IWES 2018

Third Italian Workshop on Embedded Systems

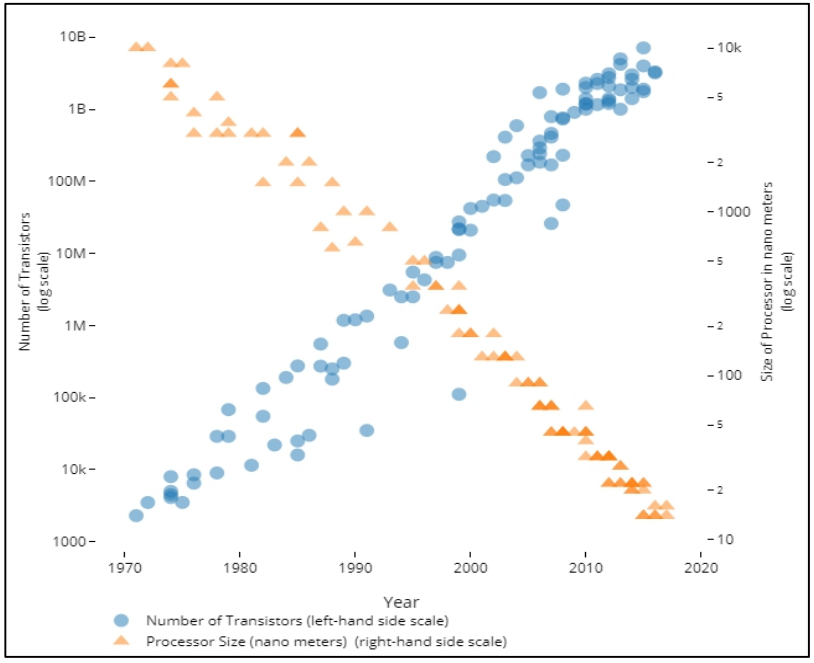
Siena – 13-14 September 2018

An FPGA-Based Scalable Hardware Scheduler For Data-Flow Models

Roberto Giorgi, Marco Procaccini, Farnam Khalili
University of Siena, Italy

The end of Dennard scaling…

- Engineering community forced to find new solutions to improve performance with a limited power budget by [1]
  - ✓ Stop increasing clock frequency
  - ✓ Shifting to multicore processors

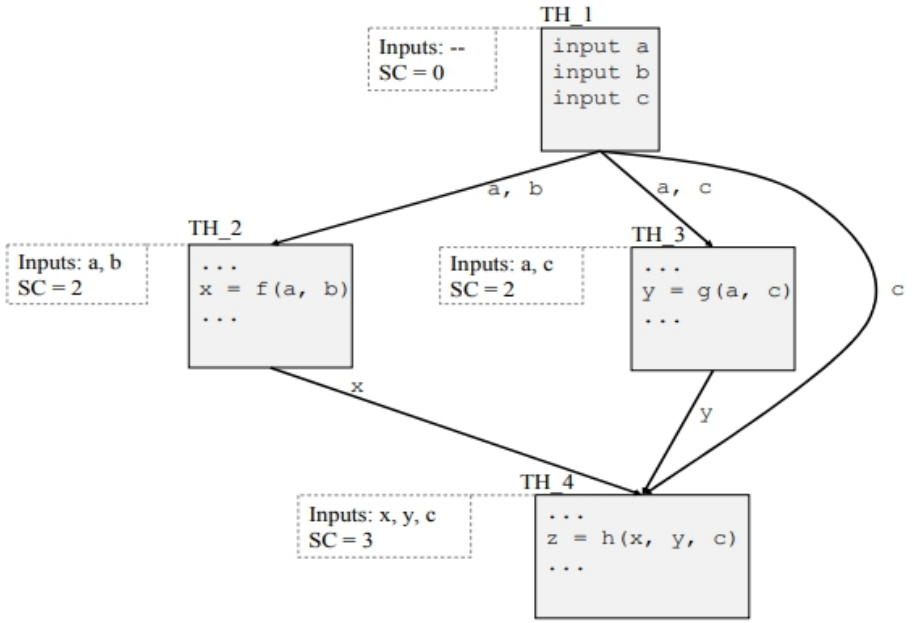


Moore's law 2018 – Source: Wikipedia

- Programming limitations to exploit full performance still remains…

The “DF-Threads” Data-Flow execution model

- The “DF-Threads” Data-Flow execution model is capable of taking advantage of the full parallelism offered by multicore systems[2][3][4][5][6][7]
  - ✓ Execution relies on data-dependencies
  - ✓ Parallel execution of data independent paths

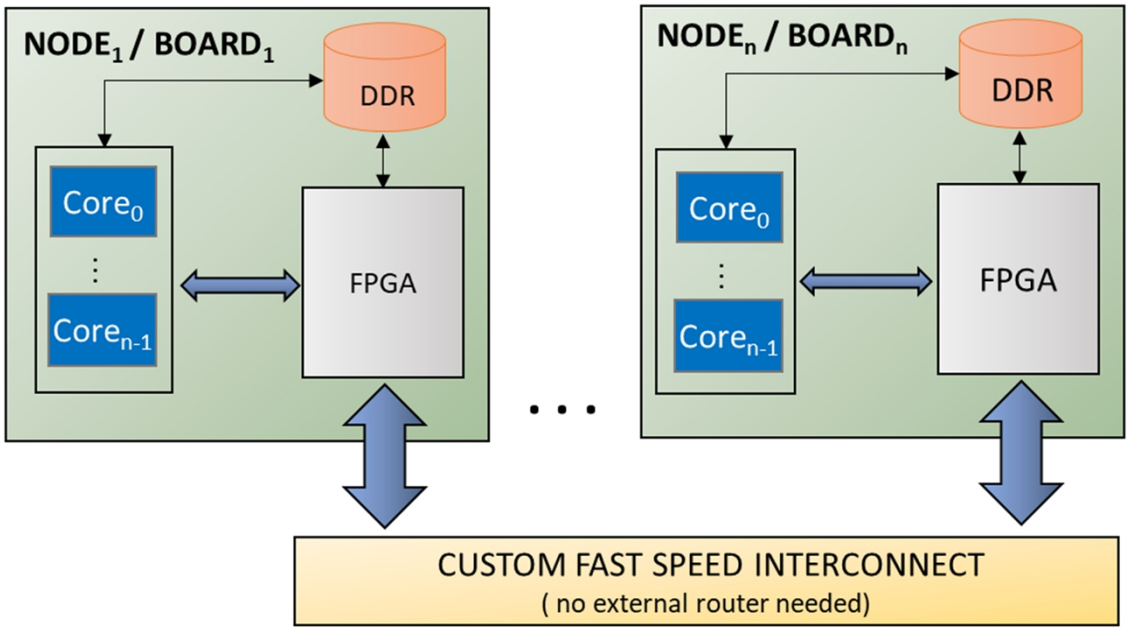


Hybrid Data-Flow Model

- DF-Threads based execution does not need to totally replace the conventional general purpose processors (GPP)
- Hybrid Model based on GPP and Field Programmable Gate Arrays (FPGA)
  - ✓ GPP cores are suitable for legacy or OS
  - ✓ FPGA can easily provide an efficient parallel execution via DF-Threads

