

# Leveraging Reconfigurability in the Design Process

Lesley Shannon and Paul Chow

Department of Electrical and Computer Engineering University of Toronto, Canada {lesley,pc}@eecg.toronto.edu

## The Concept

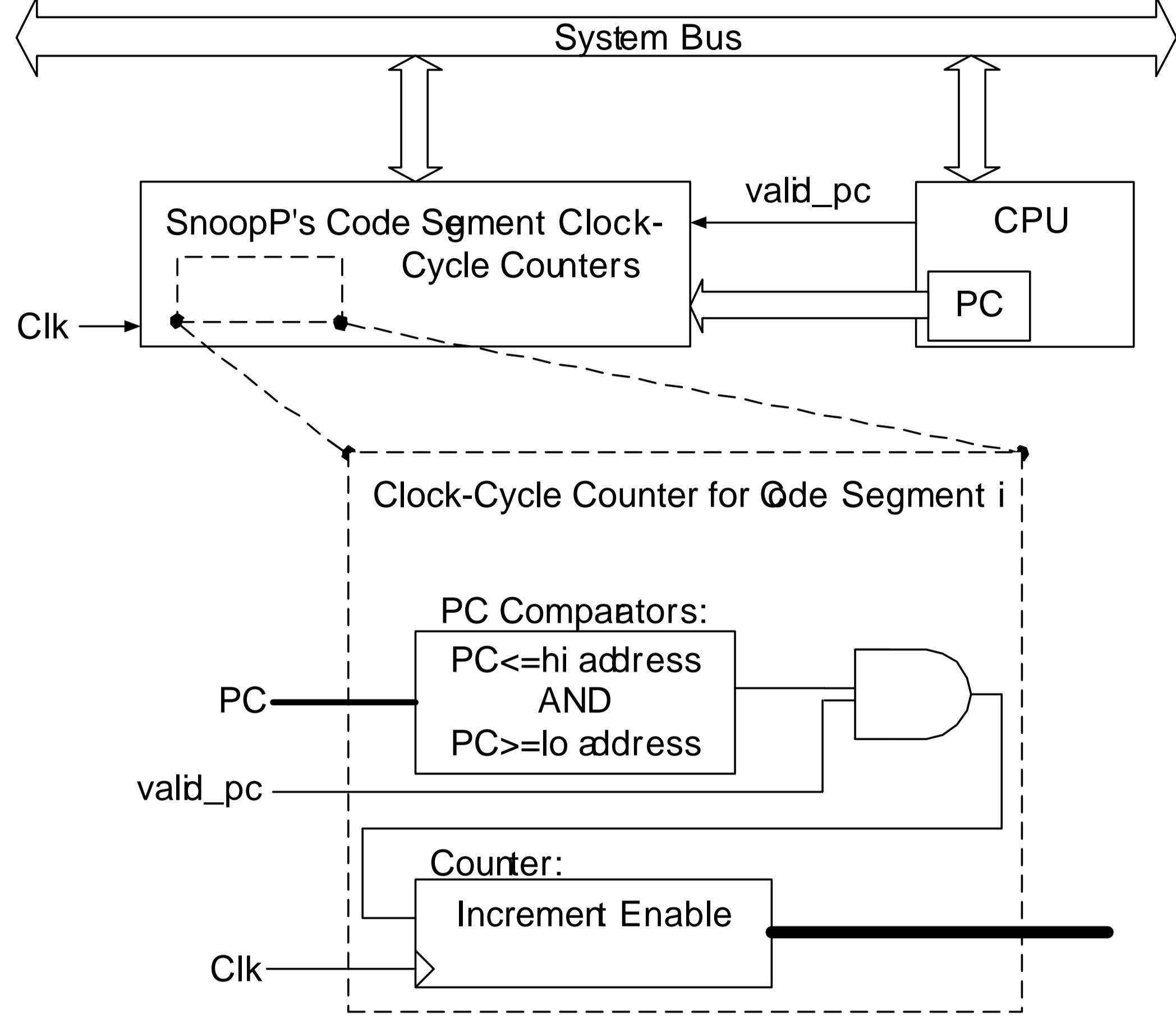
Decreasing technology sizes allow a complete embedded computing system to be implemented on a single FPGA.

A reconfigurable platform allows on-chip design tools to be used as part of the design process. They provide:

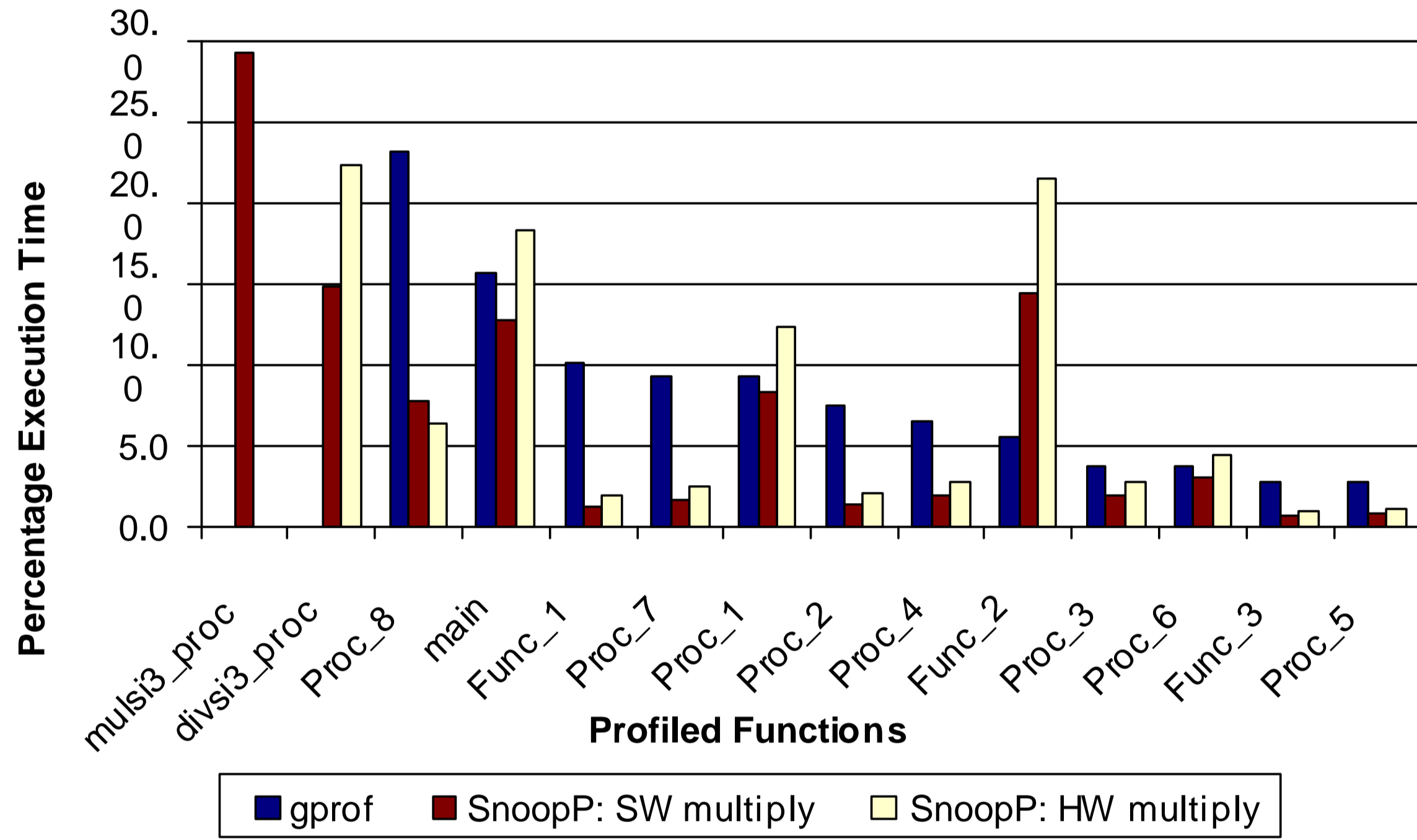
- 1) Real Time Information
- 2) Fast Clock-Cycle Accurate Results
- 3) Non-Intrusive Interface to Executing Software
- 4) Scalable Modular Design.

