Mixing Models of Computation

Wan-Teh Chang
Edward A. Lee
David G. Messerschmitt
EECS Department
UC Berkeley

Major collaborators:

Stephen A. Edwards
Alain Girault
Soonhoi Ha*
Bilung Lee

*** Seoul National University, Korea**

Motivation

Objective: system-level design of systems that are heterogeneous in

- **Design styles**: signal processing, control-oriented
- **Implementation technologies**: hardware, software

Heterogeneous approach: combines small, specialized models of computation

- Achieves generality
- Automatic synthesis and formal verification

Models of Computation

Model of computation (MoC): the semantics of the interaction between modules or components

- Organizing principles of a design specification or model
- Domain-specific, intuitive

Examples:

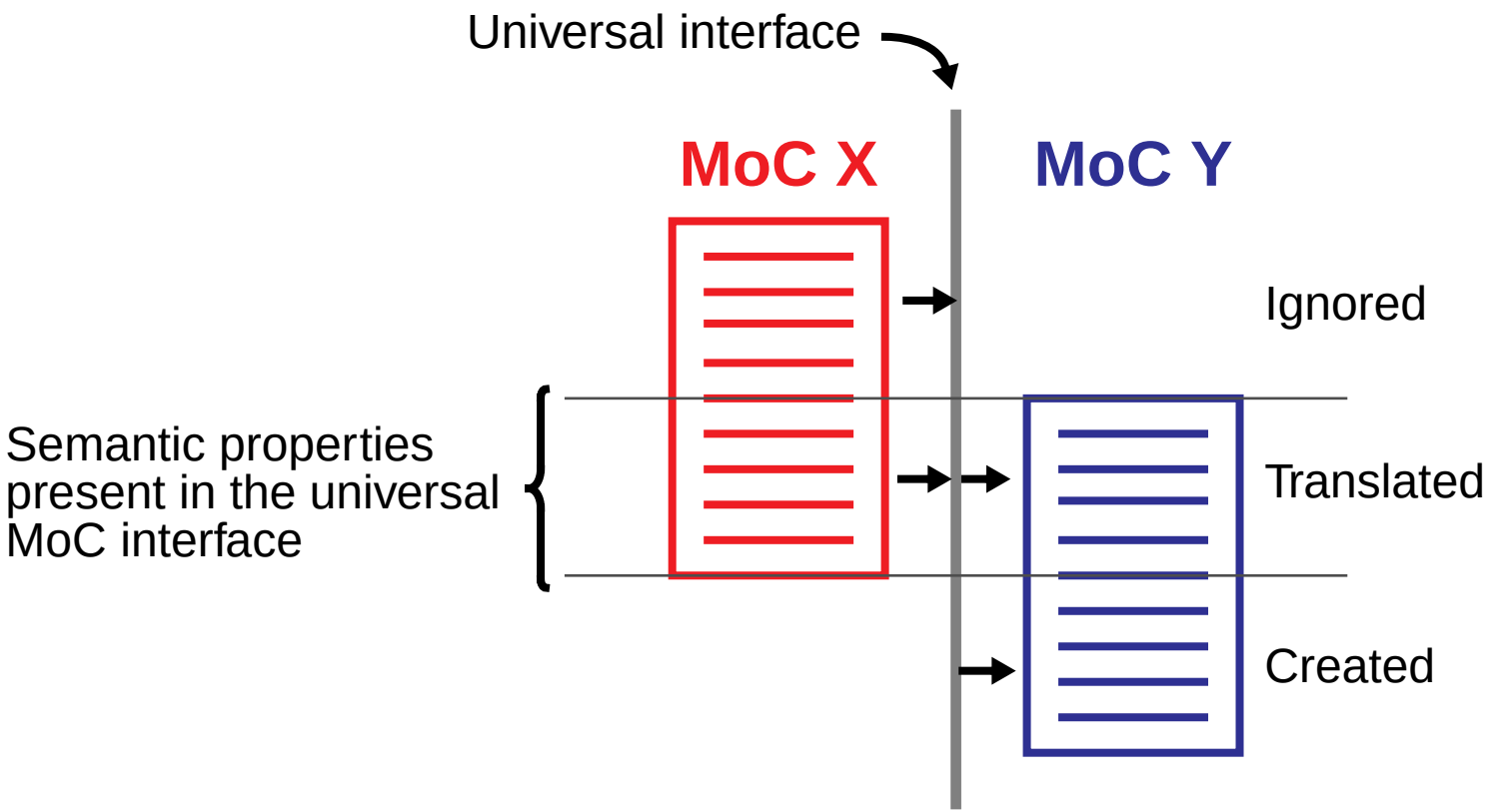
- Dataflow
- Discrete-event
- Synchronous languages
- Hierarchical finite-state machines

Approach: Universal MoC Interface

Define interaction semantics to resolve ambiguity at the interface.