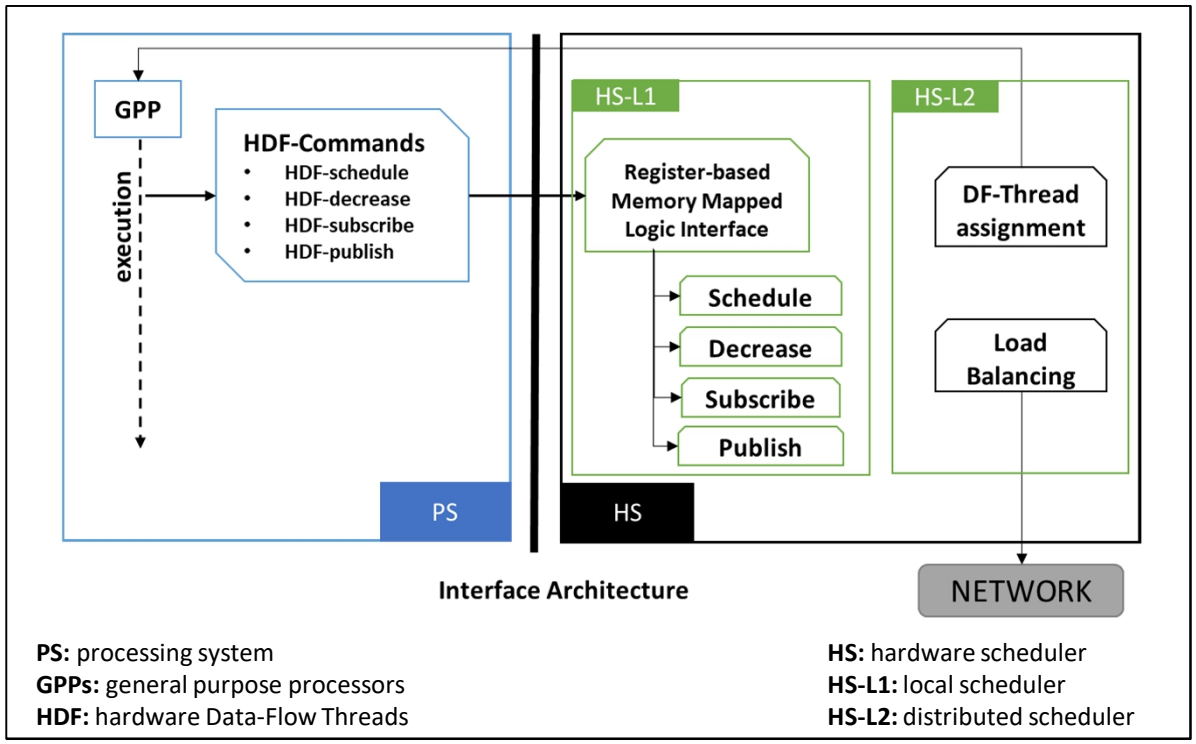
System Design

- A possible architecture to enable an easy distribution of the Data-Flow Threads (DF-Threads) among multiple core and multiple nodes [8]



The Idea

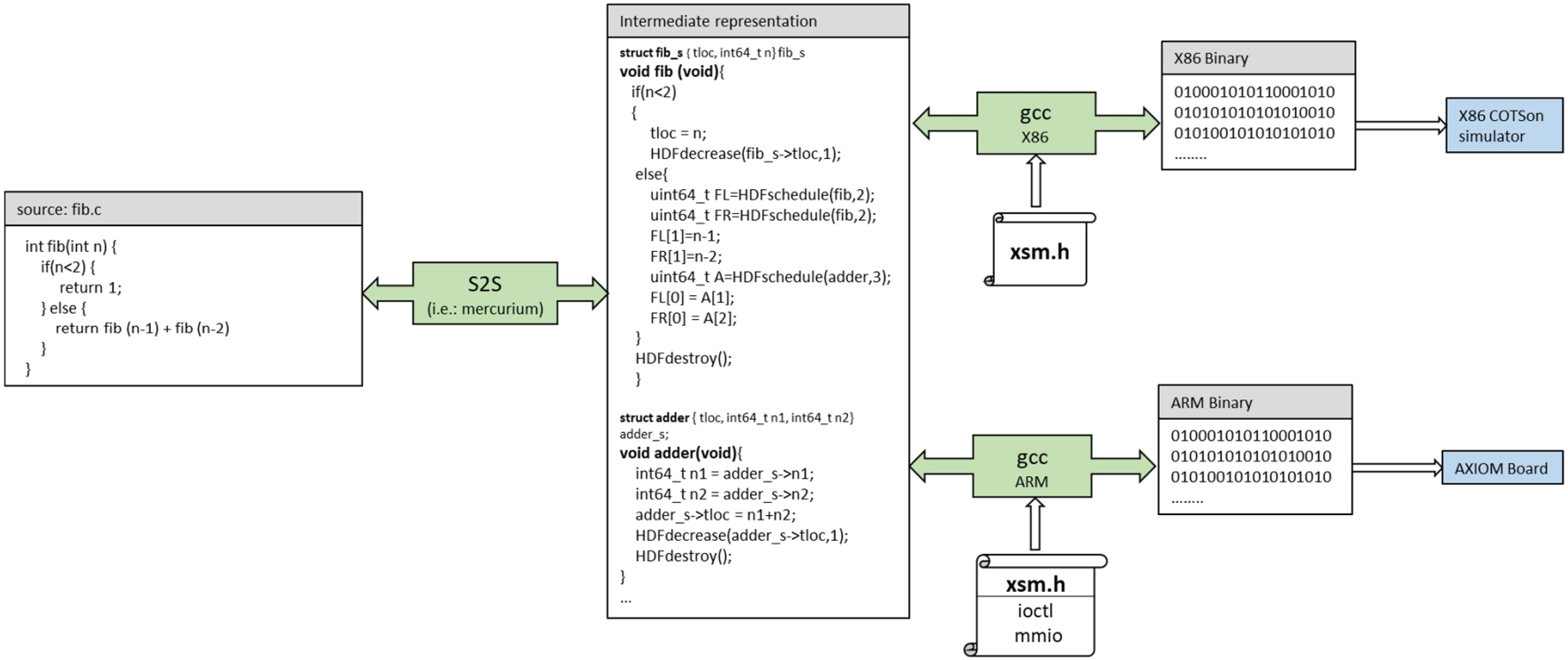
- Improving the execution of the Data-Flow Threads scheduling, by implementing an Hardware Scheduler (HS) on FPGA [9][10]



