ACCELERATOR INTEGRATION IN HETEROGENEOUS ARCHITECTURES

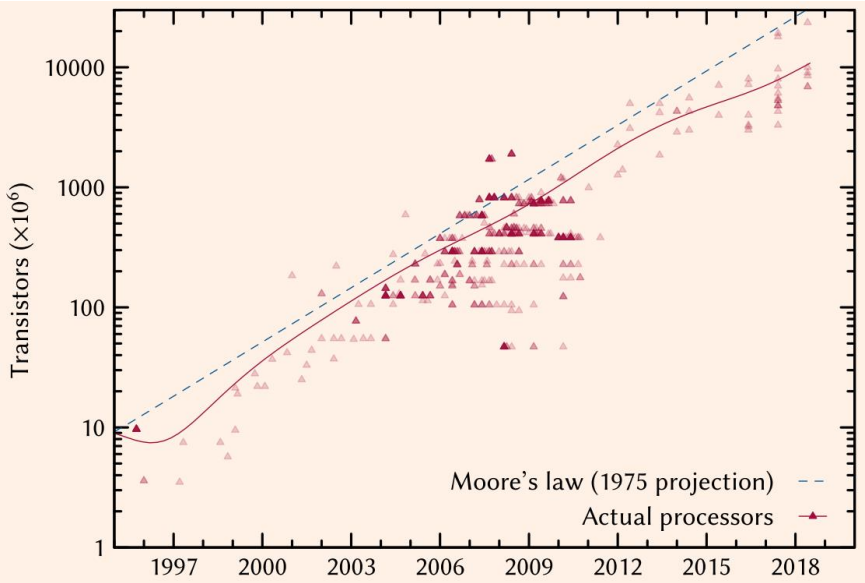
Candidacy Exam

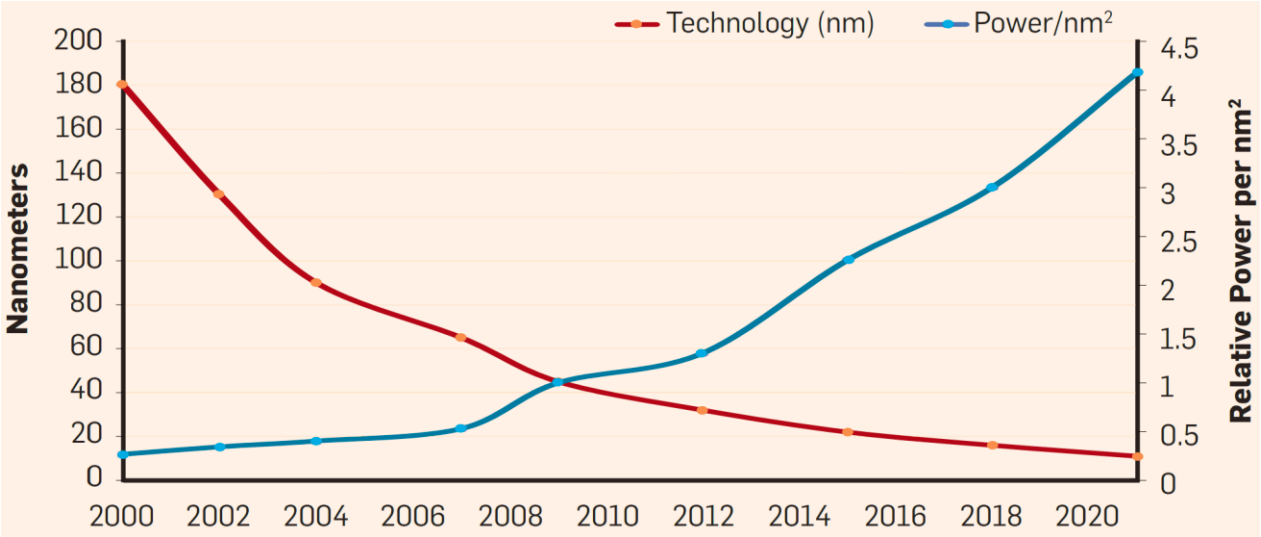
Davide Giri

January 24, 2020

A TALE OF TWO SCALINGS

- **Moore's Law** is slowing down
  - Transistors struggle to keep up
- **Dennard scaling** is dead
  - Power density has been increasing