Furthermore, incorporating reconfigurability into the system design model can facilitate module reuse and adaptability to changing system specifications.

## SnoopP: A Snooping Profiler

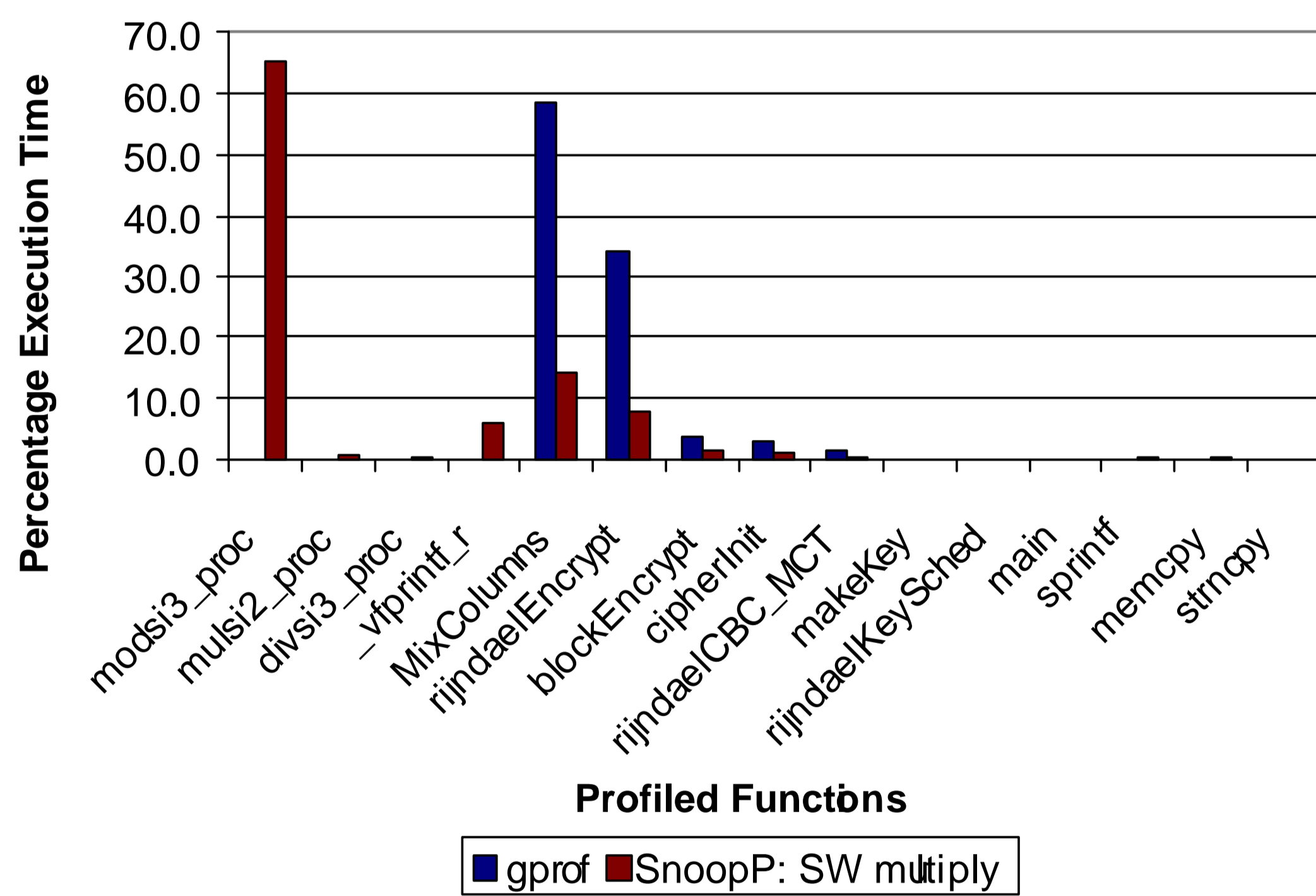


## GNU's gprof vs SnoopP Results Dhrystone Profile



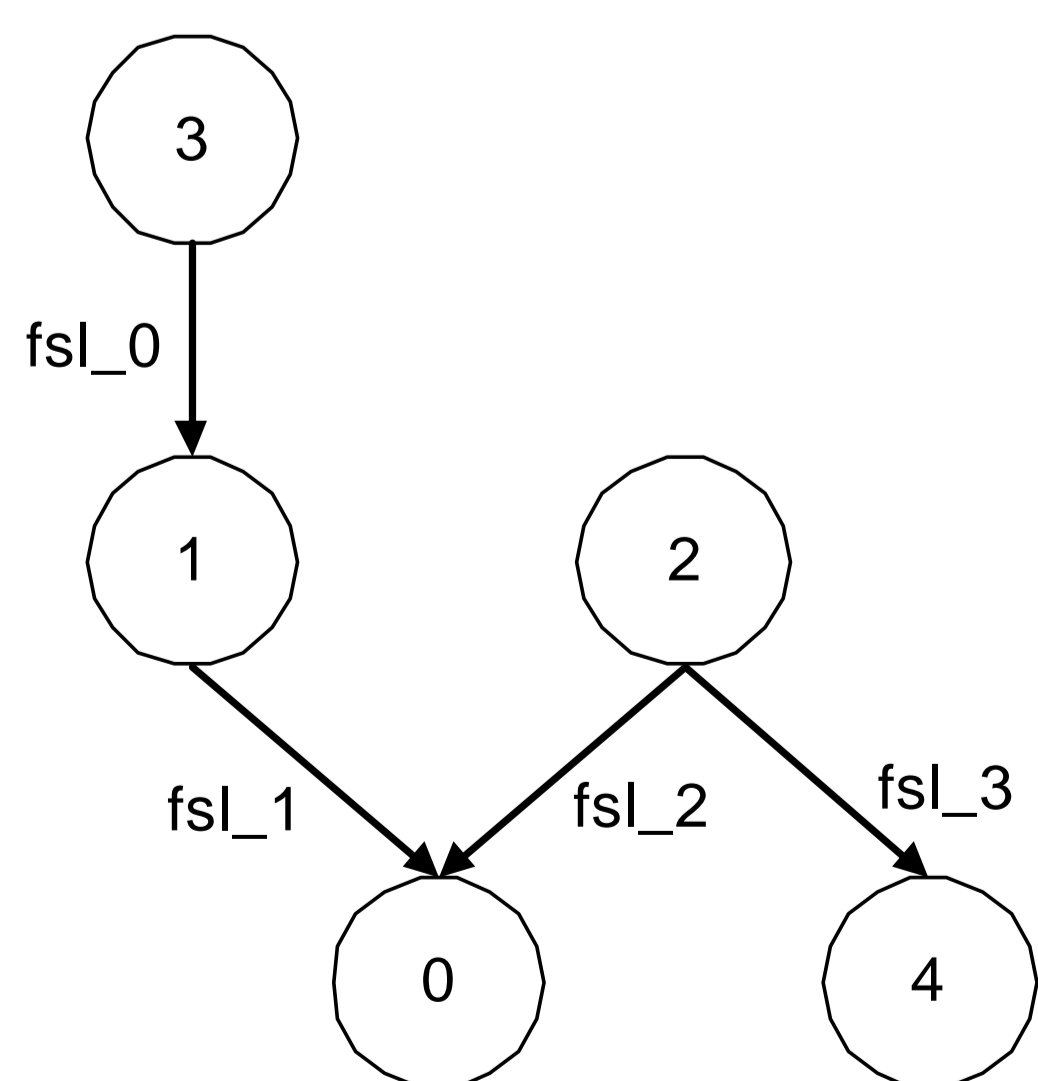
gprof results were normalized as it attributed 44% the execution time to gprof overhead.

## AES Profile



SnoopP profiled only 26% of the static code size and measured 96% of the execution time.

## Sample System



Configuration A: Equal processing delays for data produced/consumed by each processing link. Data is produced/consumed sequentially.

