A Meta-modeling Framework for Dynamic Reconfiguration of Dataflow Graphs

Sankalita Saha, Dong-Ik Ko and
Shuvra S. Bhattacharyya

Department of Electrical and Computer Engineering, and
Institute for Advanced Computer Studies,
University of Maryland, College Park MD, USA

UMIACS
University of Maryland Institute for Advanced Computer Studies

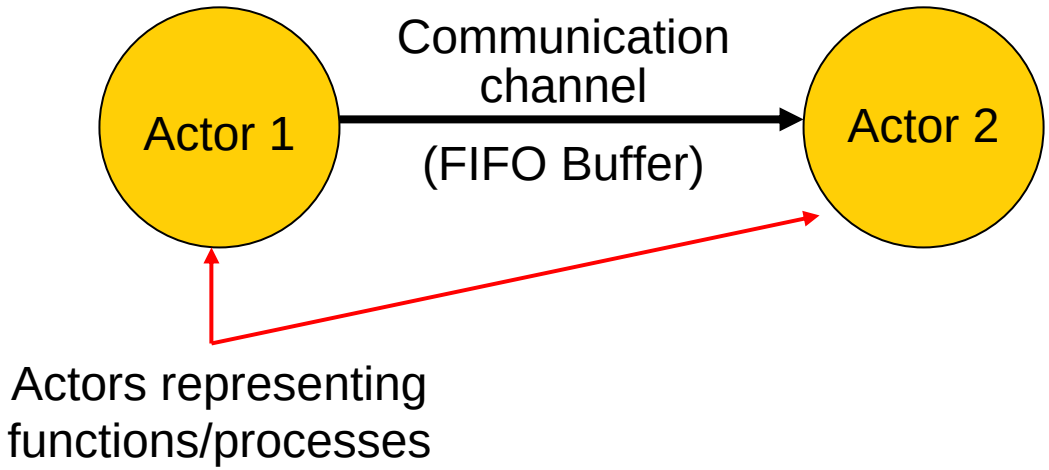
DEPARTMENT OF
ELECTRICAL &
COMPUTER ENGINEERING

Outline

- Introduction
- Background
- Dataflow Graphs
- Synchronous Dataflow
- Parameterized Dataflow
- Blocked Dataflow
- Dynamic Graph Topology
- Conclusions

Introduction: Dataflow

- Used widely in design tools for DSP.
- Application is modeled as a directed graph.



Introduction: Dataflow

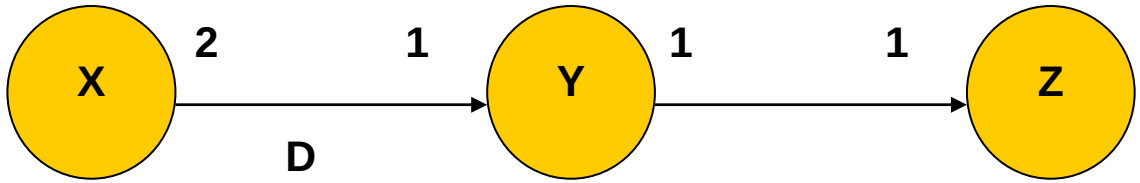
- Data-driven execution model
  - A node can execute whenever it has sufficient data on its input edges.
  - The order in which nodes execute is not part of the specification.
  - The order is typically determined by the compiler, the hardware, or both.
- Iterative execution
  - Body of loop to be iterated a large or infinite number of times.

Background

- Petri nets
  - Depicts structure of a distributed system as a directed bipartite graph consisting of places and transitions. Arcs run between places and transitions.
- Marked Graphs
  - Special case of a Petri net where every place has exactly one incoming arc and one outgoing arc.
- Computation Graphs
  - A finite graph with a set of nodes, each associated with a function and a set of arcs. A branch is a queue of data directed from one node to another.
  - A function can be fired only when the number of tokens present on the arc exceeds a given threshold.
- Kahn Process Networks
  - Consists of nodes representing arbitrary sequential programs communicating via channels of the process networks with blocking read and non-blocking write operations.