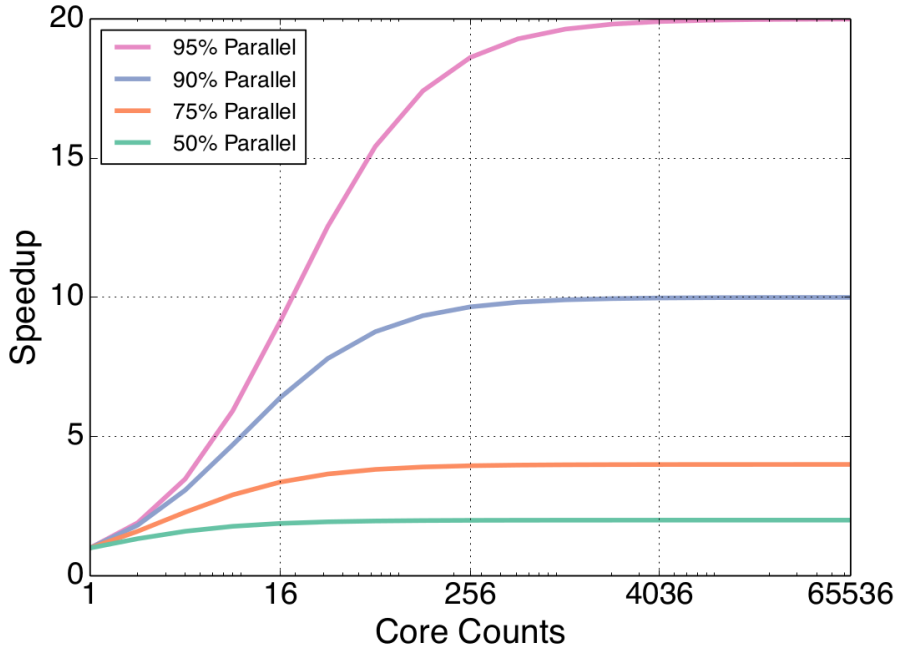


Graphs from *"E. G. Cota, PhD Dissertation Defense, 2019"*

[Shao 2015]

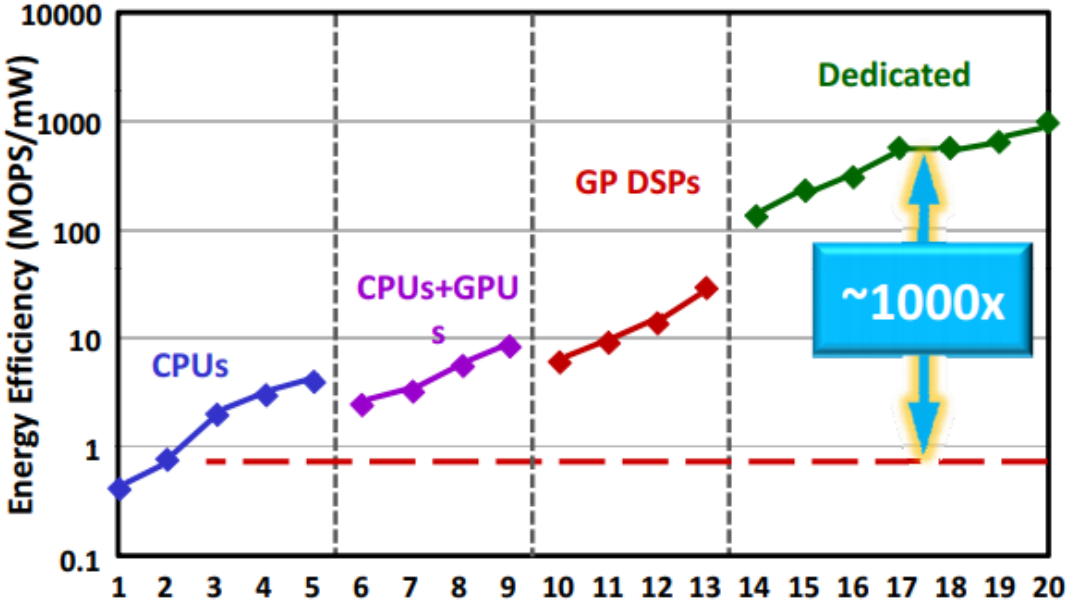
WHY ACCELERATORS?

