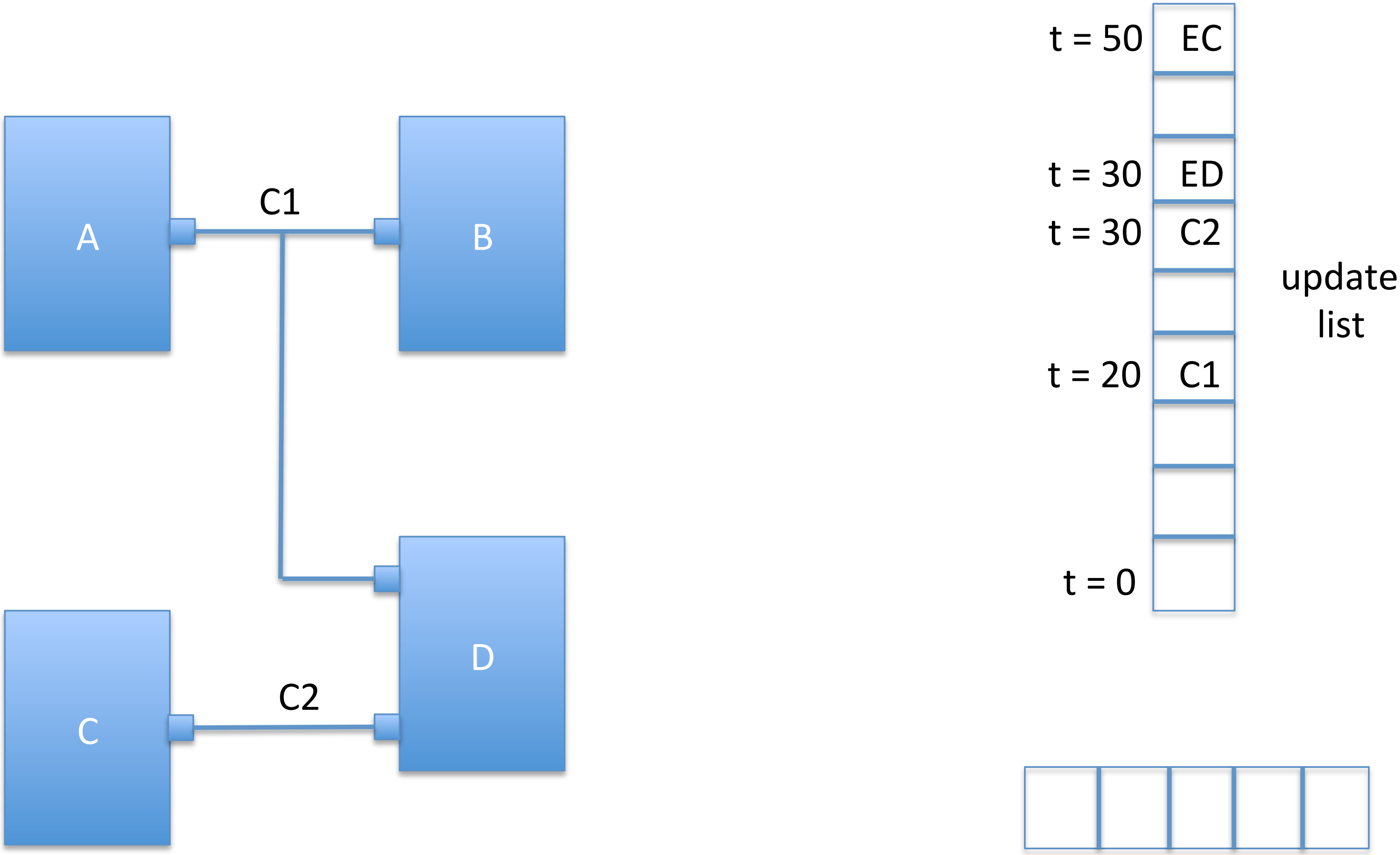
The Two-list Simulation Algorithm



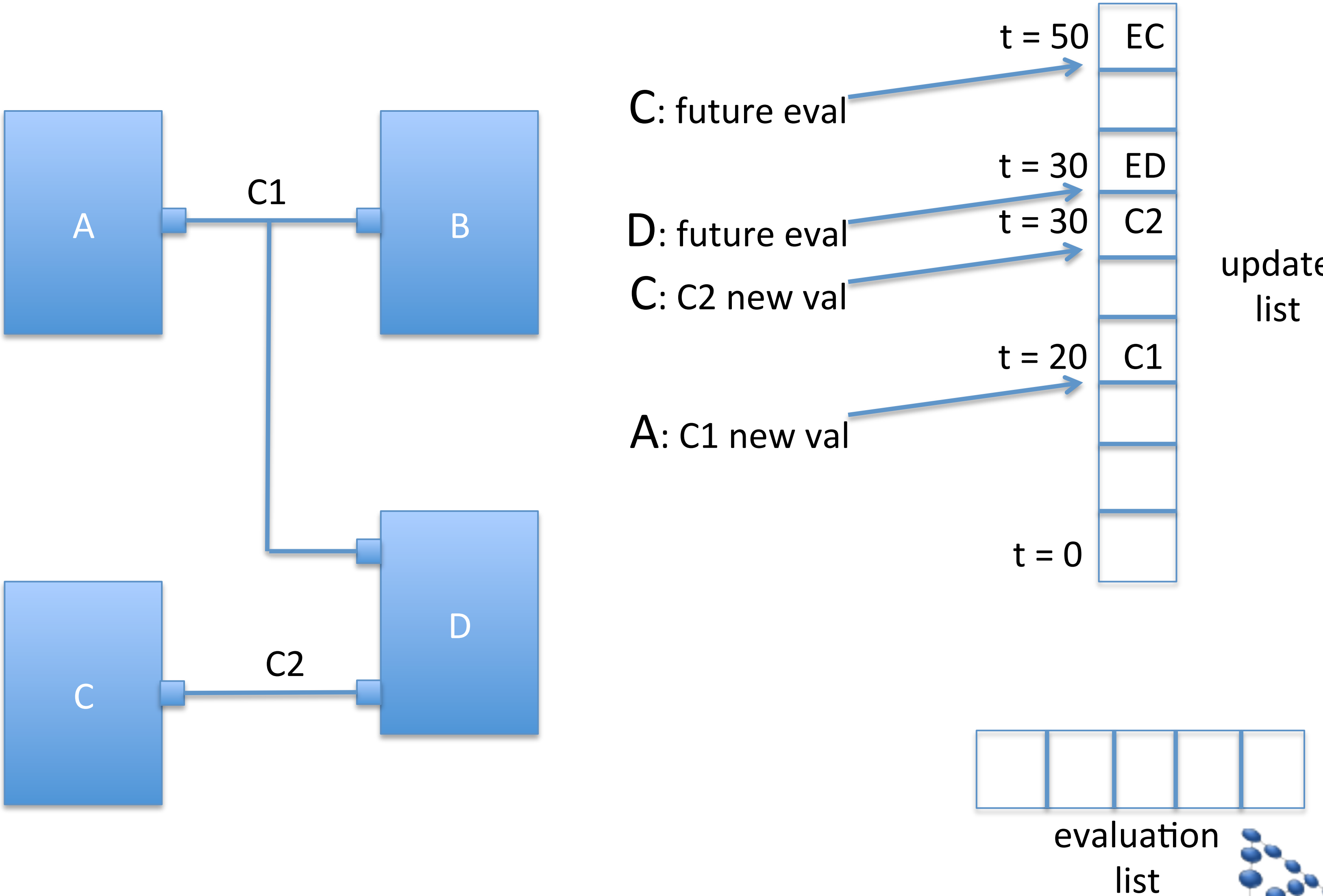
Time Zero Initialisation: Evaluate all Components