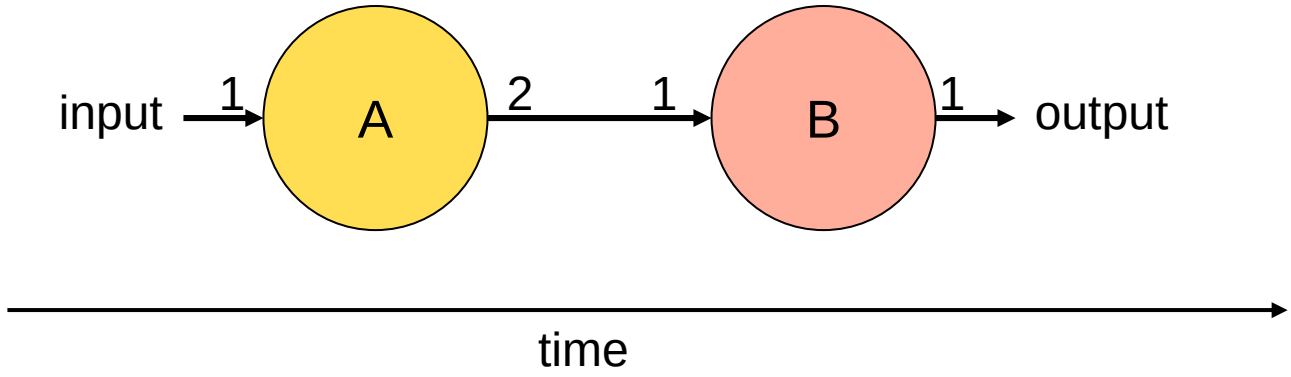
Synchronous Dataflow (SDF)

- The number of tokens produced and consumed is restricted to be a positive integer known at compile time [1].



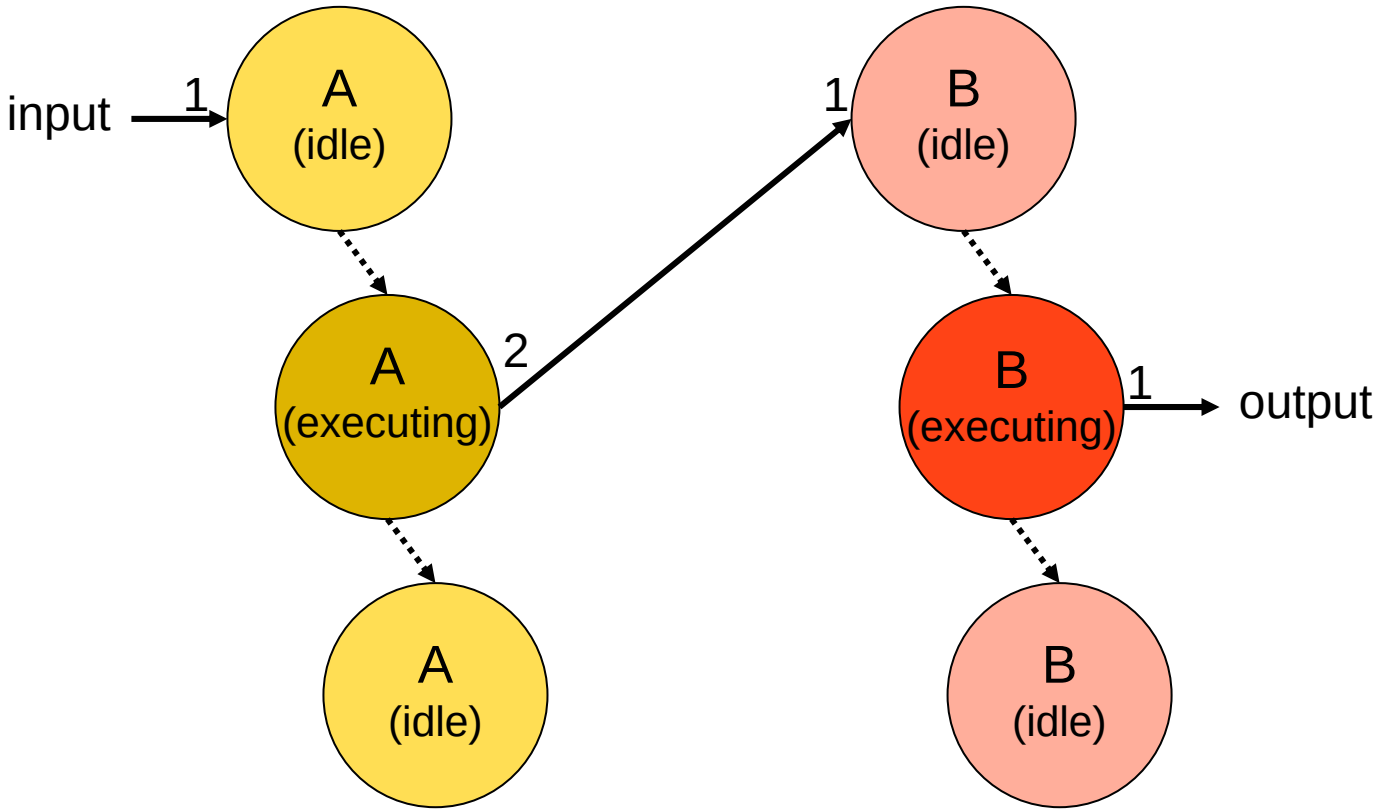
Dataflow: Data-triggered or Event-triggered?