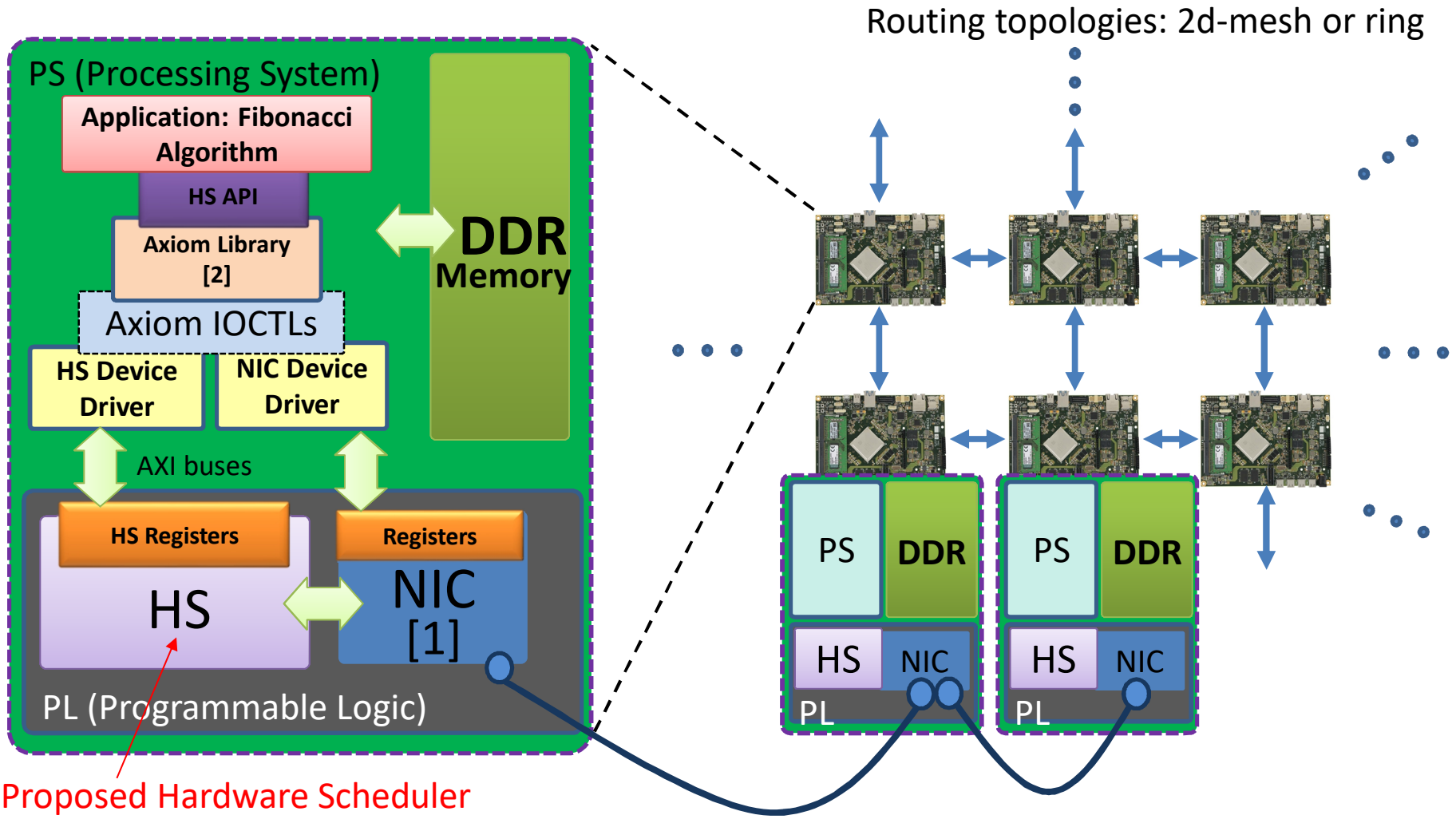
- The GPP:
  - ✓ Asynchronous APIs
  - ✓ Schedule DF-Threads
  - ✓ Execute DF-Threads
- The HS
  - ✓ Retrieves meta-information
  - ✓ Provides ready HDF-Threads
  - ✓ Distribute HDF-Threads on network

Compilation and testing flow

- Testing Environment
  - ✓ COTSon Simulator [11]
  - ✓ AXIOM Board [12]