- **Multi-core** processors are limited by Amdahl's law



[Shao 2015]

- **Specialization** makes a difference

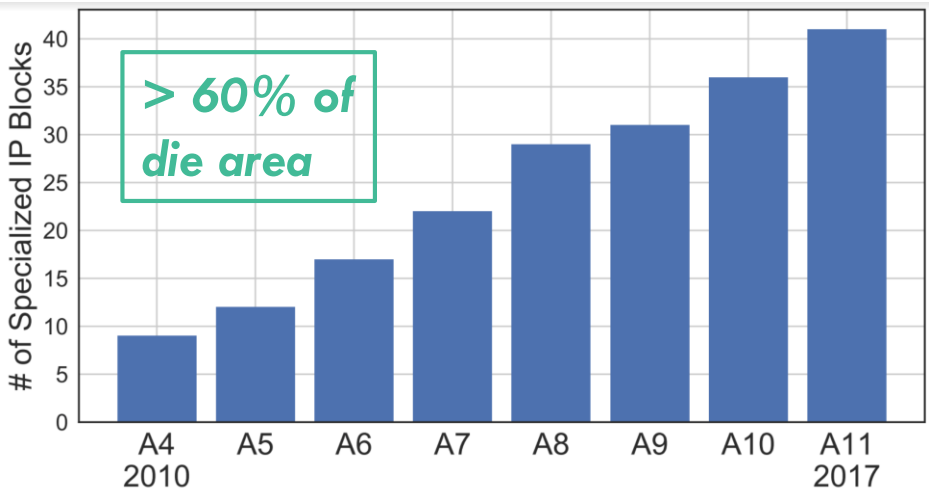


Graph from "*M. Horowitz, Computing's Energy Problem (and what we can do about it), in ISSCC, 2014*"

THE ERA OF ACCELERATORS

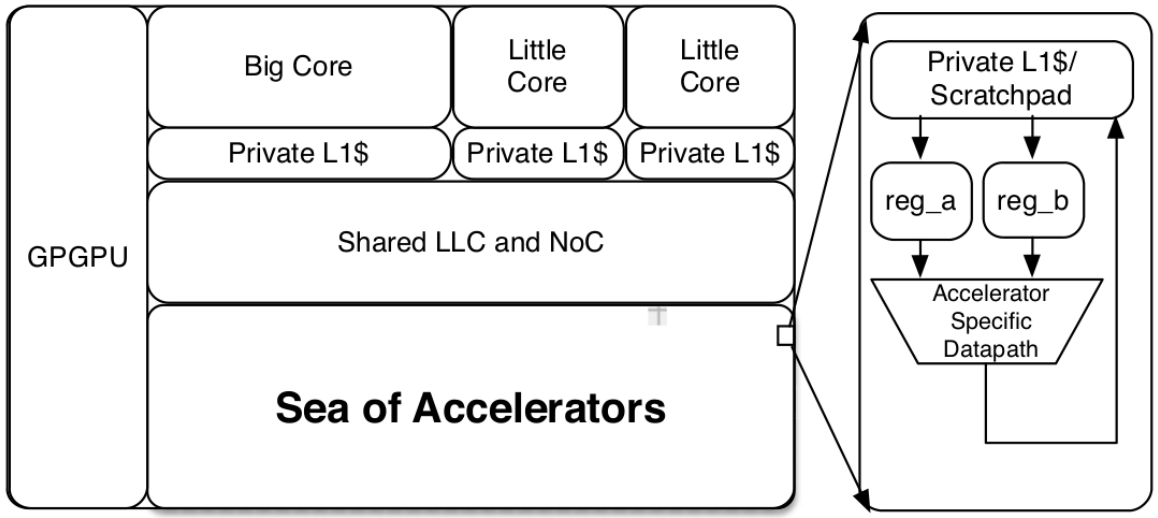
- Modern SoCs are increasingly heterogeneous
  - They integrate a growing number of accelerators