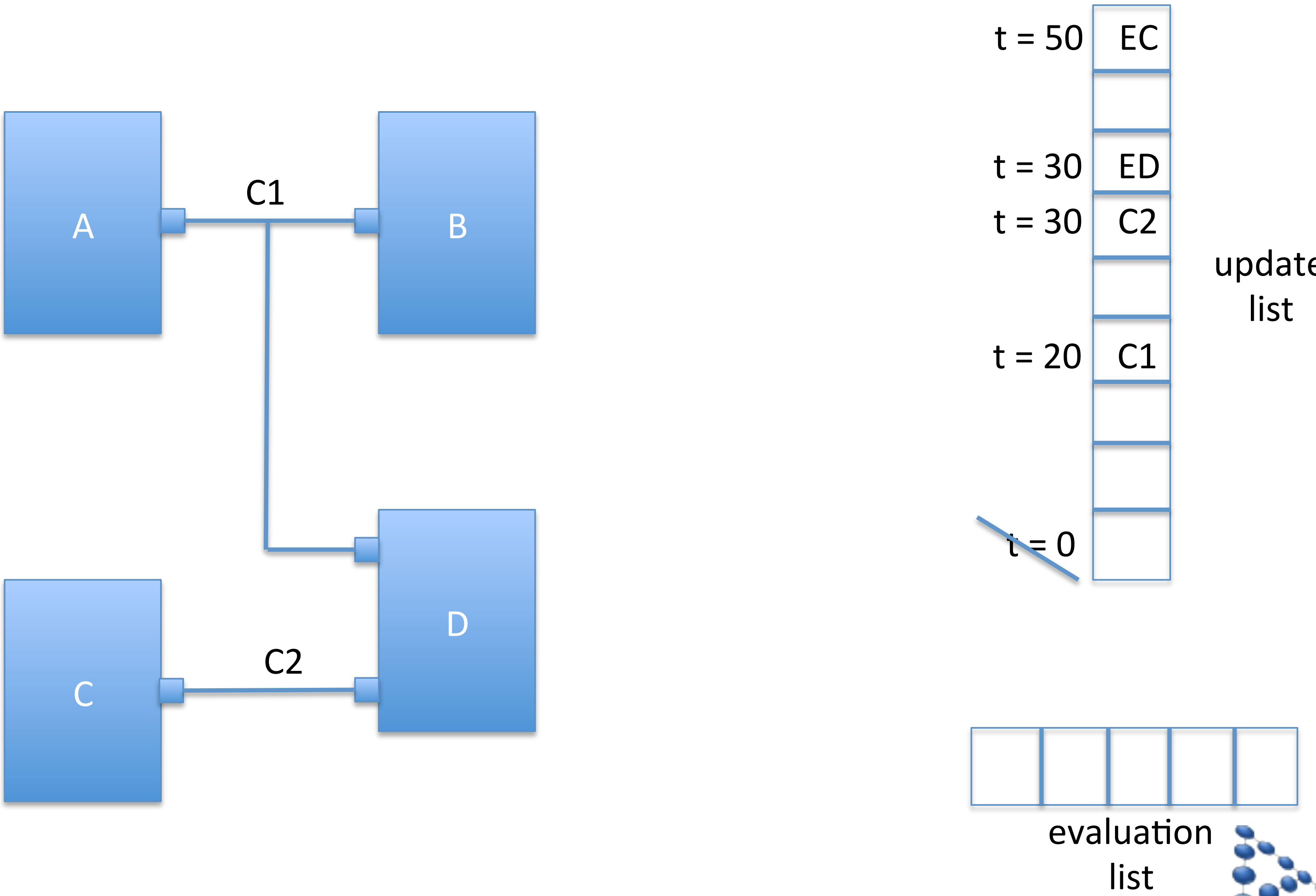
Component evaluations call *SetValue*, *ScheduleEval*



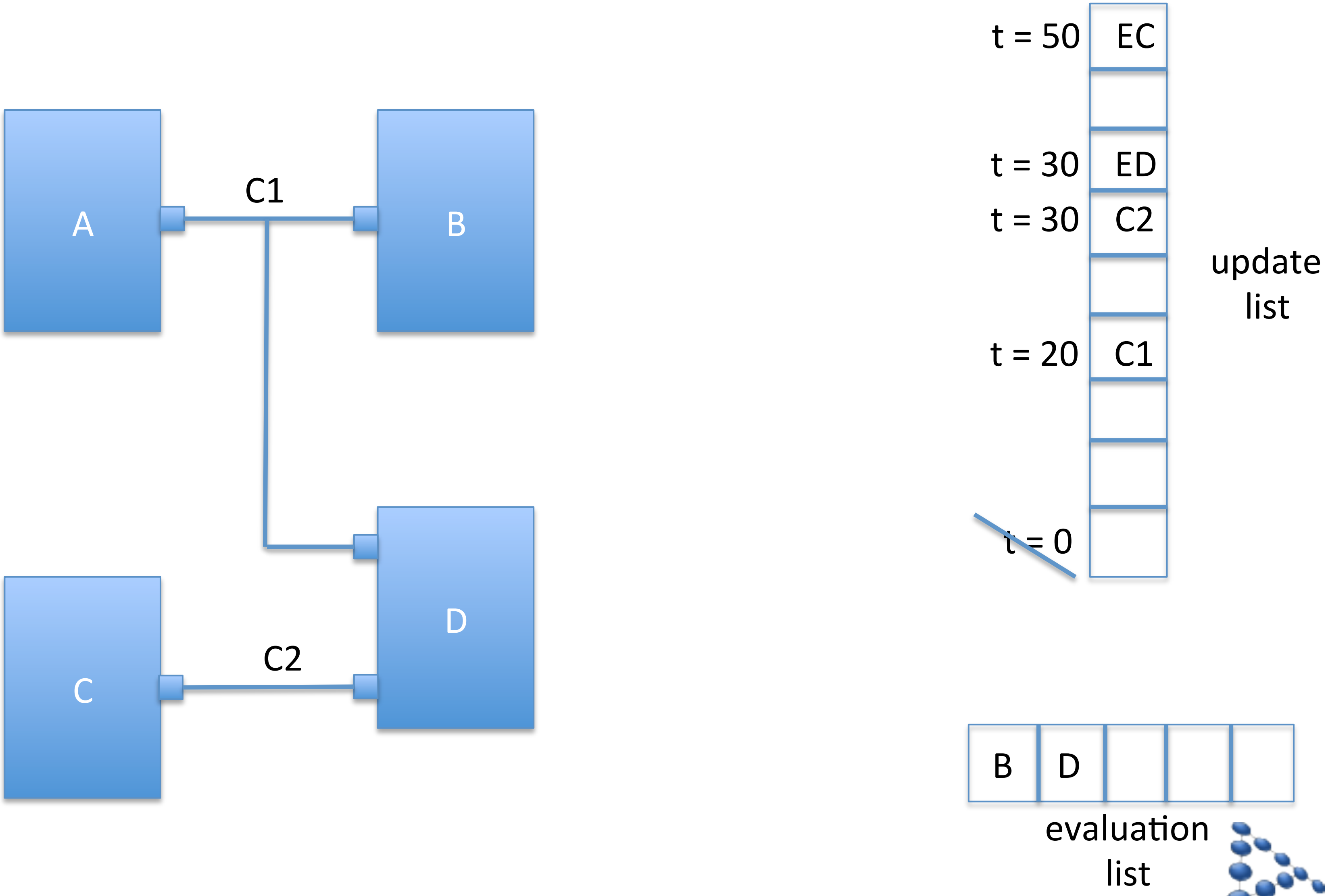
Component evaluations call *SetValue*, *ScheduleEval*