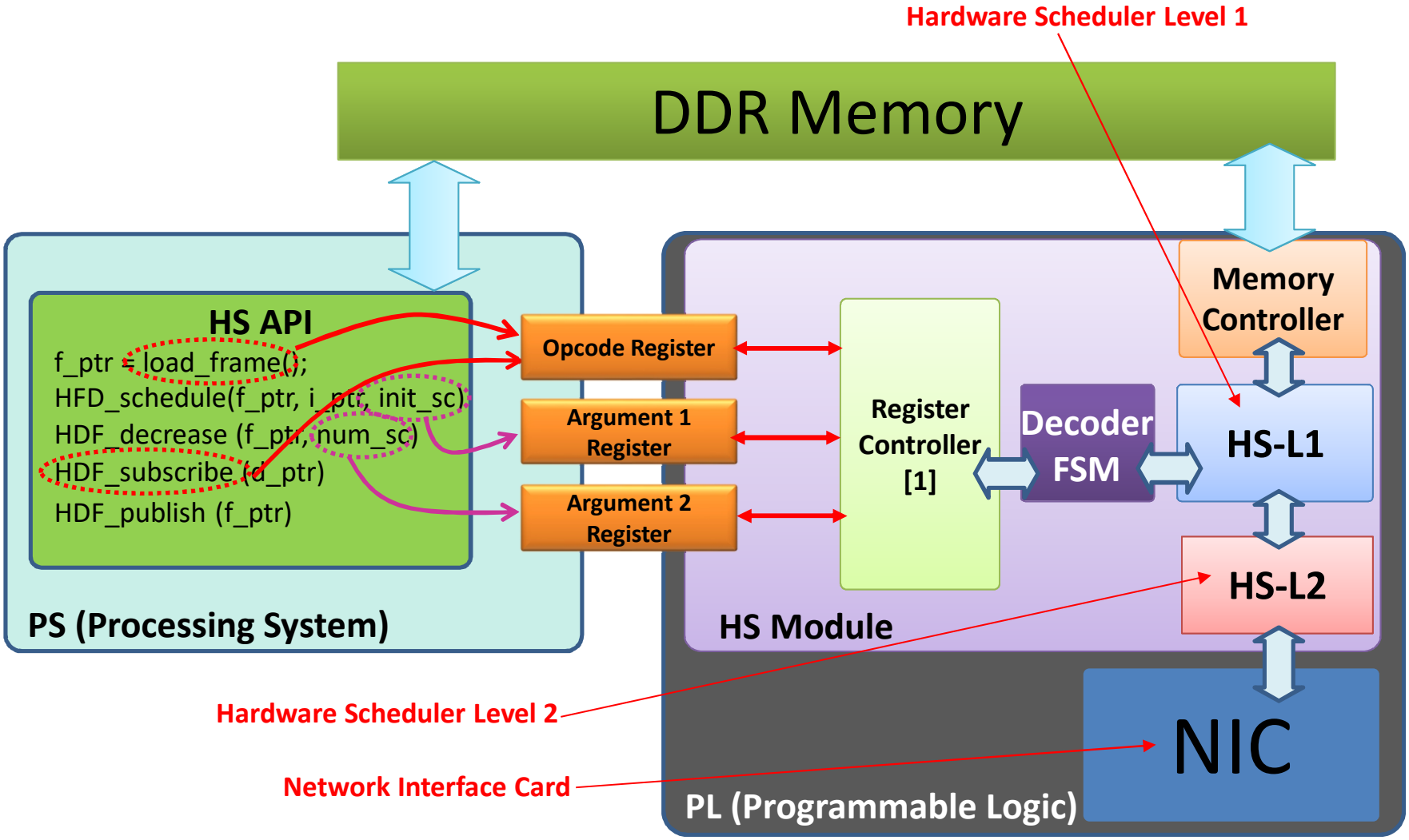
System Abstraction in a Perspective



[1] Vasileios Amourgianos-Lorentzos. “Efficient network interface design for low cost distributed systems” Master Thesis, 2017 at Technical University of Crete as part of the FORTH Axiom program.

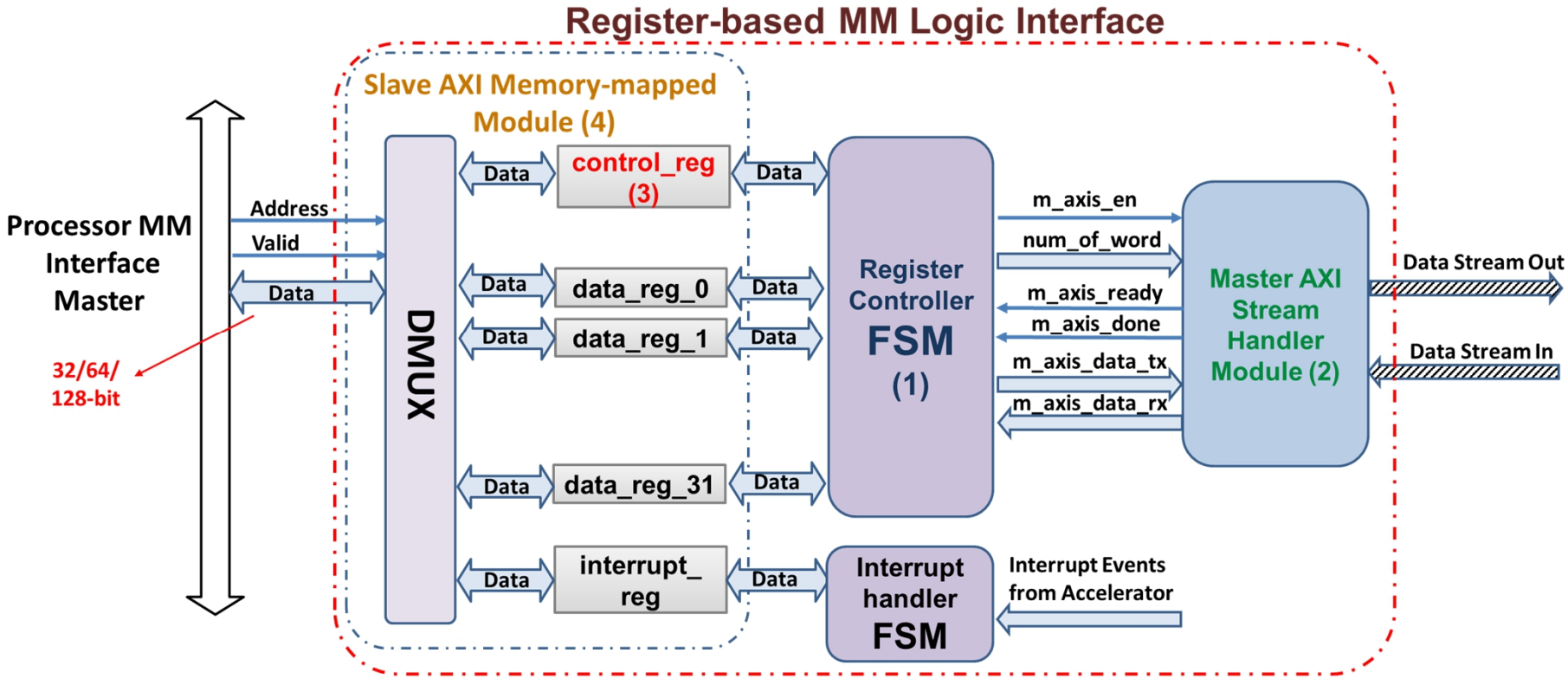
[2] Evidence Embedding Technology, 2017, “https://github.com/evidence

Hardware Scheduler (HS) Primitives



[1] F. Khalili, M. Procaccini and R. Giorgi. “Reconfigurable logic interface architecture for cpu-fpga accelerators.“ In HiPEAC ACACES-2018, pp. 1,4. Fiuggi, Italy, July 2018. Poster.