Accelerators in Apple SoCs



Graph from vlsiarch.eecs.harvard.edu/research/accelerators/die-photo-analysis

Future heterogeneous architecture



[Shao 2015]

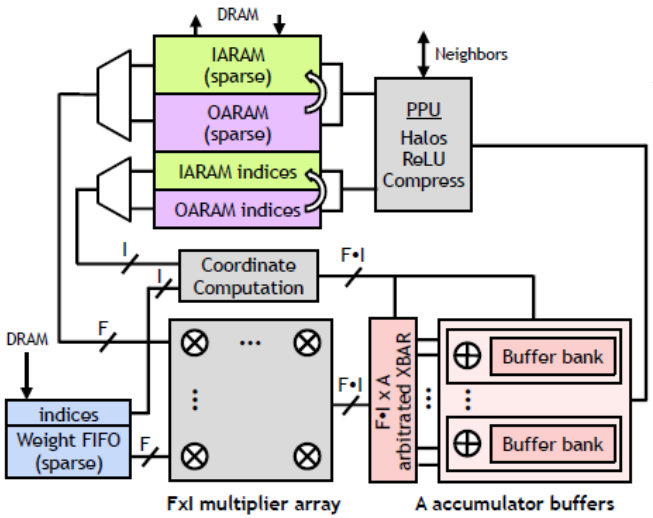
ACCELERATORS TAXONOMY: SPECIALIZATION

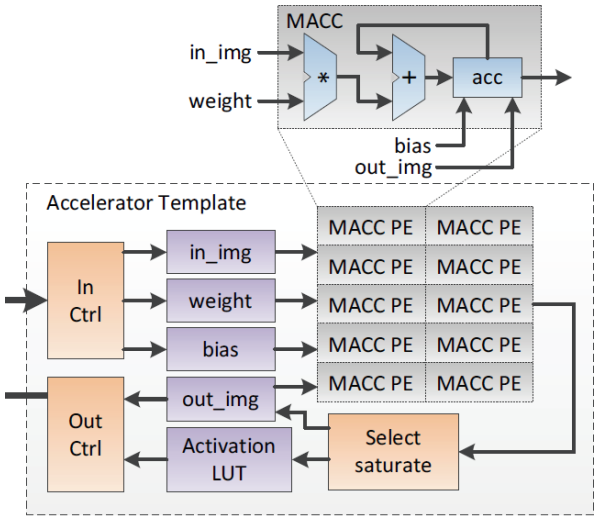
CPUs — **Programmable accelerators** — **Fixed-function accelerators**

low — specialization degree — high

GPUs

SCNN [Parashar 2017]
[Peemen 2013]





[Cascaval 2010] [Shao 2015]