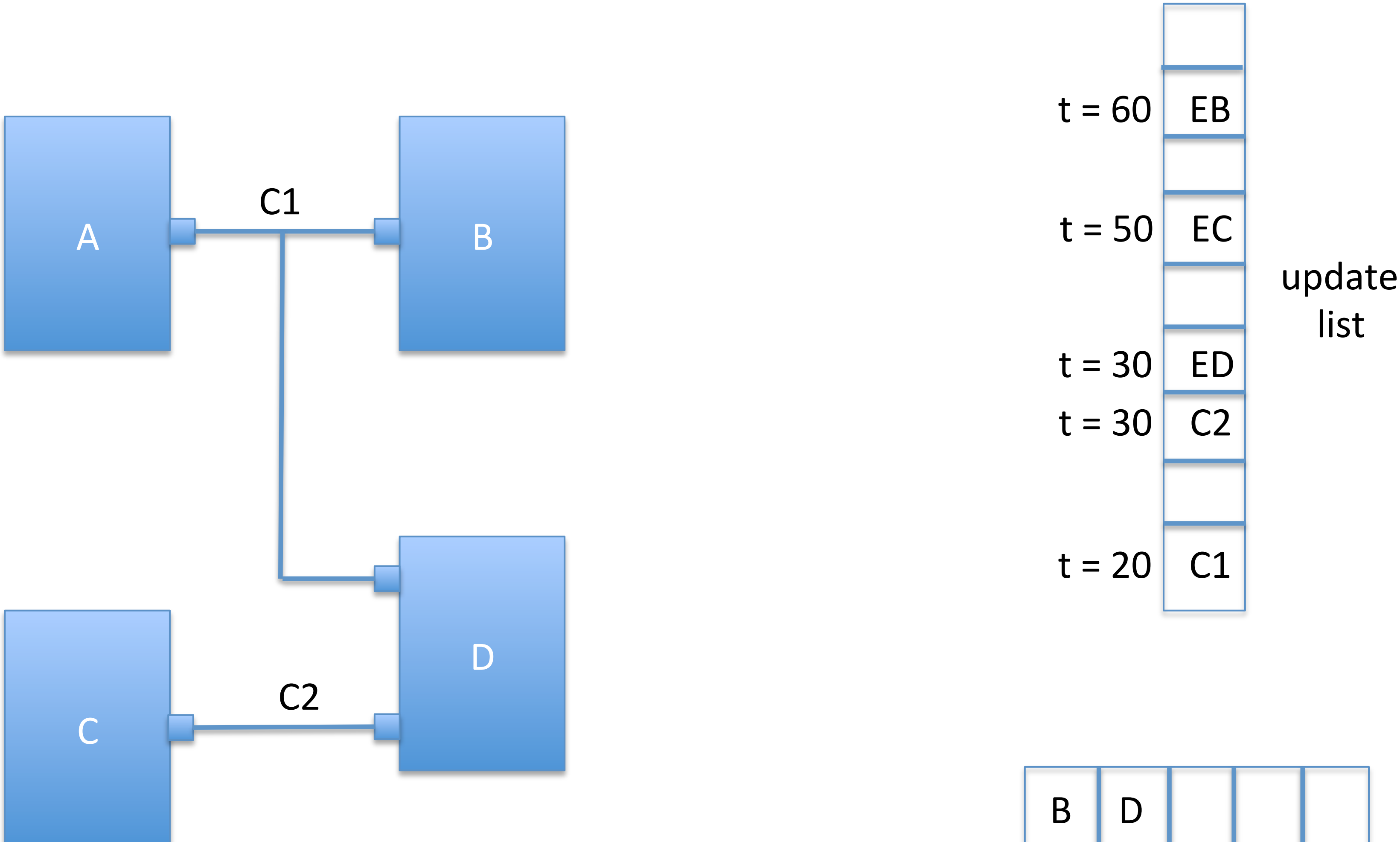
Global Time is Advanced to t = 20



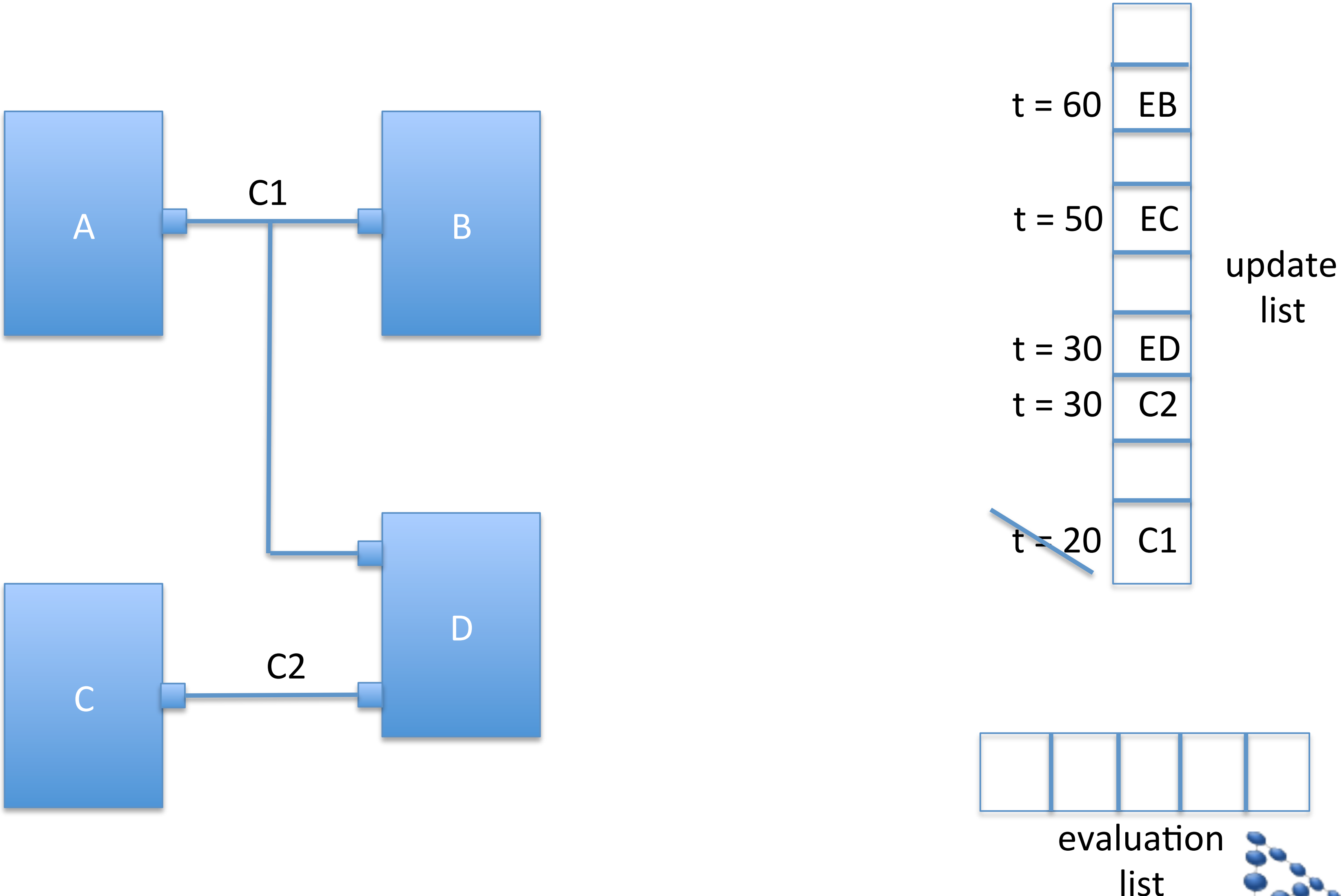
Add to *eval list* each component on C1 *fanout*