MoCs interface to a universal interface.

- N **MoCs require** N **interfaces rather than** N^2**.**

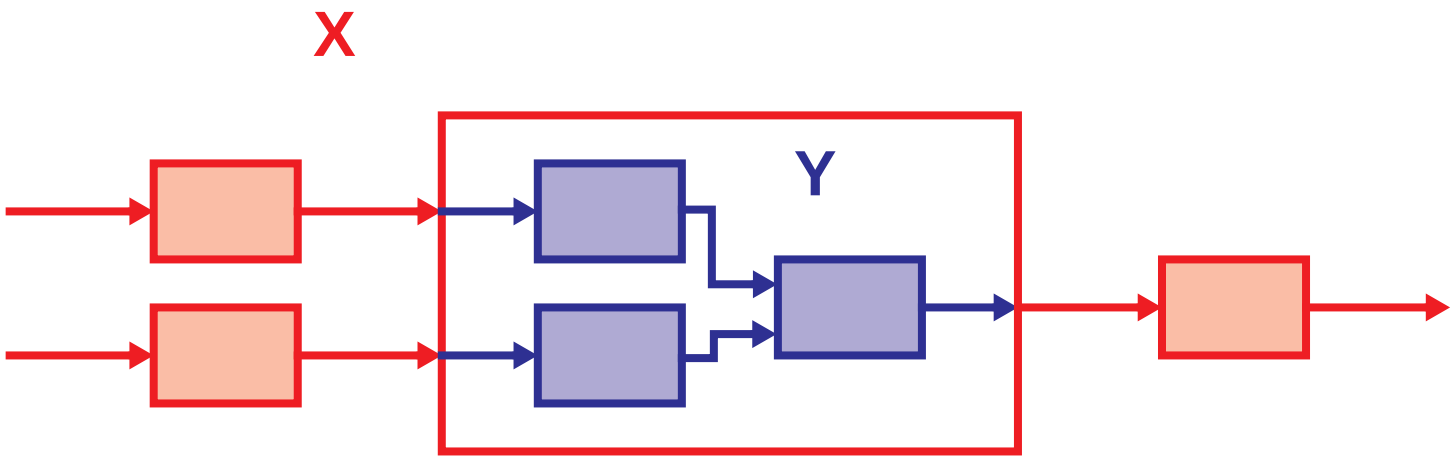


Approach: Mixing MoCs by Hierarchical Nesting

MoCs are mixed hierarchically.

Blocks in the MoCs have discrete firings.

Key constraint: each MoC have a well-defined quantum of computation.



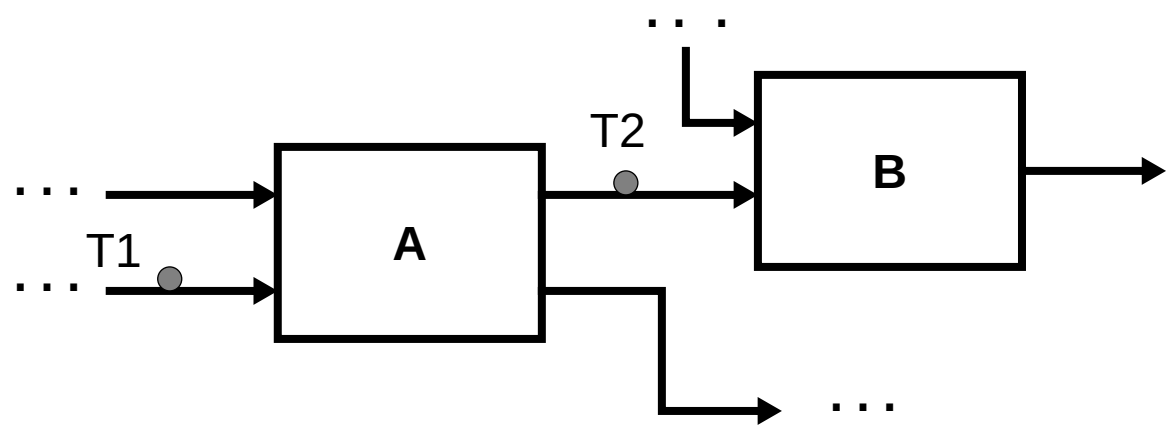
A subsystem of MoC Y embedded in MoC X as a hierarchical block

Discrete-Event

Discrete-event (DE): events have time stamps.