ACCELERATORS TAXONOMY: SPECIALIZATION

CPUs

Programmable accelerators

Fixed-function accelerators

low — specialization degree — high

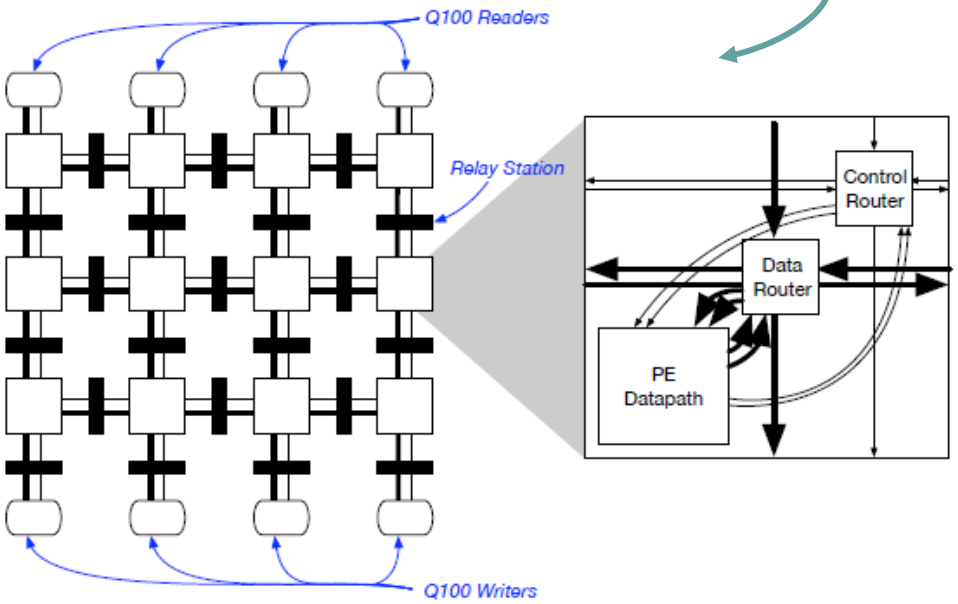
GPUs

[Lottarini 2019]

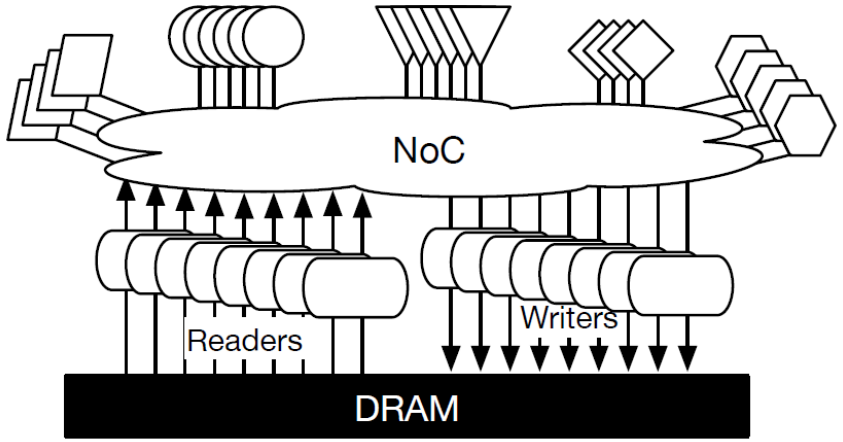
Q100 [Wu 2014]

SCNN [Parashar 2017]

[Peemen 2013]



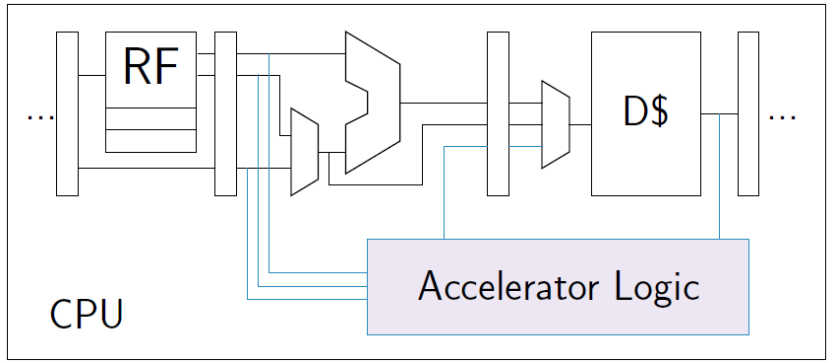
9 types of operator modules.
(Here, shape indicates function.)