Configuration B: The processing delays for CE 0 are each reduced by 50% so that data can be consumed at twice the rate.

Configuration C: The processing delays for CE 2 is also reduced by 50% so that data is produced at twice the rate.

Counter 7 reports the total runtime of WOODSTOCK for each configuration in millions of clock cycles.

Each Configuration of the Sample System is run three times: first with 20 data packets passing through the system, then with 100 data packets, and finally with 200 data packets:

**Con A:** Configuration A illustrates that CE 0 is too slow and will stall the system.

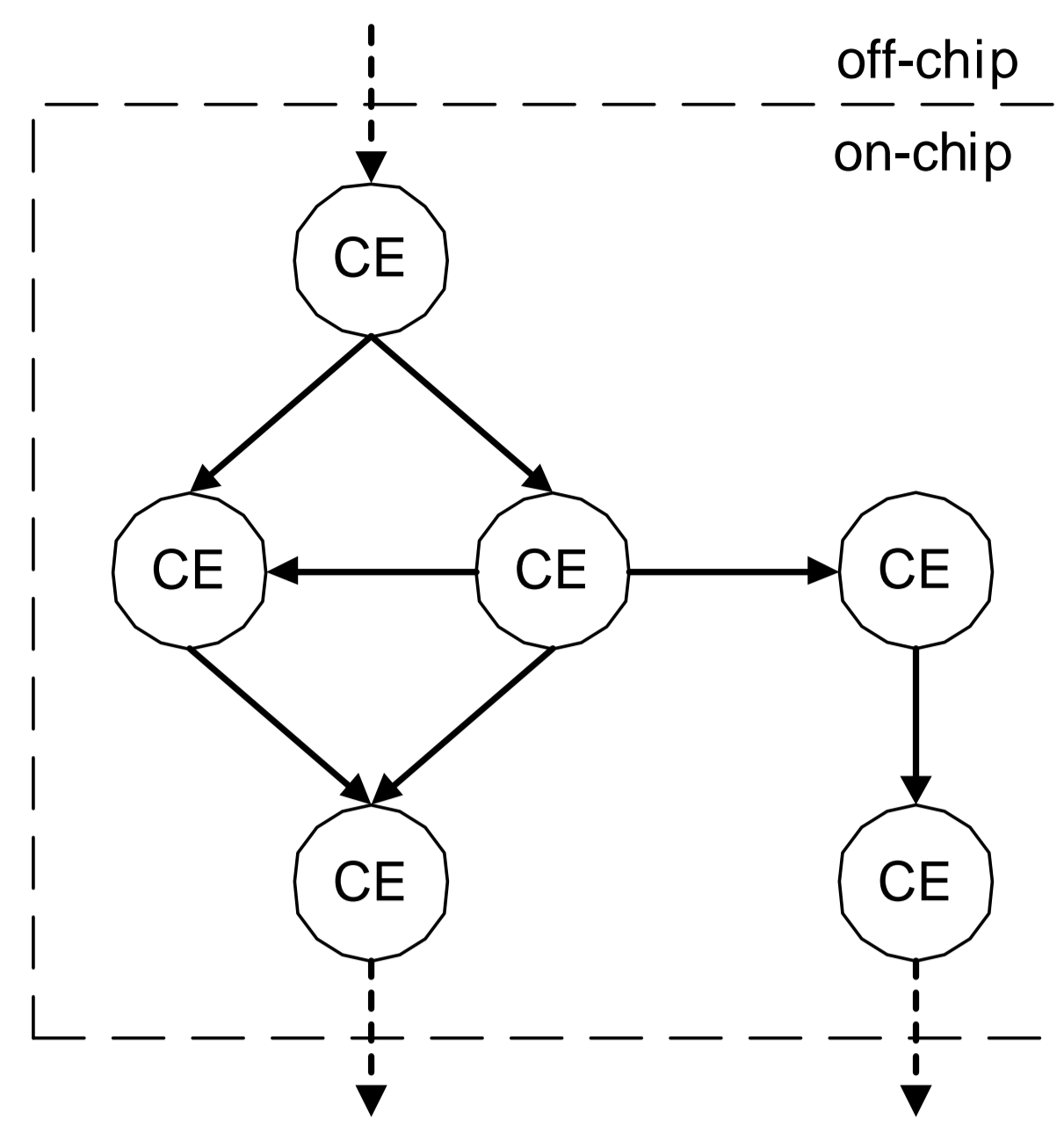
**Con B:** Configuration B suggests that CE 2 may also be too slow causing system stalls.