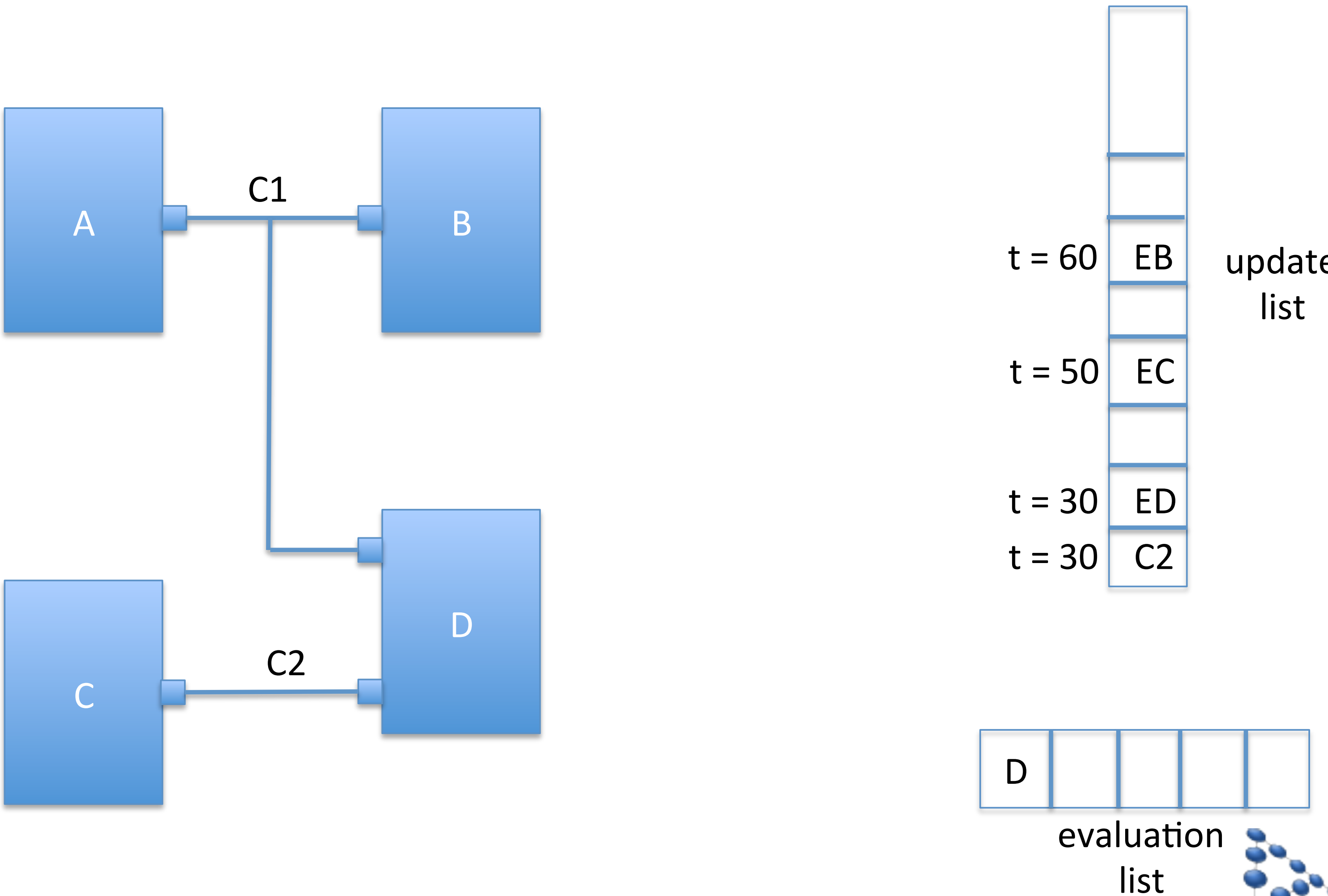
B calls *ScheduleEval* with delay 40



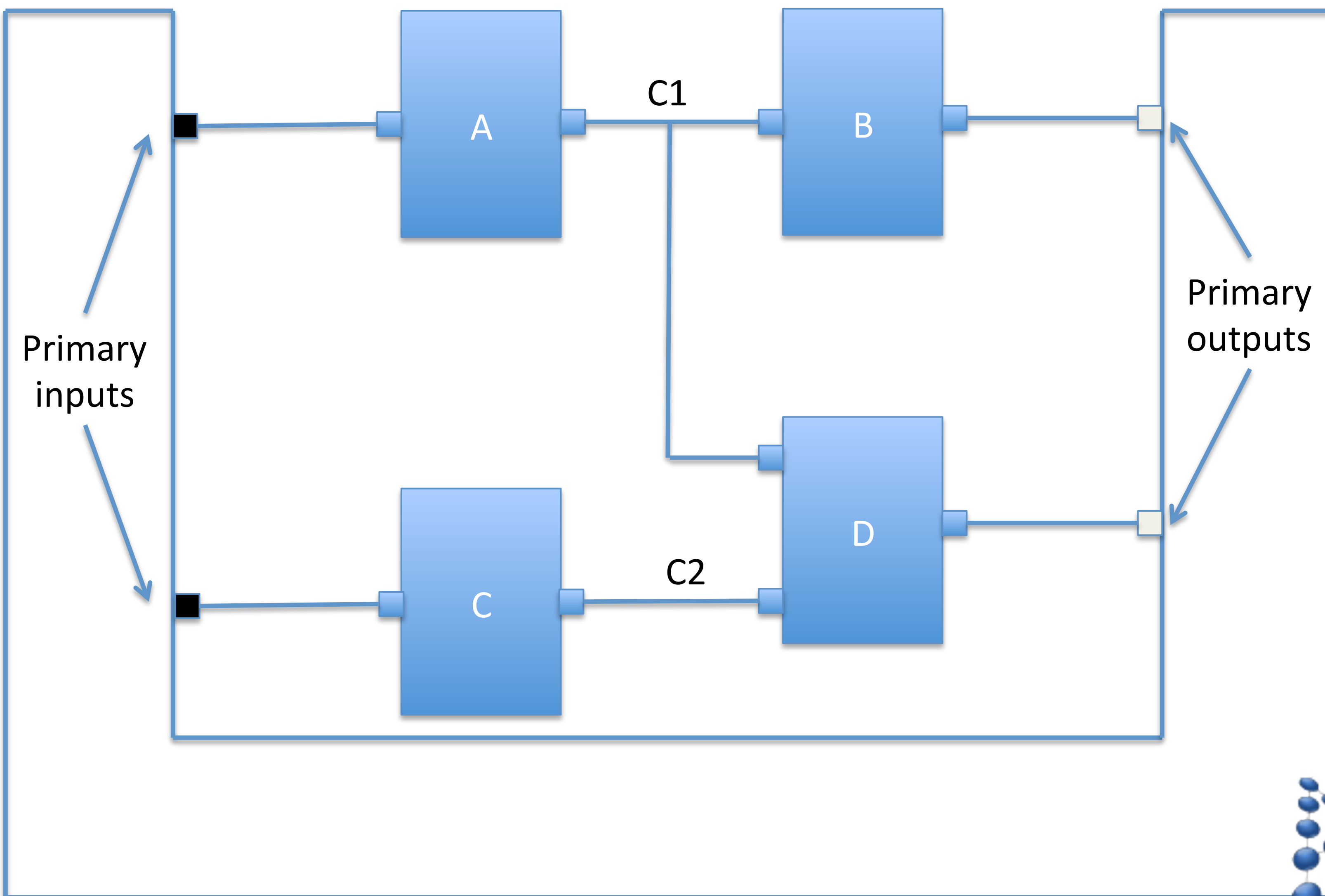