Register Controller [2]

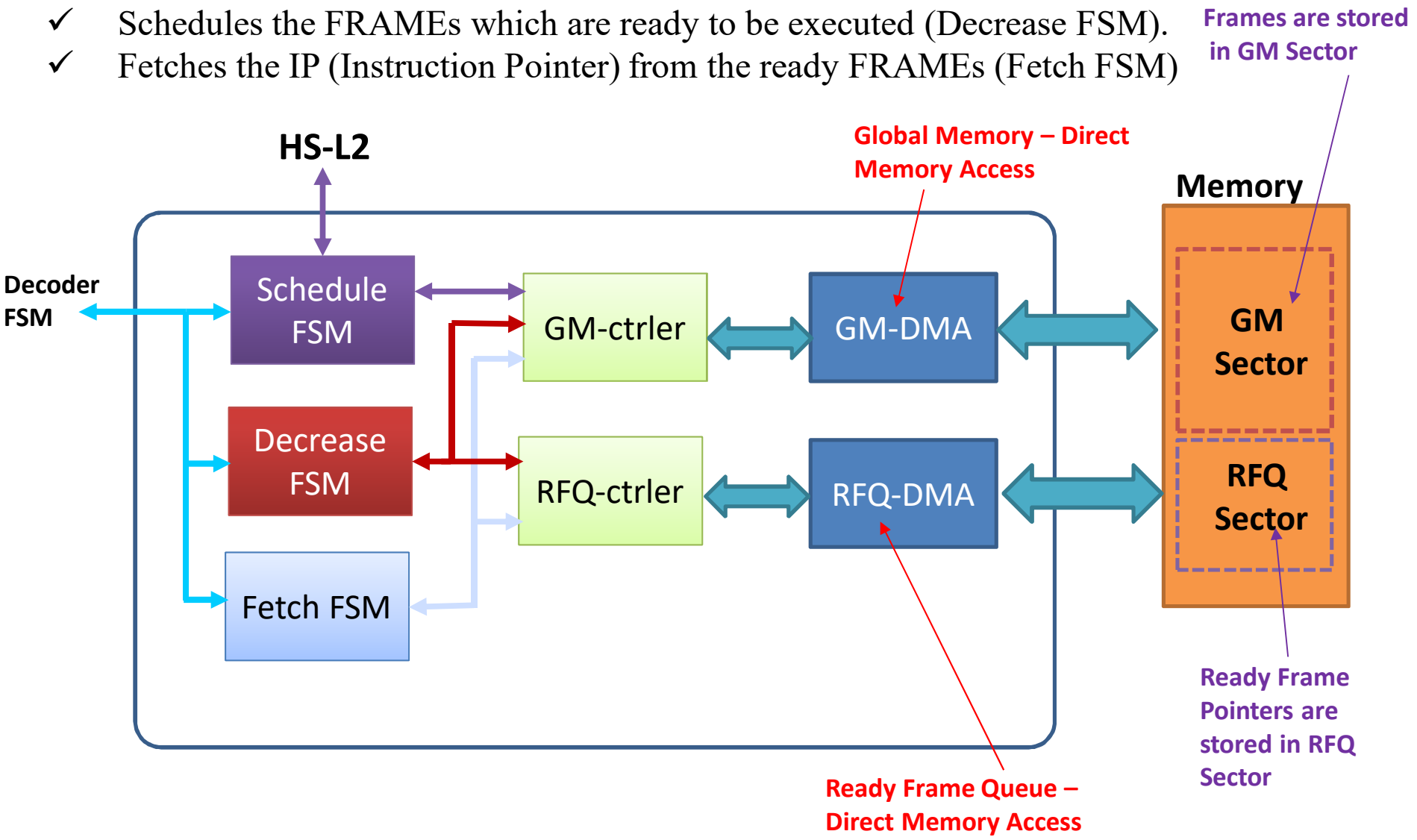


- ✓ The Write/Read access of each registers are separately controllable through the 'Control' register.
- ✓ The **Register Controller FSM (1)** is responsible to control **Master AXI Stream Handler Module (2)** and exchange data between AXI Stream and AXI memory mapped Domains.
- ✓ **Register Controller FSM (1)** also polls **control_reg (3)** and checks corresponding bit fields of each register if it is configured as write access or read access to set the direction of the data.

[2] F. Khalili, M. Procaccini and R. Giorgi. "Recongurable logic interface architecture for cpu-fpga accelerators." In HiPEAC ACACES-2018, pp. 1{4. Fiuggi, Italy, Julyy 2018. Poster.

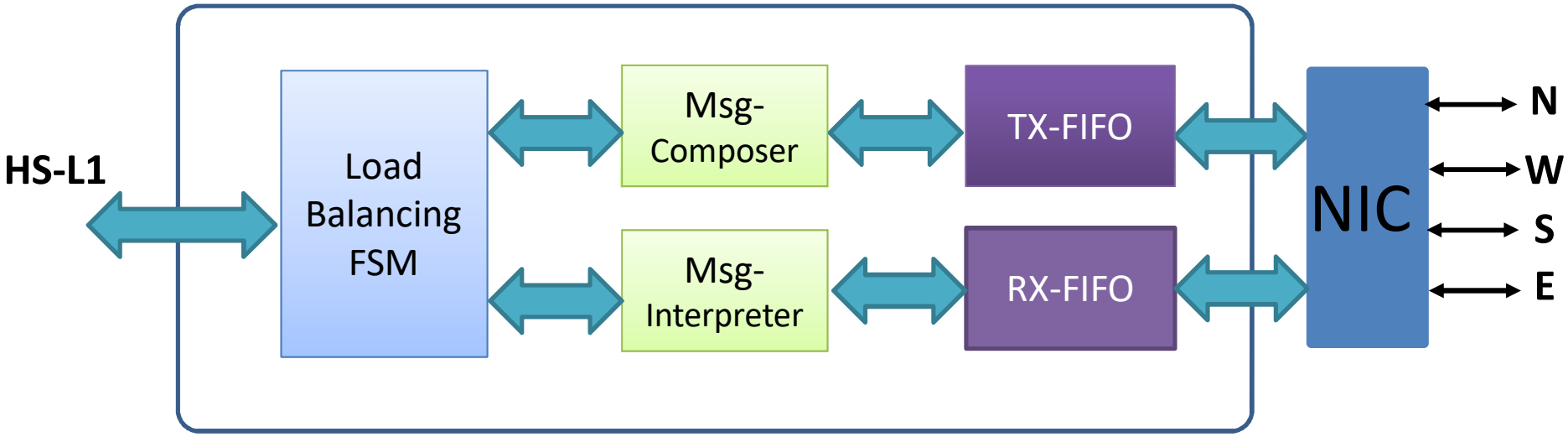
HS-L1 (Hardware Scheduler Level 1)

- ✓ Retrieves meta-information of FRAMEs (Schedule FSM)
- ✓ Schedules the FRAMEs which are ready to be executed (Decrease FSM).
- ✓ Fetches the IP (Instruction Pointer) from the ready FRAMEs (Fetch FSM)