**Con C:** Configuration C represents an abalanced system.

## WOODSTOCK Profiling Results

Cnts	Enable Condition	Possible Meaning	20 Data Packets			100 Data Packets			200 Data Packets		
			Con A	Con B	Con C	Con A	Con B	Con C	Con A	Con B	Con C
0	fsl_1_full	CE 0 slow	21.4	10.1	0	83.7	82.2	0	91.8	91.1	0
1	fsl_2_full	CE 0 slow	0	0	0	0	0	0	0	0	0
2	fsl_0_full and (not fsl_1_full)	CE 1 slow	0	0	0	0.0	0.0	0	0.0	0.0	0
3	fsl_2_empty	CE 2 slow?	2.4	94.9	2.3	0.5	99.0	0.5	0.2	100.0	0.2
4	fsl_3_empty	CE 2 slow?	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
5	fsl_0_empty	CE 3 slow?	83.3	94.9	100.0	17.3	18.8	100.0	8.7	9.4	100.0
6	fsl_3_full	CE 4 slow	0	0	0	0	0	0	0	0	0
7	running	monitors running	672	632	352	3232	3192	1632	6432	6392	3232

## SIMPPL Model



Systems Integrating Modules with Predefined Physical Links (SIMPPL).

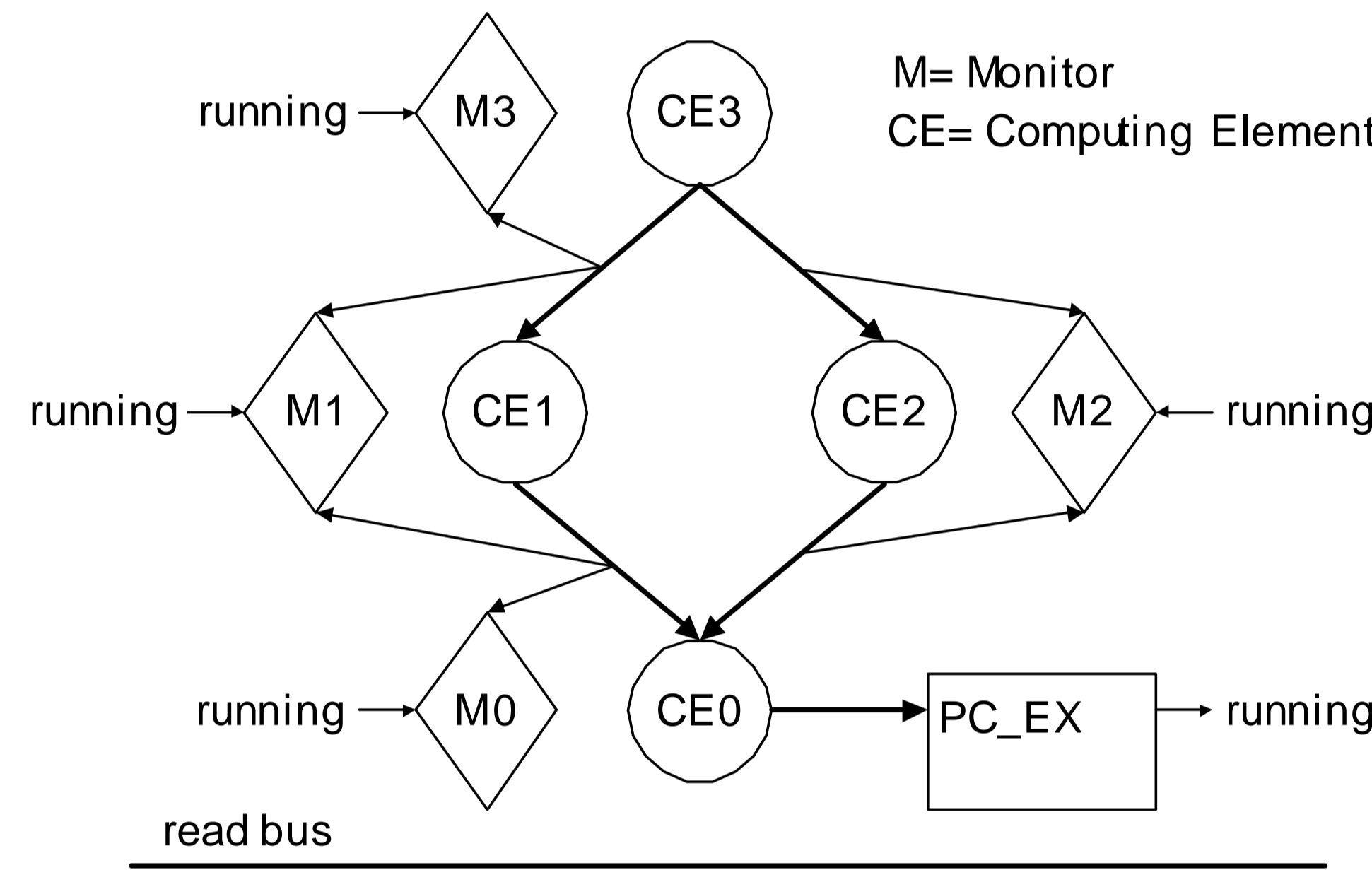
All on-chip connections between Computing Elements (CEs) are made using asynchronous FIFOs (indicated by solid arrows).