Time advances to t = 30: *two* updates



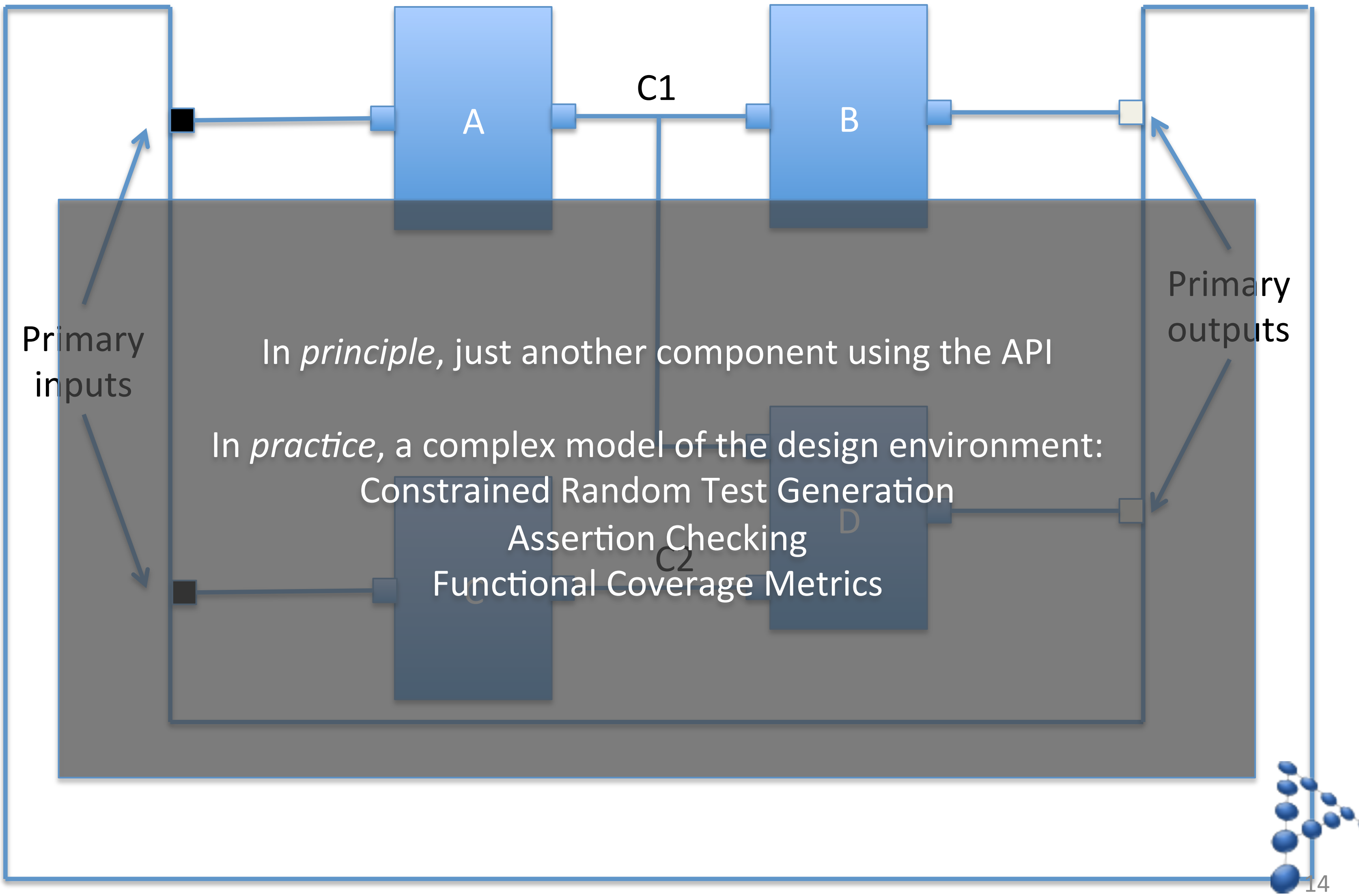
Time advances to t = 30: *two* updates, *one* eval