- Dataflow graphs essentially represent data-triggered system.



Dataflow: Data-triggered or Event-triggered?

- Dataflow can model event triggered systems too



Dataflow: Pros and Cons

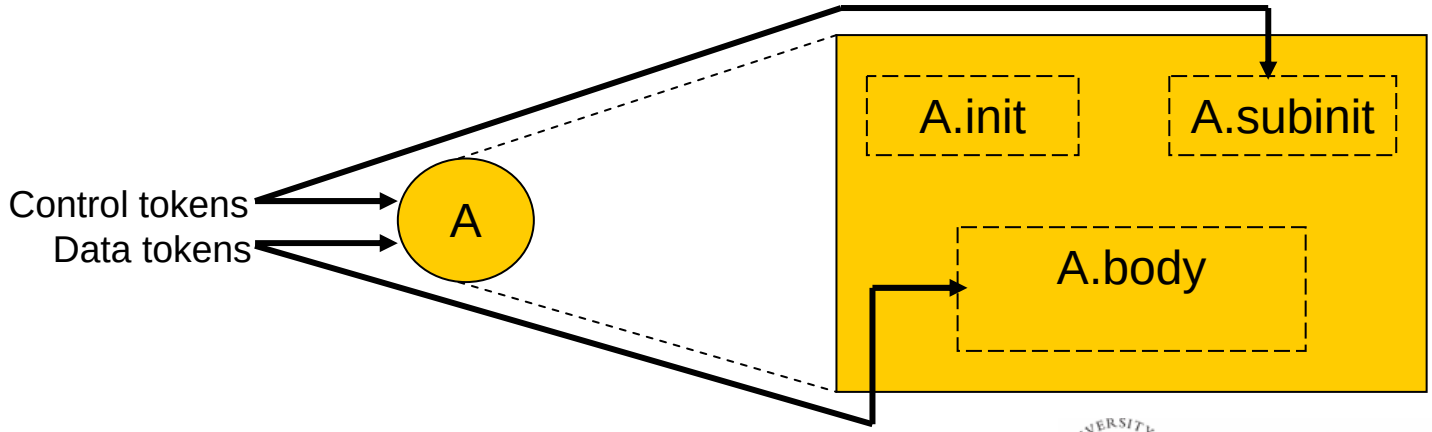
- Advantages
  - Static scheduling and compile-time predictability for useful restricted forms of dataflow
  - Exposes opportunities for coarse-grain optimizations
- Disadvantages
  - Limited expressive power of the static (restricted forms)
  - Lack of intuitive appeal for the dynamic (Turing complete) forms

Parameterized Dataflow

- (PDF) is a meta-modeling framework that can be applied to any underlying form of dataflow graph (“base model”) that has a well-defined notion of graph iteration [2].
- Increases expressive power while allowing many static analysis techniques, and keeping much of the intuitive structure of the base model.