Each CE (Module) implements either a processor or a dedicated logic block.

Off-chip connections are left to be defined for the specific application (indicated using the arrows with the dotted lines).

## WOODSTOCK: Watching Over Data Streaming On Computing element links

### WOODSTOCK System Interface

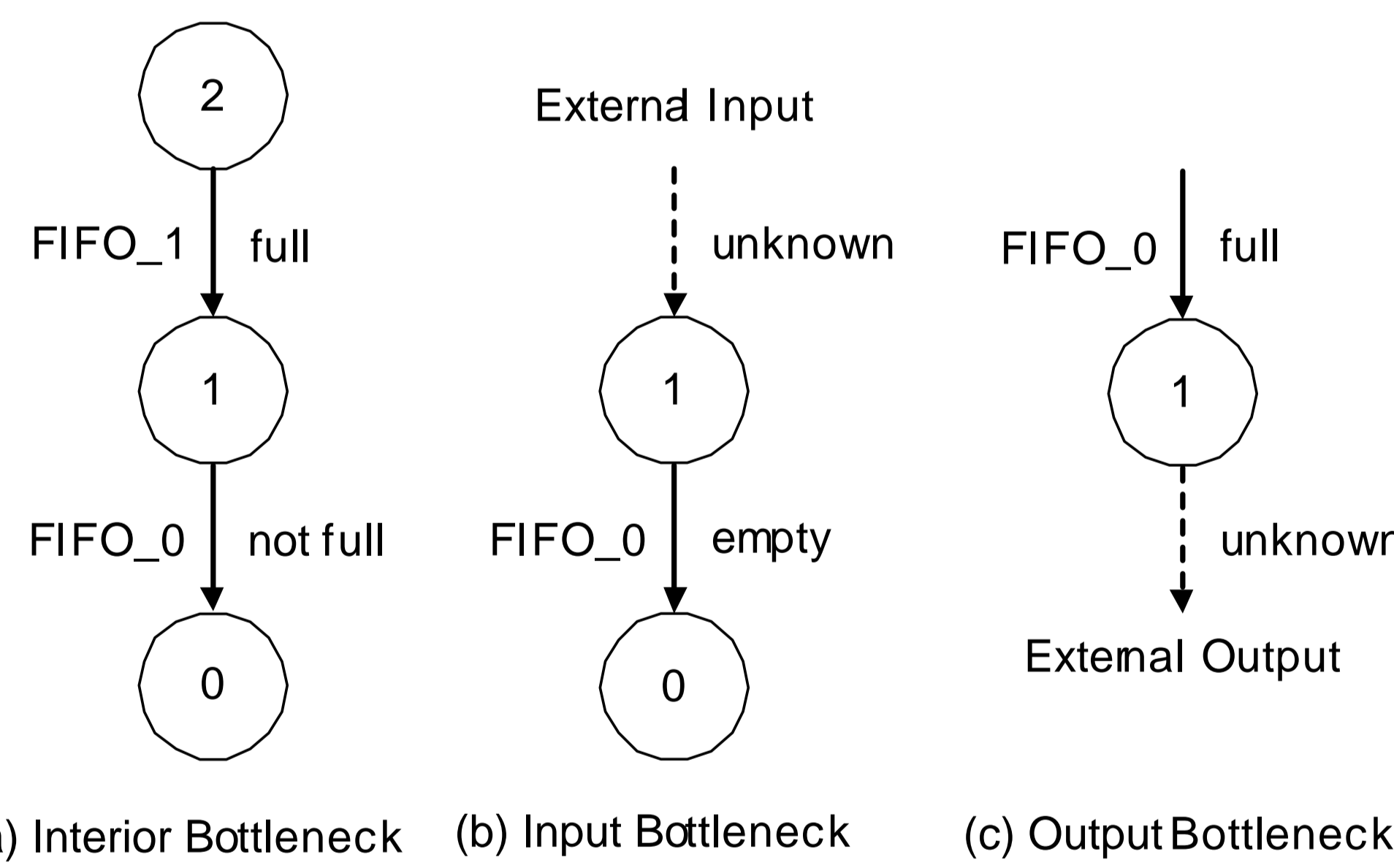


Using an input file that describes the system structure, we generate the system using a System Generator and profile it using WOODSTOCK.

The System Generator creates templates for each CE and the interconnections between CEs.

WOODSTOCK generates the Monitors and associates each one with a specific CE to monitor the communications and determine potential bottlenecks.