The Simulation Testbench



The Simulation Testbench



In *principle*, just another component using the API

In *practice*, a complex model of the design environment:
Constrained Random Test Generation
Assertion Checking
Functional Coverage Metrics

Simulation-based Verification

- As opposed to Formal Verification
  - Theorem Proving
  - Model Checking
- Structural Coverage
- Functional Coverage
- Assertion Checking
- Assertion Coverage
- Integrating Formal and Simulation-based Verification

Structural Coverage

Do the requirements- based test cases adequately exercise the structure of the source code?

DO 178B

- Statement Coverage
- Branch (Decision) Coverage
- MC/DC (Modified Condition/Decision Coverage)
  - Unique Cause
  - Masking

MC/DC
(Modified Condition/Decision Coverage)

Z = (A or B) and (C or D)

A	B	C	D	Z
F	F	F	T	F
…				
T	F	F	T	T

MC/DC
(Modified Condition/Decision Coverage)

Z = (A or B) and (C or D)

	A	B	C	D	Z
Unique Cause	F	F	F	T	F
	…				
	T	F	F	T	T

MC/DC
(Modified Condition/Decision Coverage)

Z = (A or B) and (C or D)

Masking

A	B	C	D	Z
F	F	F	T	F
…				
T	F	F	T	T
T	F	T	F	T
T	F	T	T	T

MC/DC (Modified Condition/Decision Coverage) Summary

- Masking MC/DC results in fewer tests and shorter simulation runs
- Unique-Cause MC/DC cannot deal with repeated conditions
- Unique-Cause MC/DC may detect more errors (NB not the *actual* DO 178B requirement)

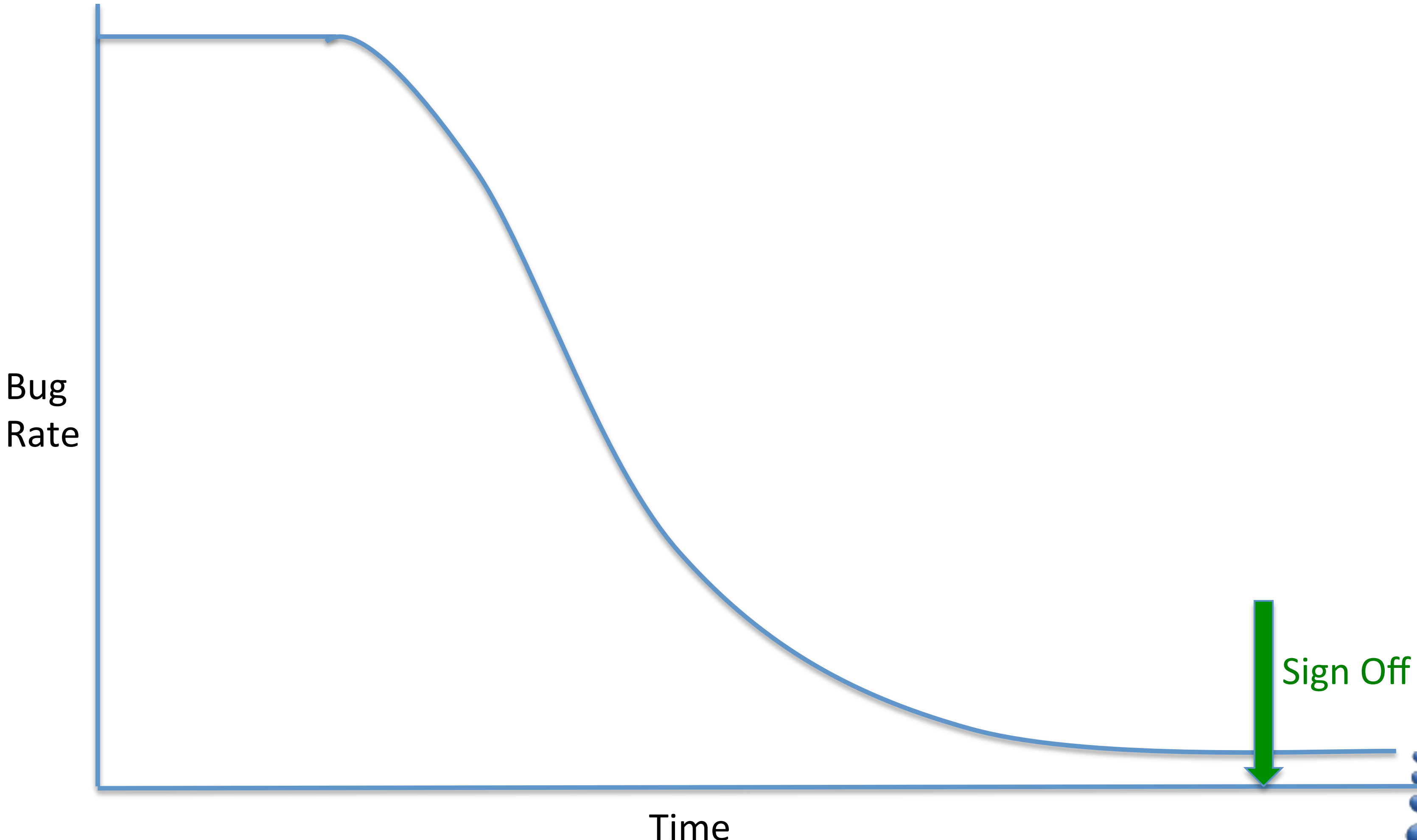
Do the requirements- based test cases adequately exercise the structure of the source code?