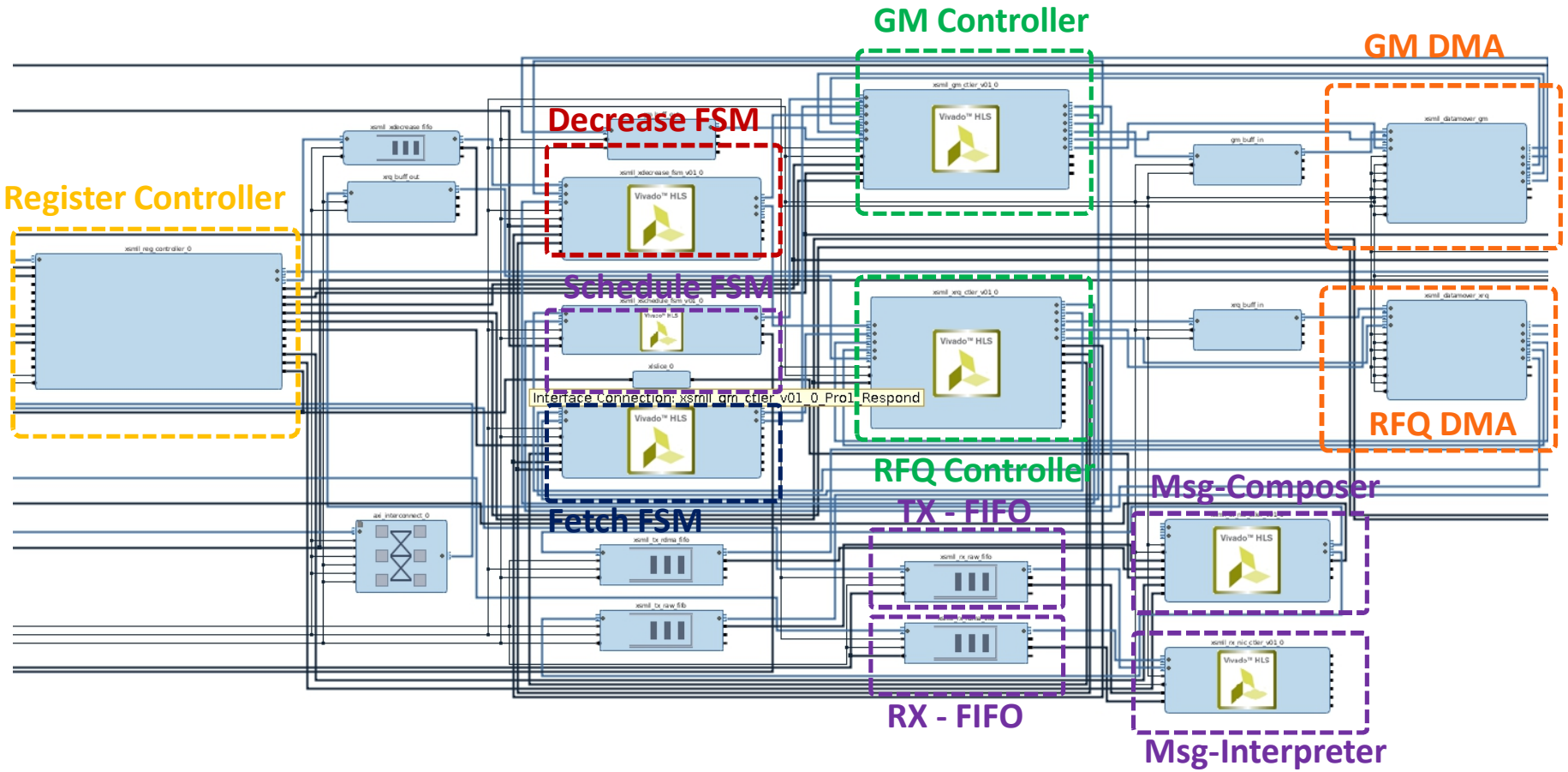


HS-L2 (Hardware Scheduler Level 2)

- ✓ Distribute FRAMEs in order to balance the loads throughout the network.
  - Work-stealing from remote nodes.
  - Off-load the works to remote nodes



Design Snippets



Evaluation – Execution Cycles

Operation	Data Width	Number Of Clock Cycles (PL).	
		Worst	Best
FIFO Enqueue/Dequeue	**64 bits**	**2**	**1**
Global Memory Write (DDR4)	16 bytes	48	40
Global Memory Read (DDR4)	16 bytes	38	38
Ready Queue Write	32 bits	48	40
Ready Queue Read	32 bits	44	44

Instruction Name	Delay Contributors	Number of Clock Cycles (PL).	
		Worst	**Best**
HDF-Schedule	*Total*	*49*	*40*
	DMA IP	48	39
	Decoder FSM	1	1
HDF-Decrease	*Total*	*89*	*43*
	DMA IP	86	40
	Decoder FSM	3	3
HDF-Fetch	*Total*	*85*	*34*
	DMA IP	*82*	*31*
	Fetch FSM	*3*	*3*

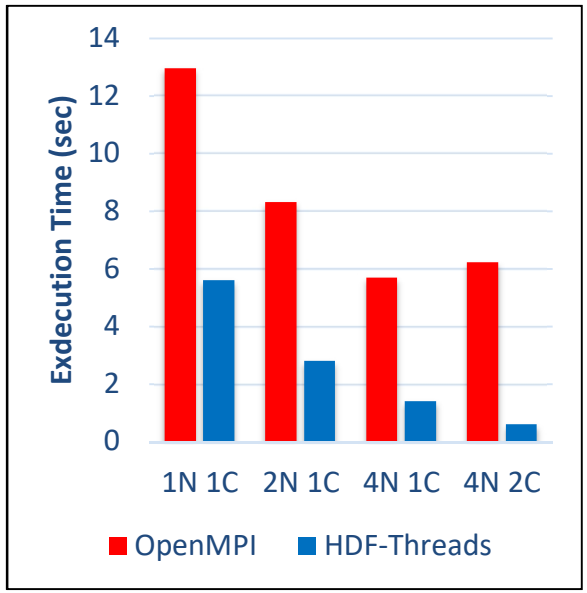
Evaluation – Resource Utilization

✓ Extracted resource utilization from Vivavo Design Suit 2016.4.
- Axiom board Zynq UltraScale+ XCZU9EG platform.