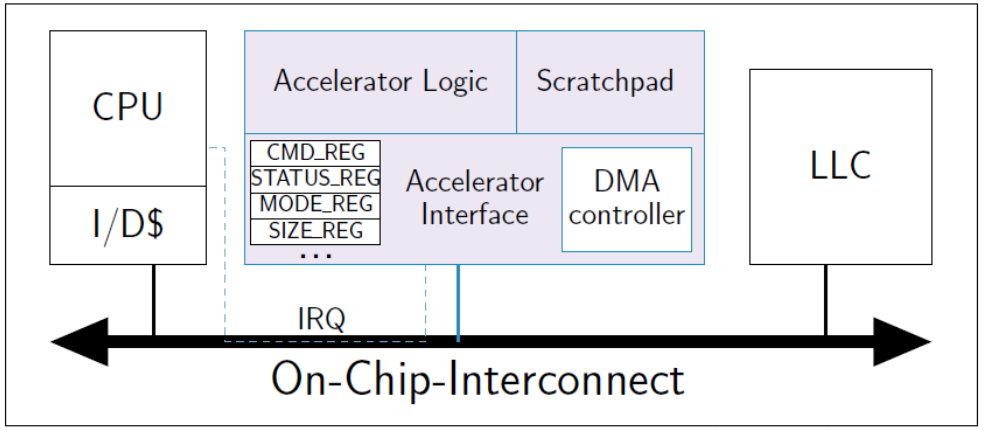
[Cascaval 2010] [Shao 2015]

ACCELERATORS TAXONOMY: COUPLING

- Tightly coupled
  - Part of the processor pipeline *or*
  - Attached to the private caches



- Loosely coupled
  - **Attached to the on-chip interconnect** or
  - Off-chip



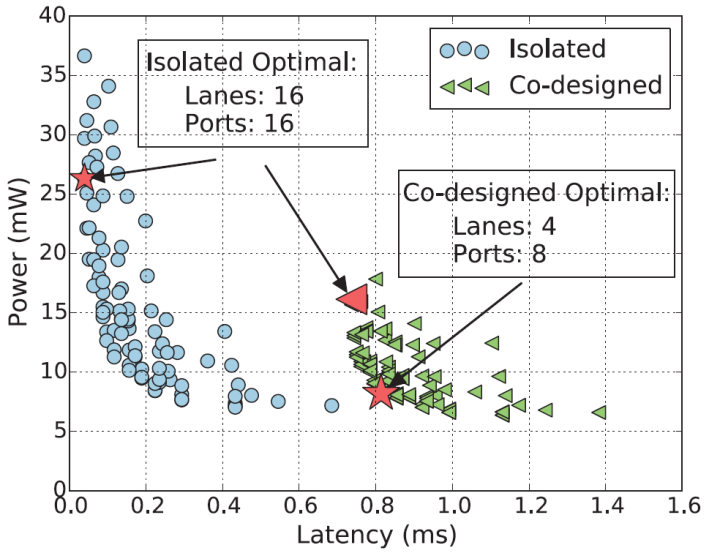
[Cascaval 2010] [Cong 2012] [Cota 2015] [Shao 2015]

ACCELERATOR DESIGN vs INTEGRATION

Most research focused on the accelerator design in isolation with little attention to its system integration