- **Events are totally ordered by time stamp.**

The DE simulator sorts the events by time stamp and process the events in chronological order.

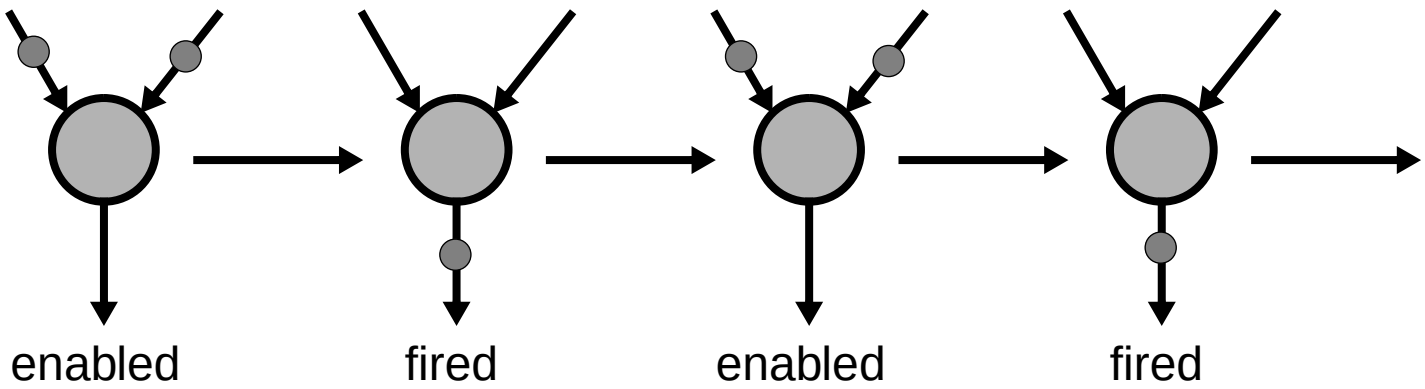


Block A fires at time T1, generating an event with time stamp T2 (T2 >= T1). Block B fires at time T2.

Synchronous Dataflow

Synchronous dataflow (SDF): a block consumes a fixed number of tokens and produces a fixed number of tokens in each firing.

- **Block firings can be scheduled statically at compile time.**
- **Good for modeling multirate digital signal processing.**



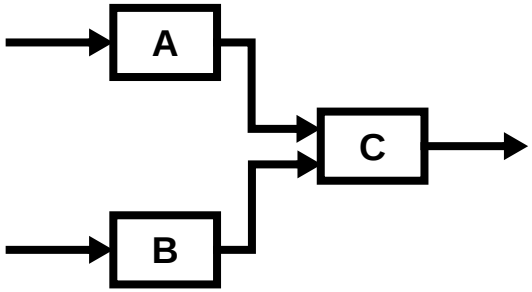
Repeated firings of a dataflow block

Synchronous Dataflow (continued)

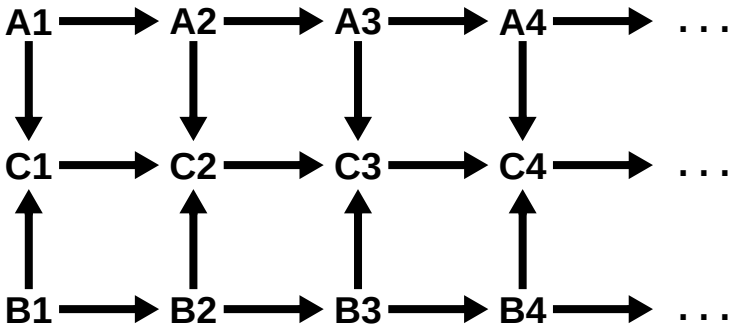
Block firings are partially ordered, sequenced only by data dependency.

Does not have a notion of time.

Quantum of computation: a complete cycle that returns the graph to its original state (number of tokens on each arc).