DO 178B

DO 178 C

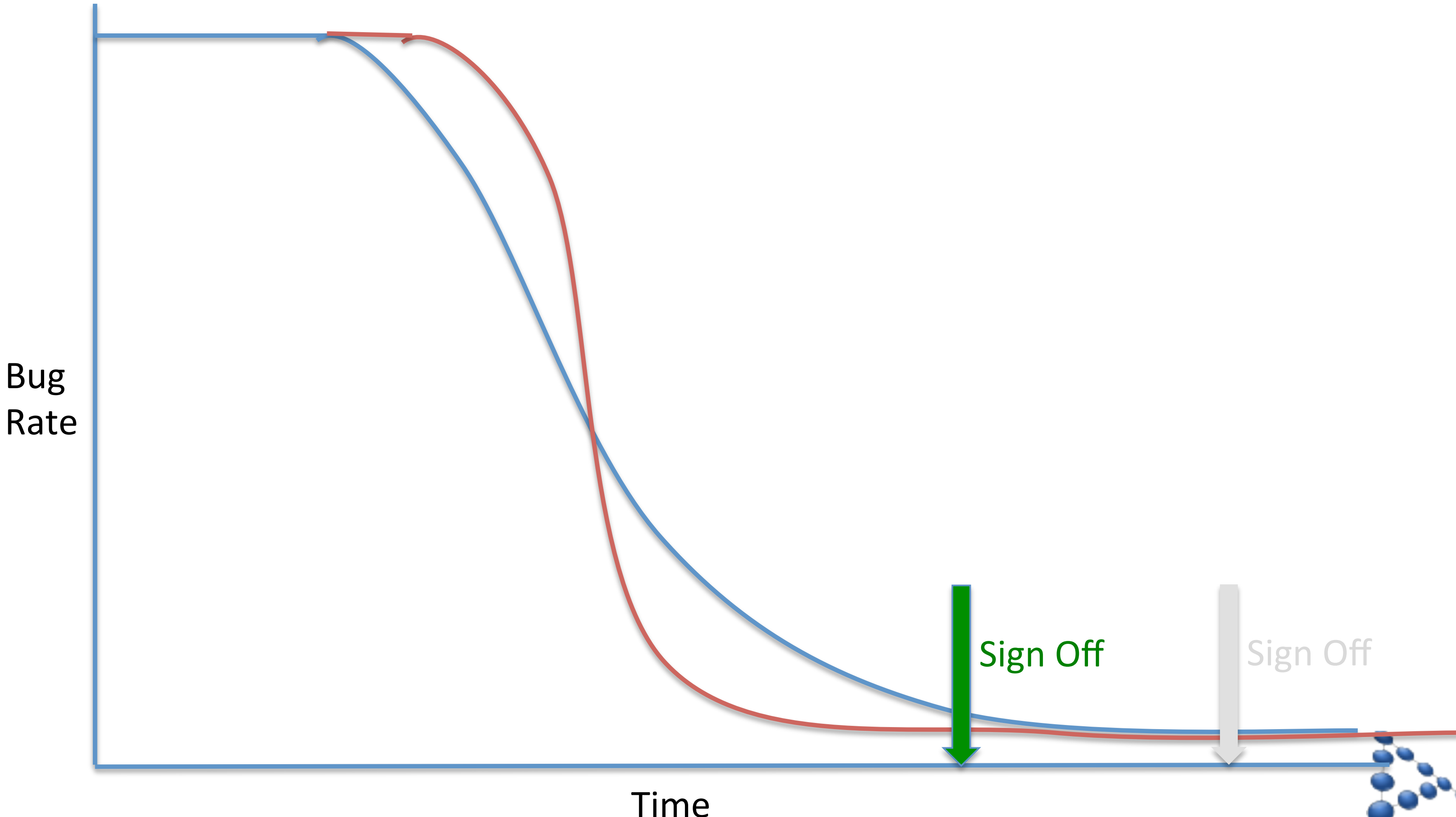
- Approved: December 2011
- Provision for Formal Methods
- Provision for Object-Oriented Code Development
  - As opposed to Structured Code Development
  - Does MC/DC have the same value?