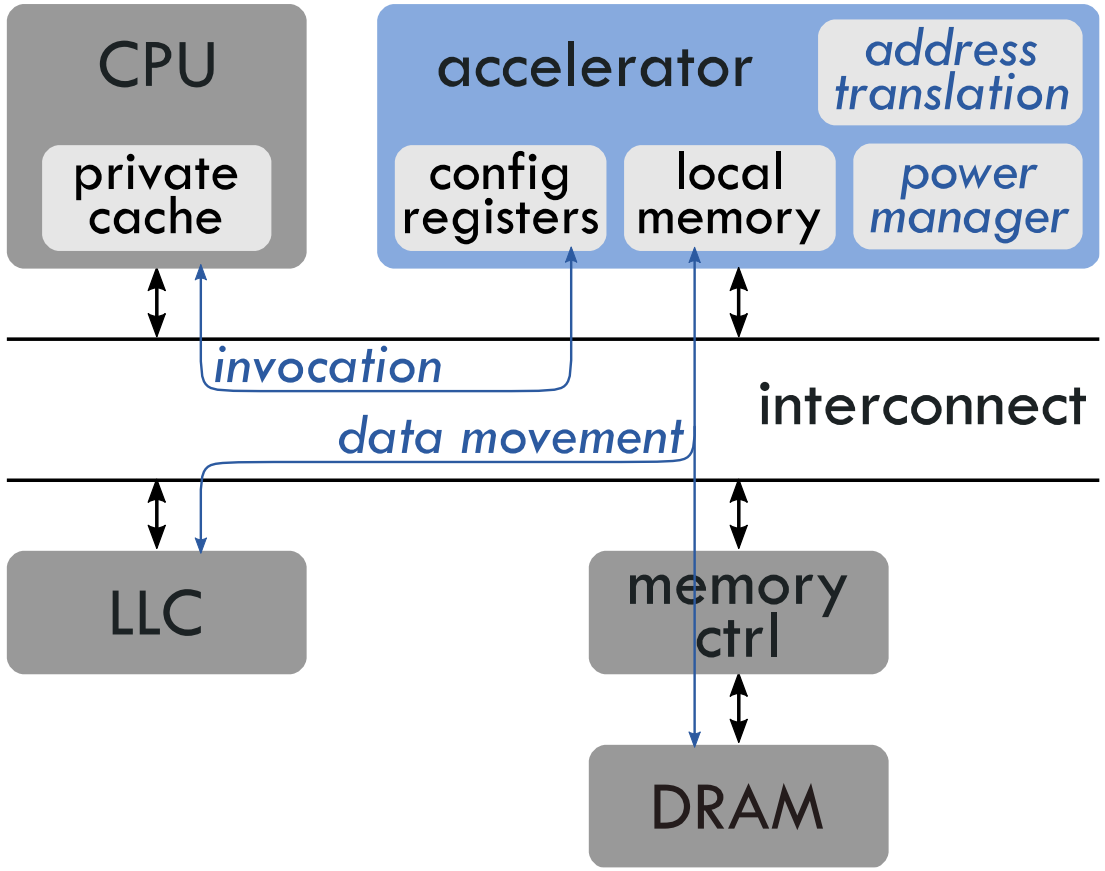
"Existing research on accelerators has focused on computational aspects and has disregarded design decisions with practical implications, such as the model for accelerator invocation from software and the interaction between accelerators and the components [...] surrounding them."
[Cota 2015]

"The co-design of the accelerator microarchitecture with the system in which it belongs is critical to balanced, efficient accelerator microarchitectures."
[Shao 2016]



ACCELERATOR INTEGRATION CHALLENGES

- Invocation
- Addressing
- Data movement
- Power management



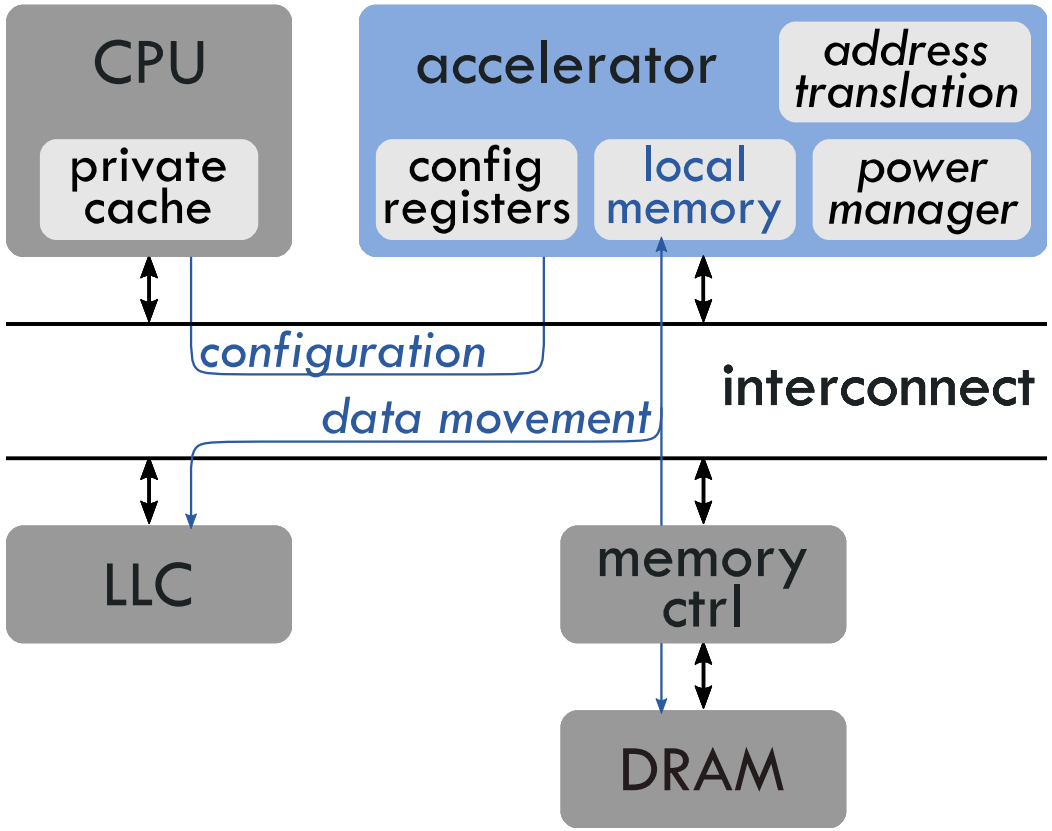
[Cascaval 2010] [Shao 2015] [Cota 2015]