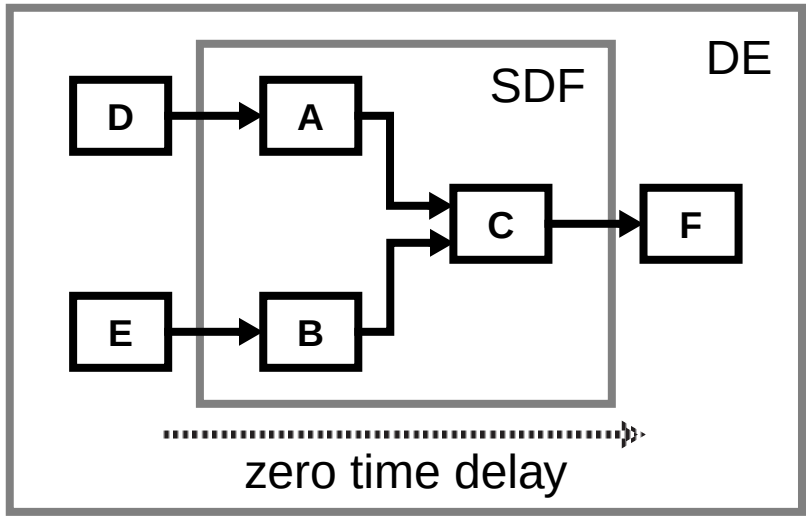


A complete cycls is
(A, B, C) or (B, A, C)

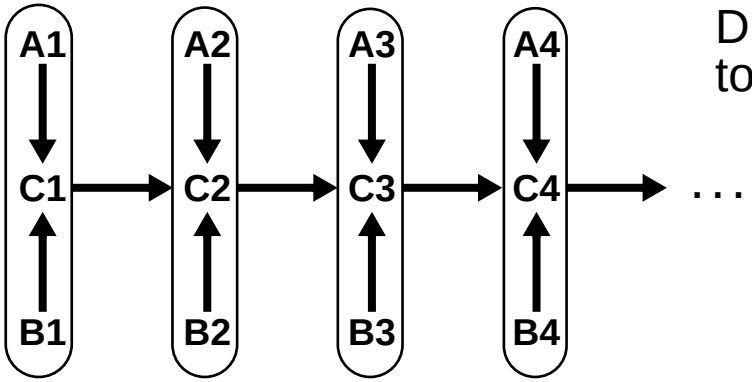


Partial ordering of firings imposed
by the dataflow MoC

SDF in DE



Quantum of computation: a complete cycle



DE assigns time stamps to the complete cycles.

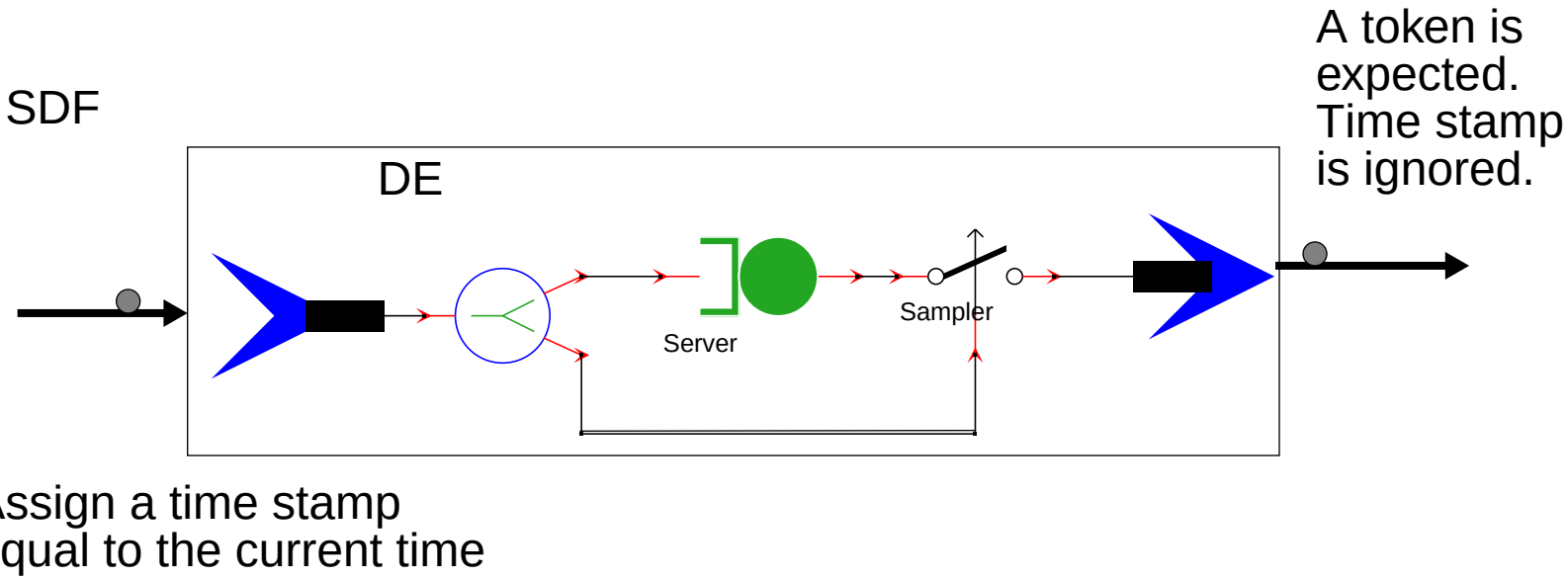
Partial order of firings after imposing a total order on complete cycles