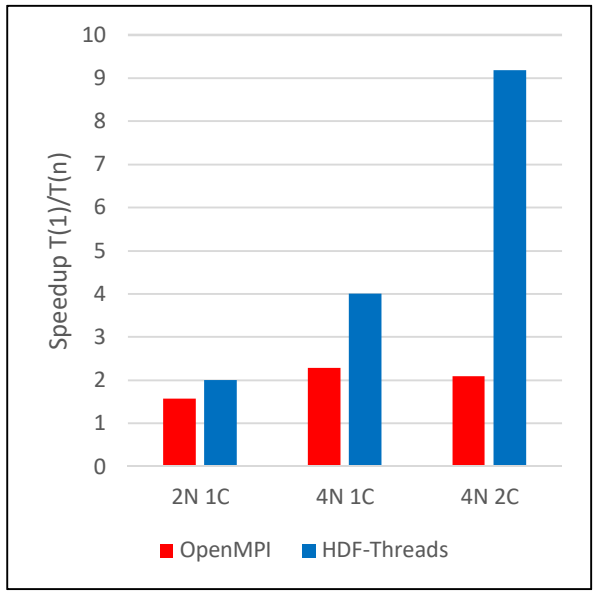
PL Units	Number of Units	Available	Utilization %
LUT	20357	274080	7.43
LUTRAM	2876	144000	2.00
FF	26116	548160	4.76
BRAM	49.50	912	5.43
IO	27	204	13.24
GT	2	16	12.50
BUFG	6	404	1.49

Results

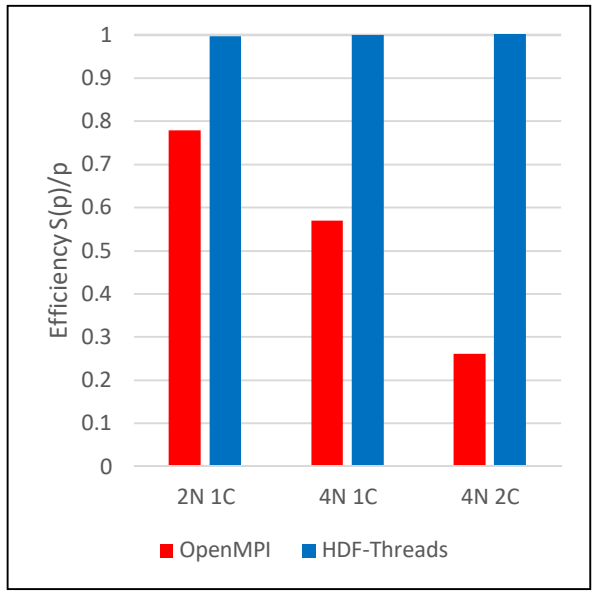
- HDF-Threads vs OpenMPI – Matrix Multiply Test 512+8



Execution Time



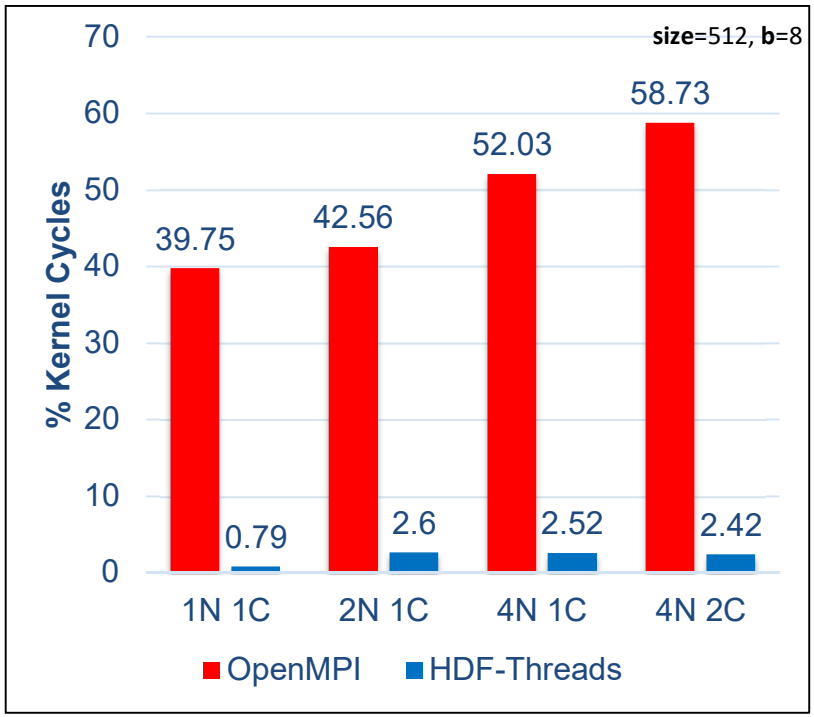
Speedup



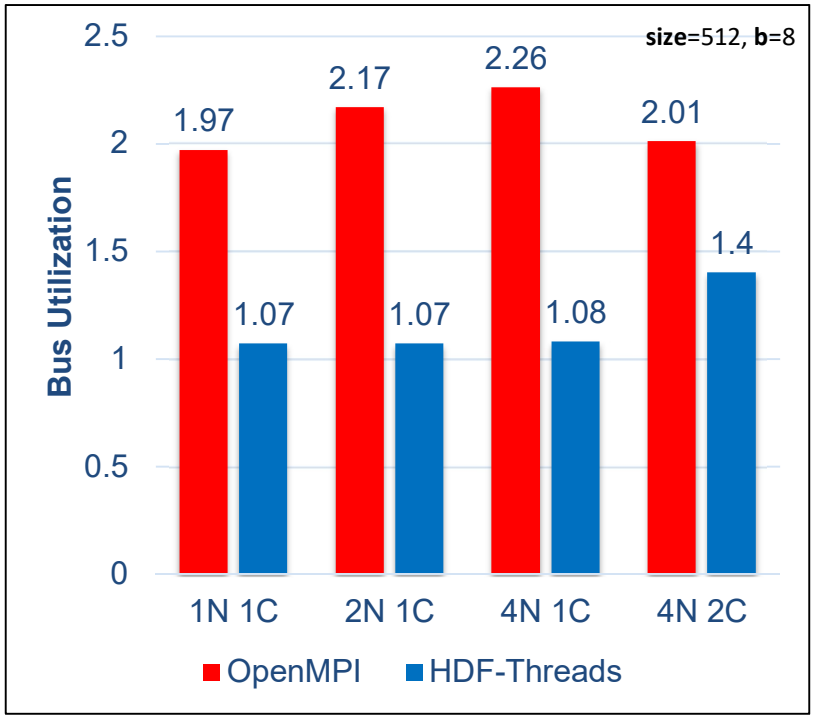
Efficiency

Results

- HDF-Threads vs OpenMPI – Matrix Multiply Test



Kernel Cycles



Bus Utilization

References

[1] Frank, D. J., Dennard, R. H., Nowak, E., Solomon, P. M., Taur, Y., & Wong, H. S. P. (2001). Device scaling limits of Si MOSFETs and their application dependencies. *Proceedings of the IEEE*, *89*(3), 259-288.