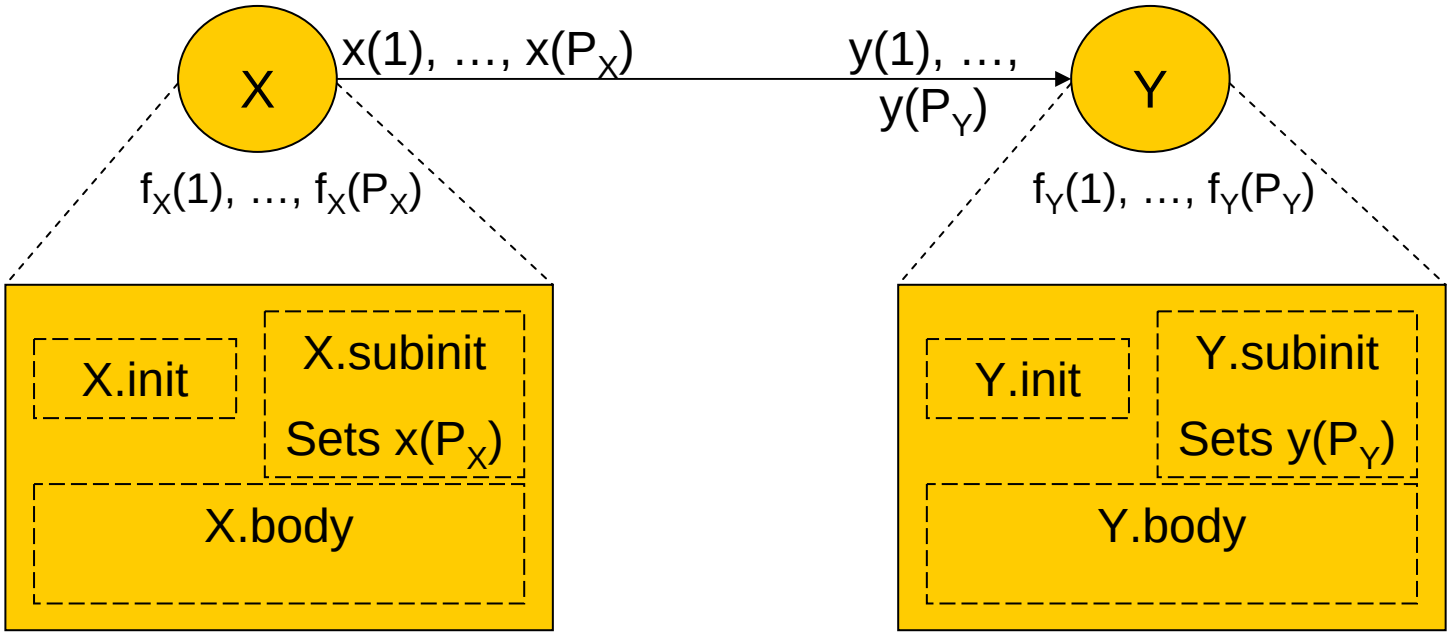
Parameterized SDF (PSDF)

- Parameterized SDF is obtained by parameterizing arbitrary aspects of an SDF graph (actors, edges, dataflow properties) and allowing parameters to be dynamically changed according to a structured hierarchical discipline [2].
  - *init* and *subinit* graphs are dataflow graphs that are responsible for reconfiguring parameters associated with a subsystem.
  - *init* executes less frequently but can perform less restricted reconfiguration operations compared to subinit.
- An actor is characterized by a set of parameters that can control the actor's functionality, as well as its dataflow behavior.