DE in SDF

SDF must maintain the current time at the interface.

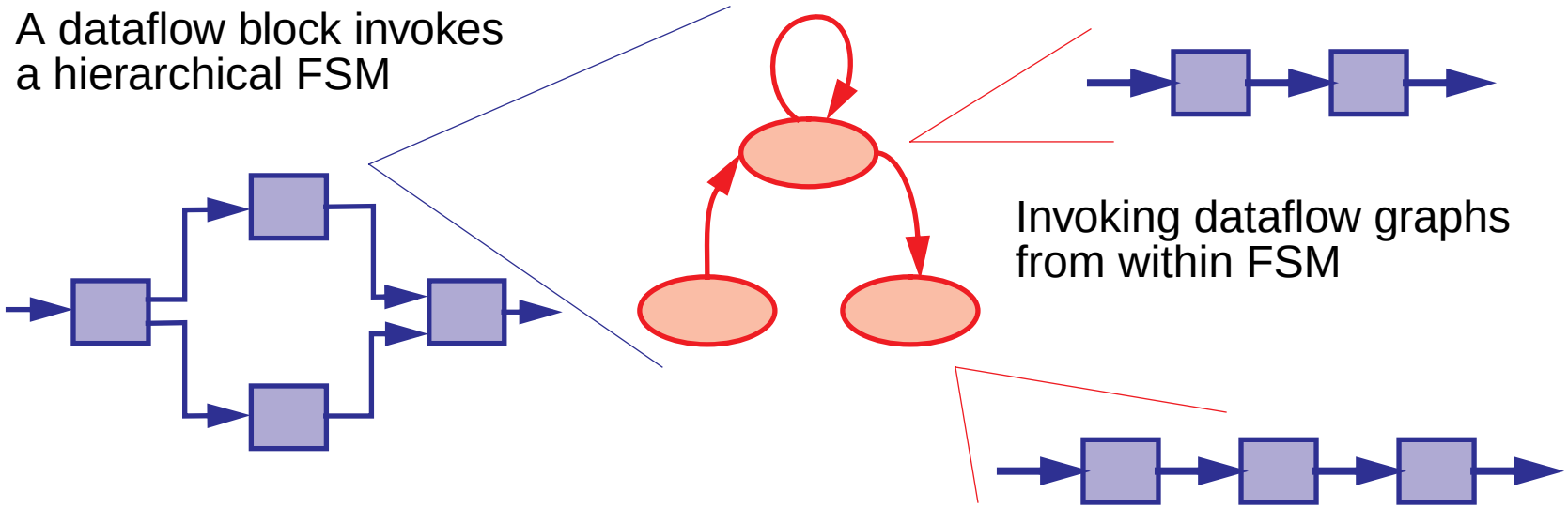
- **Given by another outside MoC (see previous example)**
- **If top level, advanced by schedule period in each complete cycle**

Inner DE subsystem must behave externally like a SDF block.



Mixing Hierarchical FSM with Concurrency

- **Sequential behavior (finite-state machine), hierarchy, and concurrency are orthogonal semantic components.**
- **Hierarchical finite-state machines (FSMs) can be nested with different concurrency models (SDF, synchronous reactive) to get (essentially) variants of Statecharts.**



Conclusions

Heterogeneous approach to system-level design

- **Define interaction semantics to resolve ambiguity**

Mix MoCs by hierarchical nesting

- **Universal interface: N rather than NxN**
- **Execution proceeds as a sequence of quanta of computation with time stamps**

We can mix dataflow, discrete-event, synchronous reactive, and hierarchical FSM models.

- **See the Ptolemy demos**
- **Visit our Web site: `http://ptolemy.eecs.berkeley.edu`**