[2] Mondelli, Andrea, et al. "Dataflow support in x86_64 multicore architectures through small hardware extensions." *Digital System Design (DSD), 2015 Euromicro Conference on*. IEEE, 2015

[3] Dennis, J. B. (1980). Data flow supercomputers. *Computer*, (11), 48-56.

[4] Giorgi, R., & Faraboschi, P. (2014, October). An introduction to DF-Threads and their execution model. In *Computer Architecture and High Performance Computing Workshop (SBAC-PADW), 2014 International Symposium on* (pp. 60-65). IEEE.

[5] Verdoscia, L., Vaccaro, R., & Giorgi, R. (2014, August). A clockless computing system based on the static dataflow paradigm. In *Data-Flow Execution Models for Extreme Scale Computing (DFM), 2014 Fourth Workshop on* (pp. 30-37). IEEE.

[6] Giorgi, R., Popovic, Z., & Puzovic, N. (2007, October). DTA-C: A decoupled multi-threaded architecture for CMP systems. In *Computer Architecture and High Performance Computing, 2007. SBAC-PAD 2007. 19th International Symposium on* (pp. 263-270). IEEE.

[7] Kavi, K. M., Giorgi, R., & Arul, J. (2001). Scheduled dataflow: Execution paradigm, architecture, and performance evaluation. *IEEE Transactions on Computers*, *50*(8), 834-846.

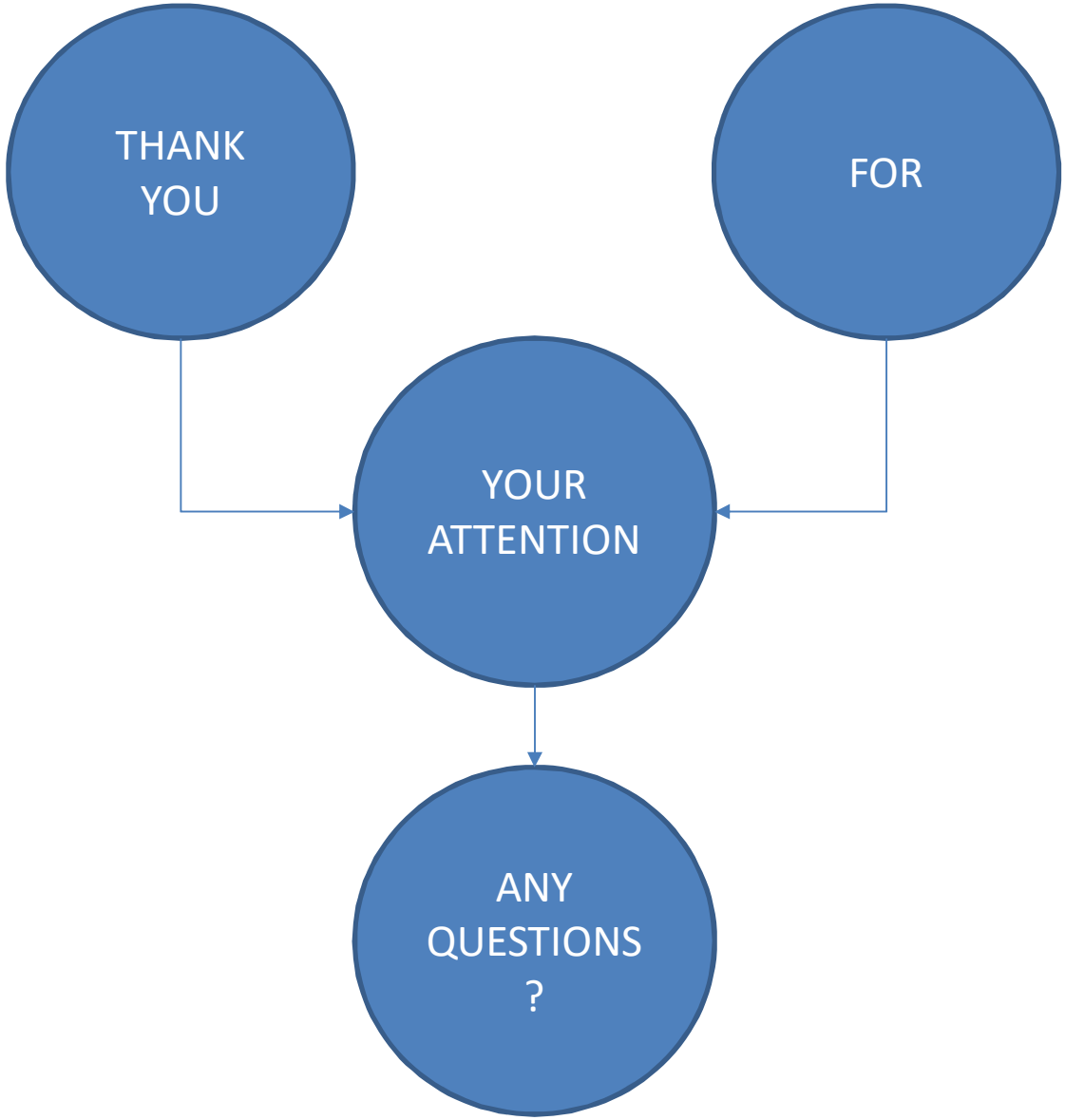
[8] Procaccini, M., Giorgi, R. (2017). A Data-Flow Execution Engine for Scalable Embedded Computing. *HiPEAC ACACES-2018.*

[9] Procaccini, M., Khalili, F., Giorgi, R. (2018). An FPGA-based Scalable Hardware Scheduler for Data-Flow Models. *HiPEAC ACACES-2018.*

[10] Khalili, F., Procaccini, M., Giorgi, R. (2018). Reconfigurable Logic Interface Architecture for CPU-FPGA Accelerators. *HiPEAC ACACES-2018.*

[11] Argollo, E., Falcón, A., Faraboschi, P., Monchiero, M., & Ortega, D. (2009). COTSon: infrastructure for full system simulation. *ACM SIGOPS Operating Systems Review*, *43*(1), 52-61.

[12] Theodoropoulos, D., Mazumdar, S., Ayguade, E., Bettin, N., Bueno, J., Ermini, S., ... & Montefoschi, F. (2017). The AXIOM platform for next-generation cyber physical systems. *Microprocessors and Microsystems*, *52*, 540-555.


THANK YOU
FOR
YOUR ATTENTION
ANY QUESTIONS ?