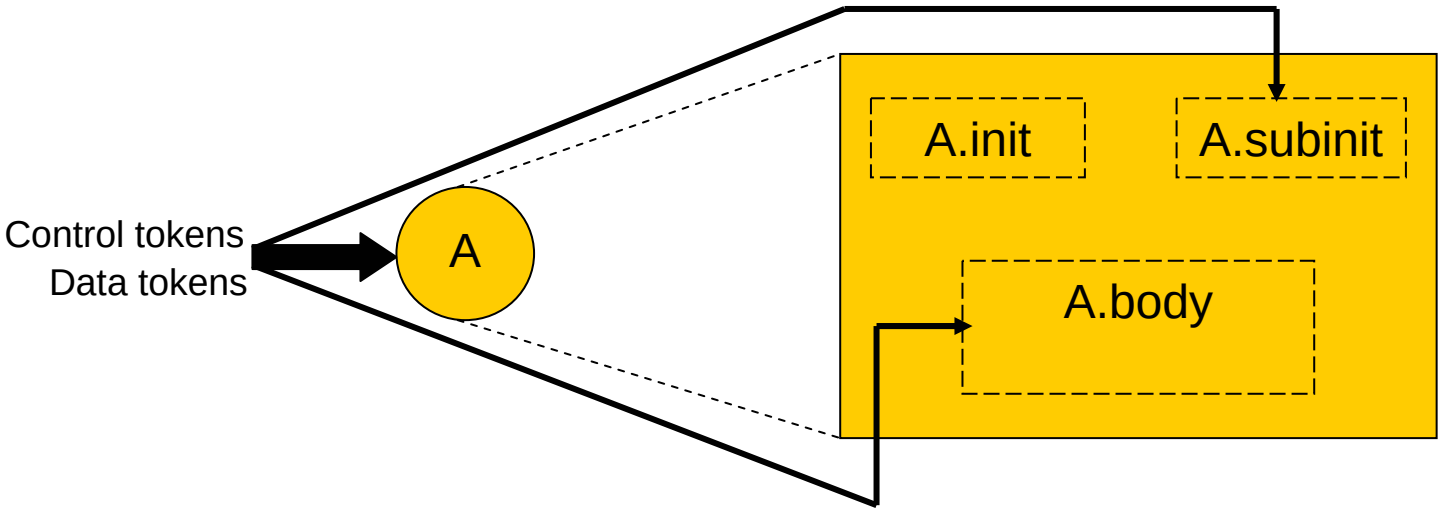
Parameterized Cyclo-static SDF (PCSDF)

- Parameterized CSDF is obtained by parameterizing cyclo-static SDF.
- There are 2 fundamental dataflow properties which can be parameterized
  - period of the cycle of phases
  - data rates associated with each phase