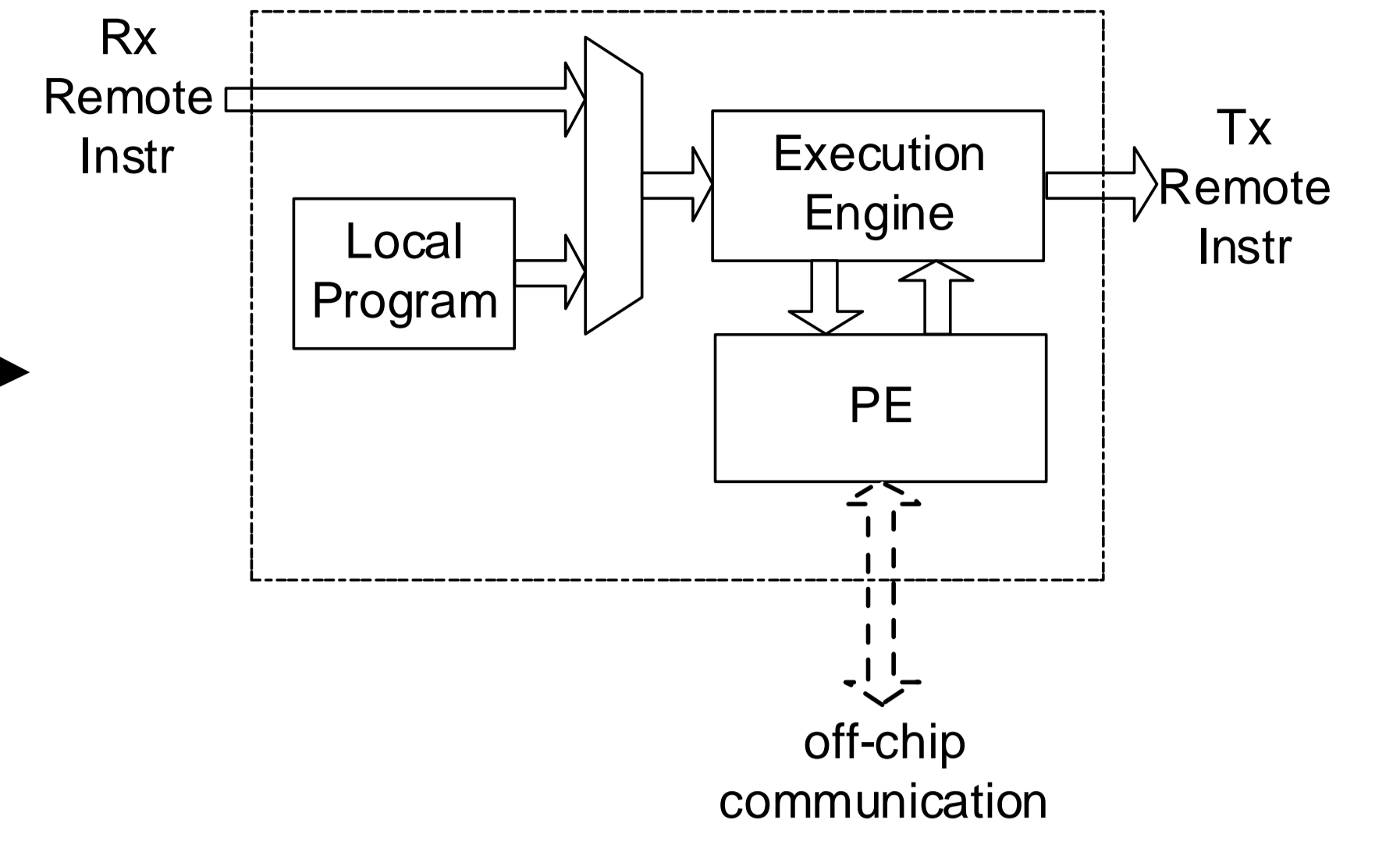
## Bottleneck Types



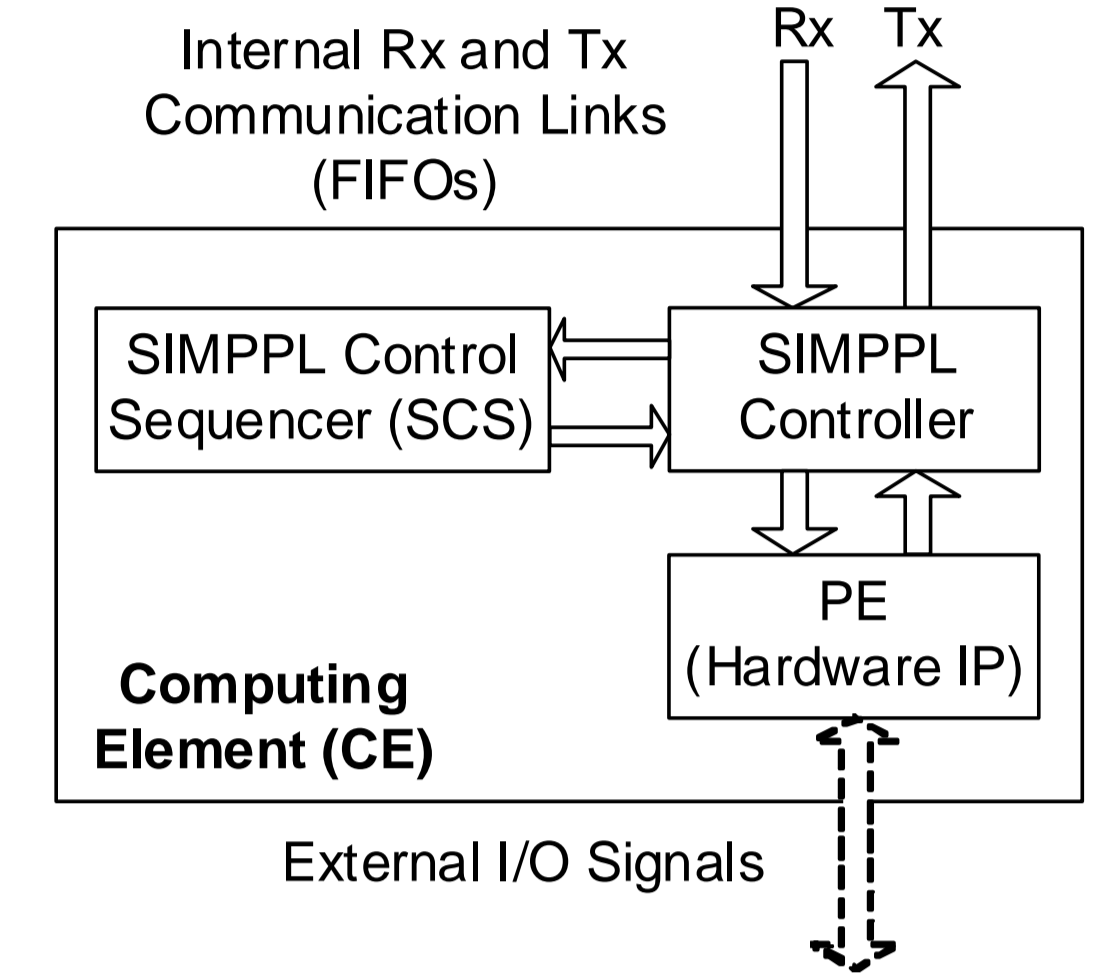
- CE 1 causes a bottleneck as it is unable to process data as fast as CE 2 produces it.
- This is a potential bottleneck. CE 0 may be starved for data or it may be processing at the same rate as CE 1.
- A bottleneck arises either due to CE 1 processing data too slowly or the external outputs throughput rate is too slow.