Bug Rate vs Time



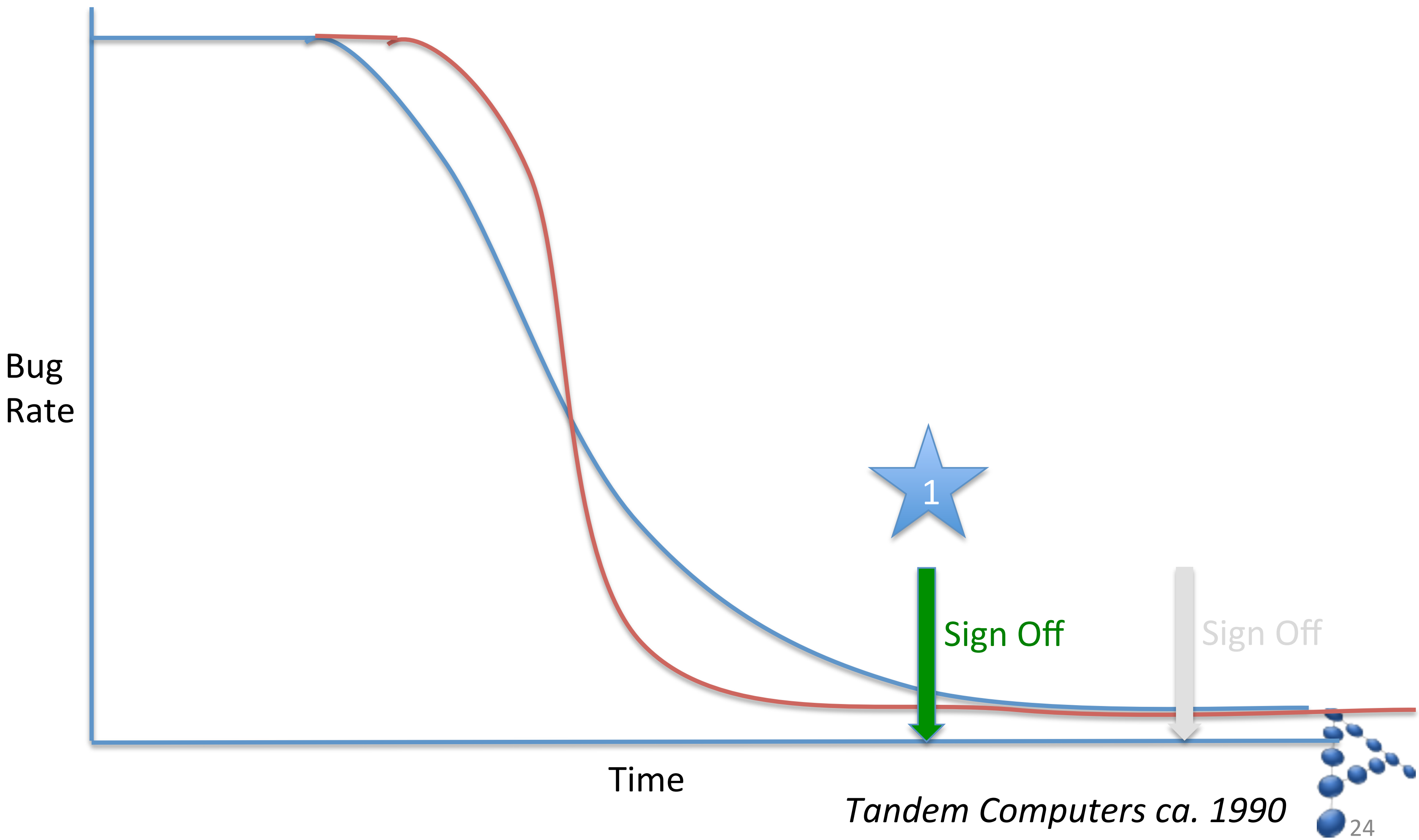
Tandem Computers ca. 1990

Bug Rate vs Time – ***Coverage Directed*** Verification

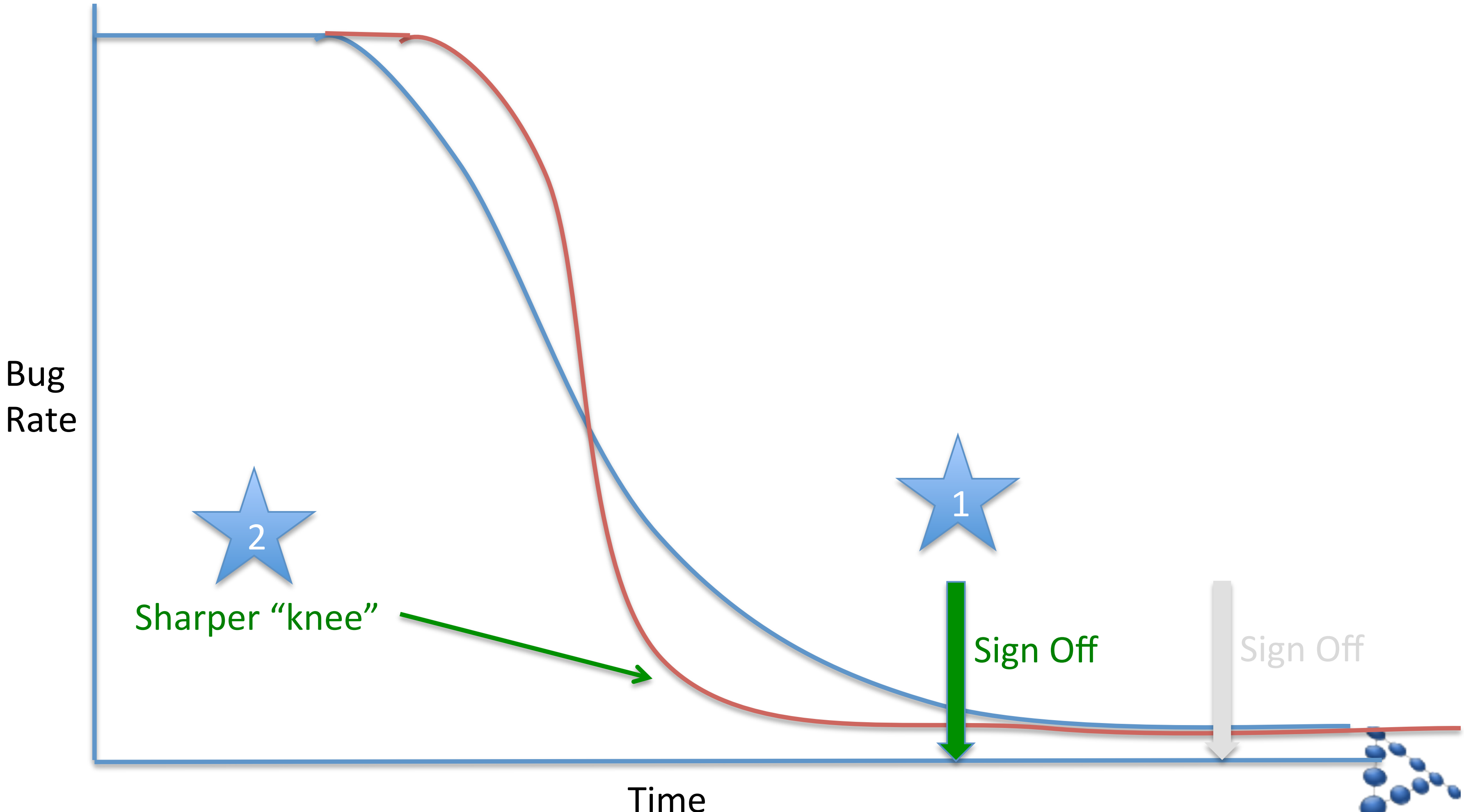


Tandem Computers ca. 1990

Bug Rate vs Time – ***Coverage Directed*** Verification



Bug Rate vs Time – ***Coverage Directed*** Verification



Tandem Computers ca. 1990

Functional Coverage

- *User-defined* Coverage Metrics
  - Typical Scenarios
  - Error Cases
  - Corner Cases
- High-Level Language Description
- Constrained Random Test Generation from Coverage Description
- “Scoreboarding”

Functional Coverage Example
Transaction Coverage

Transaction Type	Header1	Header2	Typical Value	Min. Value	Max. Value
A					
B			-	-	-
C		-			

Functional Coverage Example
Transaction Coverage

Transaction Type	Header1	Header2	Typical Value	Min. Value	Max. Value
A	●	●	●	●	●
B	●	●	-	-	-
C	●	-	●	●	●

Functional Coverage: 8/11 (73%)

Assertion Checking: PSL (Property Specification Language)

- LTL-based
- Unit of time is the Clock Cycle
- SEREs
  - Sequential Extended Regular Expressions

{ a ; not a ; b } |=> { c }

Assertion Checking: PSL (Property Specification Language)

- PSL is converted to *Simulation Monitors*
  - First, convert to Buchi Automata (non-deterministic)
  - Second, generate deterministic automata
  - Third, generate HDL representation

- Simulate the Monitors together with the Design

{ a ; not a ; b } |=> { c }

Assertion Coverage

- How effectively do the tests exercise the Assertions?
  - Vacuously Satisfied? (a is always false)
  - Are the SEREs *sensitised* to detect assertion failure?

{ a ; not a ; b } |=> { c }

- Is the set of assertions
  - Necessary?
  - Sufficient?

Integrating Formal and Simulation-based Verification

- PSL SEREs can be verified using a model checker
  - Assertion Coverage principles still apply
  - Assertion Coverage results for simulation an model checking can be combined
- Are the assertions necessary/sufficient?
  - An open problem
  - Scope for using theorem proving?

Verification Sign Off

- Ultimately, an engineer will have to physically provide a signature
- How confident is the engineer that the design meets its specification?
  - The outcome of the verification process must be measurable
    - Bug rate
    - Coverage metrics

Summary

- Discrete Event Simulation
  - Two-List Algorithm for Deterministic Execution
- Coverage-Driven Verification
  - Structural (MC/DC)
  - Functional
- Assertion Checking/Coverage
- Combining Formal and Simulation-based Verification for
  - Earlier Sign Off with
  - Increased Confidence