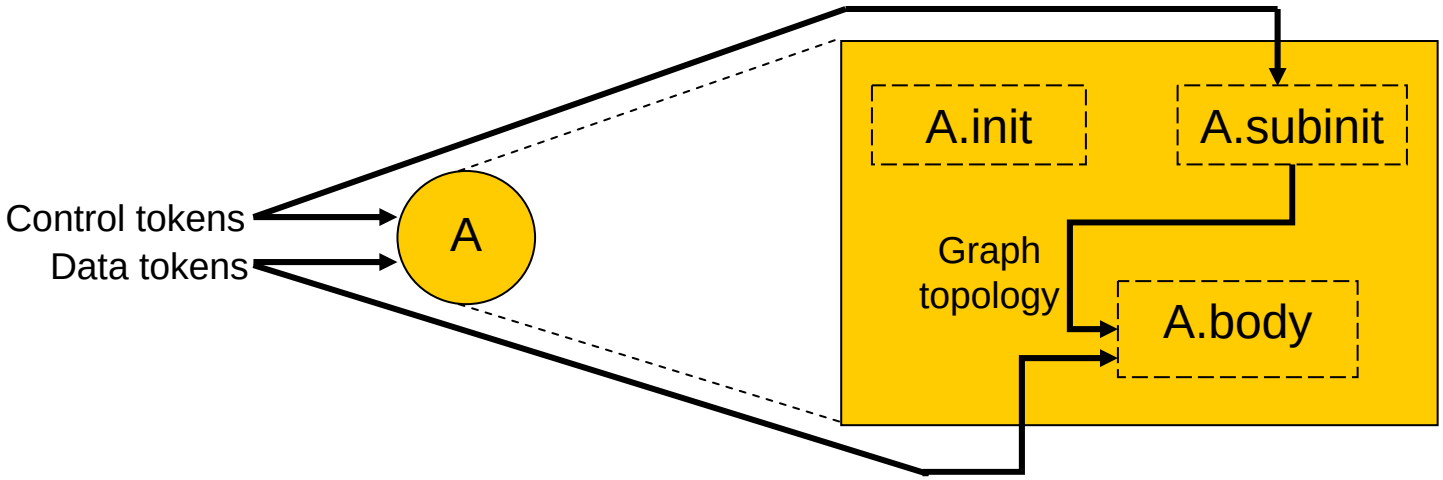
Blocked Dataflow

- Extends parameterized dataflow: the *set* of *parameters* used to reconfigure sub-systems or sub-graphs are extracted from the input data stream [3]