## Hardware CE

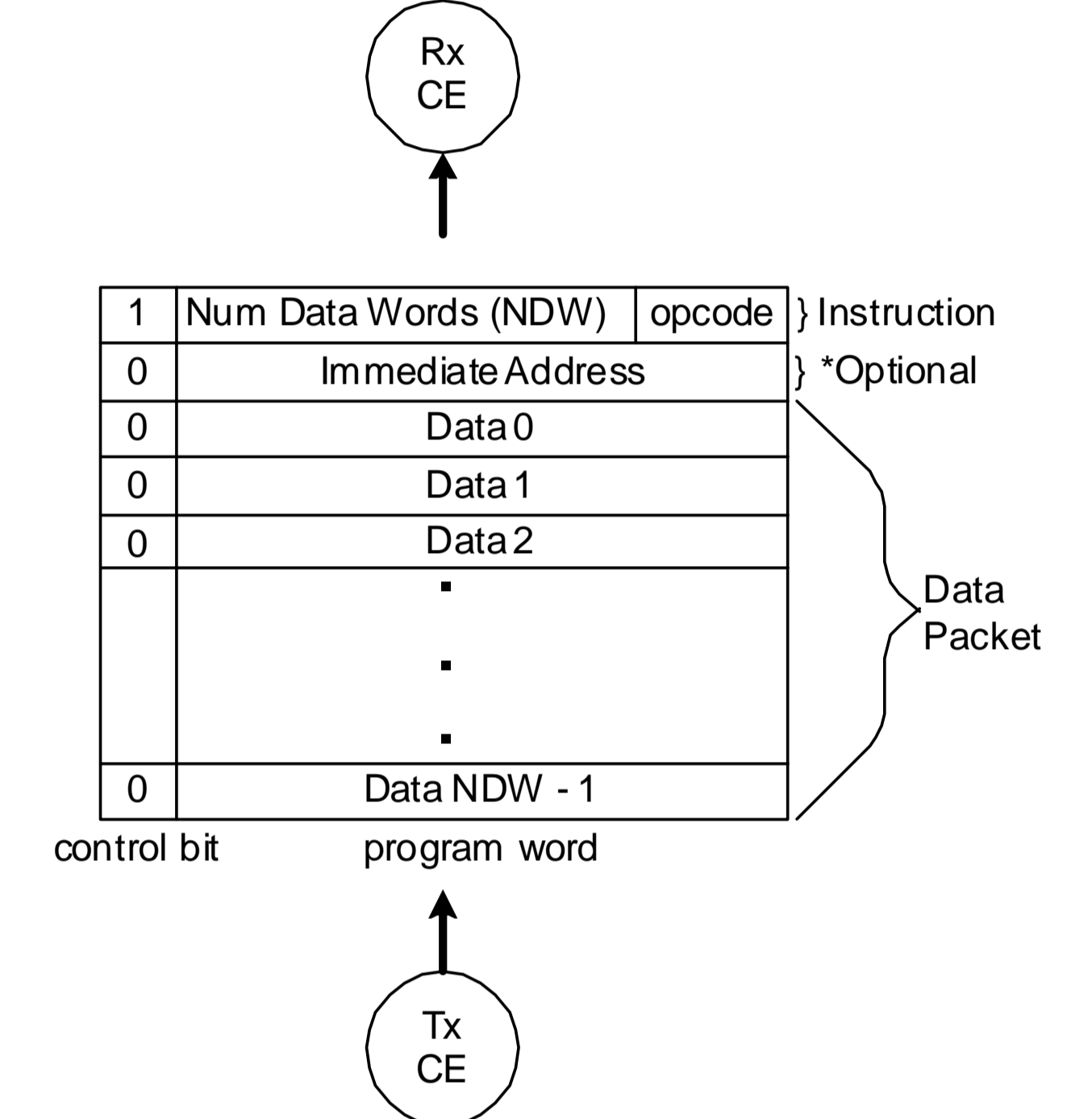
### Hardware CE Functionality



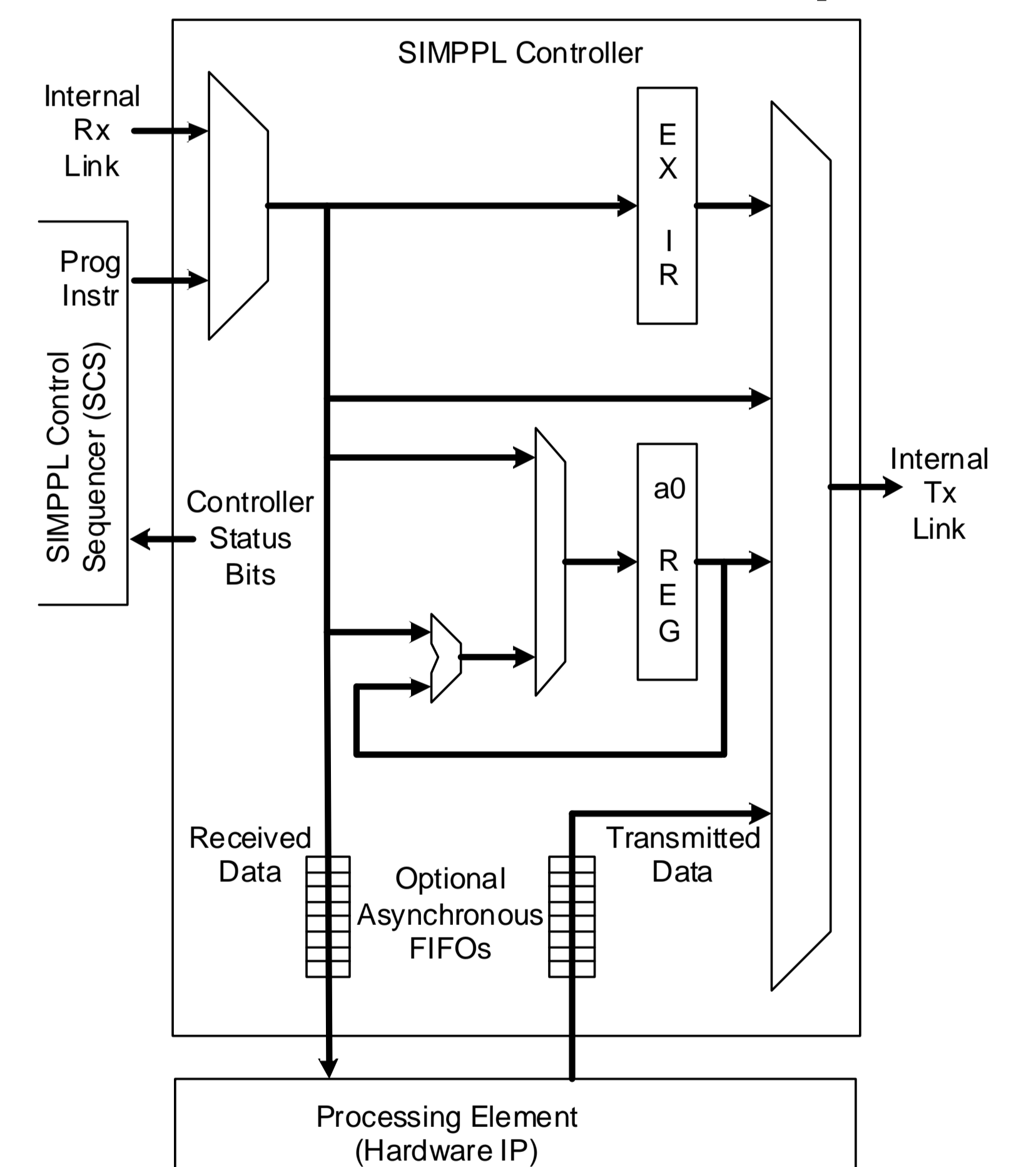
### Hardware CE Structure



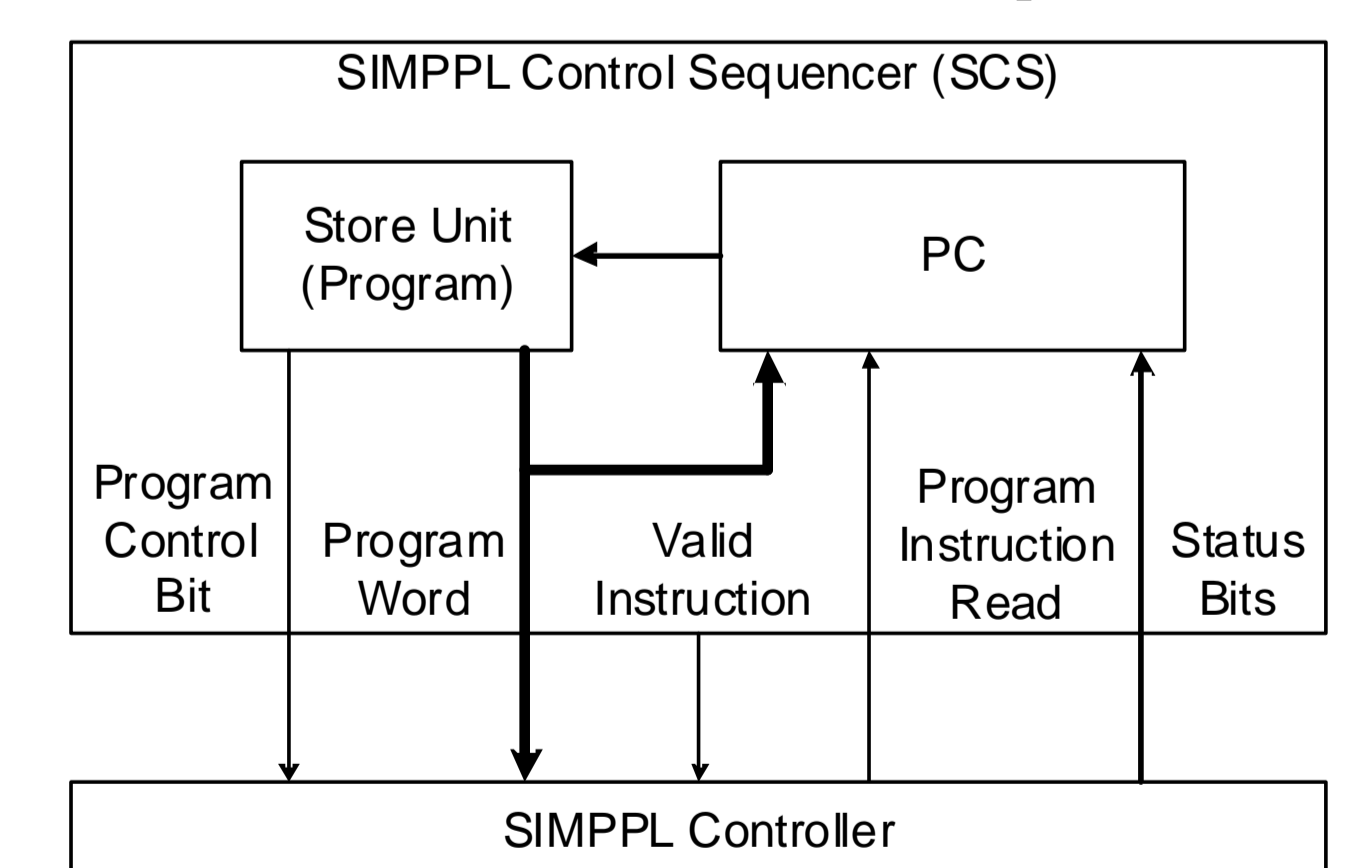
### Instruction Packet Format



### SIMPPL Controller Datapath



### SIMPPL Control Sequencer



Maximum Controller Overhead: 35 LUTs; 120 Flipflops; 7 cycles latency.

## Conclusions and Future Work

We created a software and system profiler for an on-chip design methodology.

The experimental results demonstrated that SnoopP was able to obtain more accurate information about the performance of the software than GNU's gprof.

WOODSTOCK is able to detect stall and bottlenecks to help the designer correctly repartition a system to improve system throughput.

SIMPPL provides a well defined framework for SoC design that reduces the complexity and necessary redesign for PE reuse.

The Hardware CE separates the PE's computation from the system-level communication and control to facilitate system integration.

Currently we are investigating how the SIMPPL framework affects the overall system design methodology and how to utilize on-chip debugging and verification to reduce system design time.