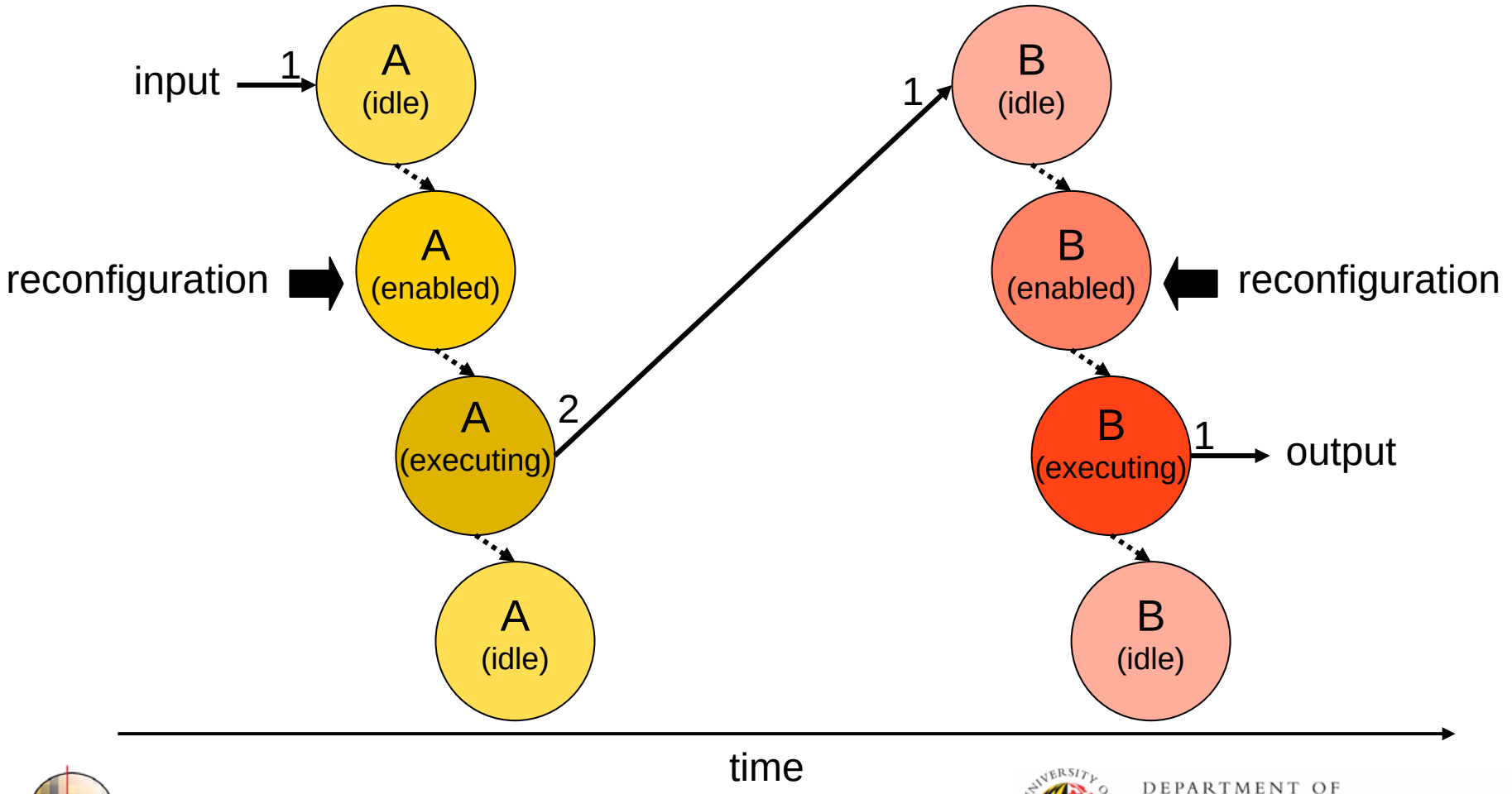


Dynamic Graph Topology

- Extends PSDF to allow dynamic change of graph topology at run-time [4].
  - The *init* graph determines the transfer rate of each port of a graph.
  - The *subinit* graph determines the graph topology of the associated body graph before the invocation of the body graph.



Parameterized Dataflow and Event-triggered Systems



Experiments (DGT)

- An MPEG2 video encoder was implemented.
- Target platform was Texas Instruments C64xx series using Code Composer Studio.
- Two implementations were done
  - Separate graph approach using a combination of SDF and FSM.
  - DGT implementation.
- SAS (Single Appearance Schedule) and MAS (Multiple Appearance Schedule) and a combination of SAS and MAS was used.
- The DGT method selects different scheduling methods (SAS or MAS) depending on graph characteristics.

Experimental Results (DGT)

< DGT Graph >
C1 : SAS
C2 : MAS
C3 : SAS+MAS

< Separate Graph >
C4 : SAS
C5 : MAS
C6 : SAS+MAS

Frame Size		DGT approach			Separate Graph approach		
		C1	C2	C3	C4	C5	C6
128 *128	Code	26,469	31,946	26,469	63,341	79,773	63,341
	Buffer	1,557	1,429	1,557	4,667	4,283	4,667
	Total	28,026	33,375	**28,026**	68,008	84,056	**68,008**
256 *256	Code	26,469	31,946	26,469	63,341	79,773	63,341
	Buffer	6,173	5,661	6,173	18,515	16,979	18,515
	Total	32,642	37,607	**32,642**	81,856	96,752	**81,856**
480 *720	Code	26,469	44,903	31,393	63,341	118,645	94,180
	Buffer	52,852	19,991	21,788	158,551	59,967	65,364
	Total	79,321	64,894	**53,181**	221,892	178,612	**159,544**
768 *1024	Code	26,469	58,074	44,564	63,341	158,157	133,692
	Buffer	130,680	45,320	49,397	392,035	135,955	148,192
	Total	157,149	103,394	**93,961**	455,376	294,112	**281,884**
1080 *1920	Code	26,469	58,074	50,041	63,341	158,157	150,124
	Buffer	1,817,064	100,940	100,937	5,451,187	302,815	302,524
	Total	1,843,533	159,014	**150,978**	5,514,528	460,972	**452,648**

Conclusions

- Parameterized dataflow is powerful modeling paradigm for signal processing applications.
- Various forms of dynamic reconfigurability can be achieved using parameterized dataflow and associated models of computation.
- Though traditional view of dataflow based systems is as data-triggered systems, they can be used to model event-triggered systems as well.

Thank you!

References

- E.A. Lee, D.G. Messerschmitt. Static Scheduling of Synchronous Dataflow Programs for Digital Signal Processing. *IEEE Transactions on Computers*, February, 1987.
- B. Bhattacharya and S. S. Bhattacharyya. Parameterized dataflow modeling for DSP systems. *IEEE Transactions on Signal Processing*, 49(10):2408–2421, October 2001.
- D. Ko and S. S. Bhattacharyya. Dynamic configuration of dataflow graph topology for DSP system design. In Proceedings of the ICASSP, pp. V–69–V–72, Philadelphia, Pennsylvania, March 2005.
- D. Ko and S. S. Bhattacharyya. Modeling of block-based DSP systems. Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology, 40(3):289 – 299, July 2005.