INVOCATION

INVOCATION MODEL

Device driver approach

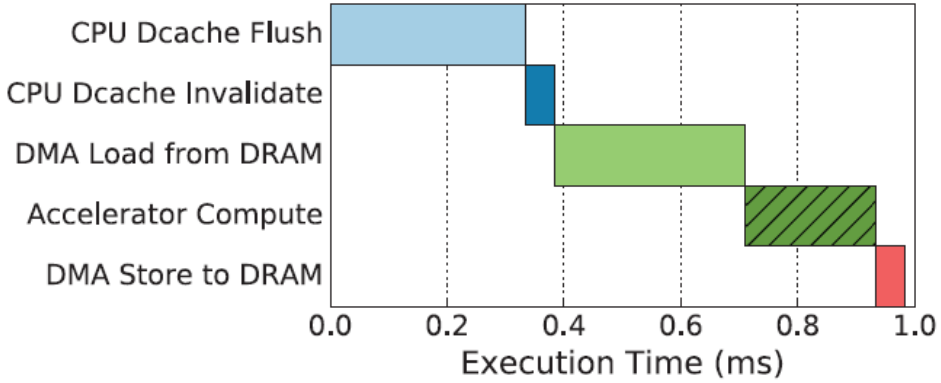
- A user app calls the device driver
- The device driver
  - (optional) flushes the caches
  - configures the accelerator
  - waits for the accelerator completion
  - returns control to the user app
- The programmer must guarantee race-free accelerator execution



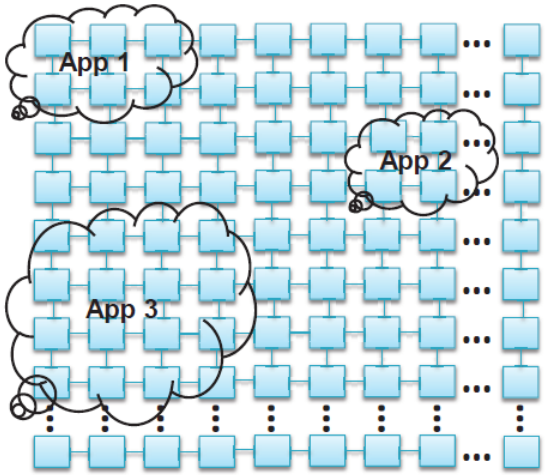
[Chen 2013] [Cota 2015] [Shao 2016] [Mantovani 2016A]

INVOCATION CHALLENGES

- Invocation overhead
  - Negligible if the task offloaded to the accelerator is coarse enough [Chen 2013] [Cota 2015] [Shao 2016] [Mantovani 2016A]



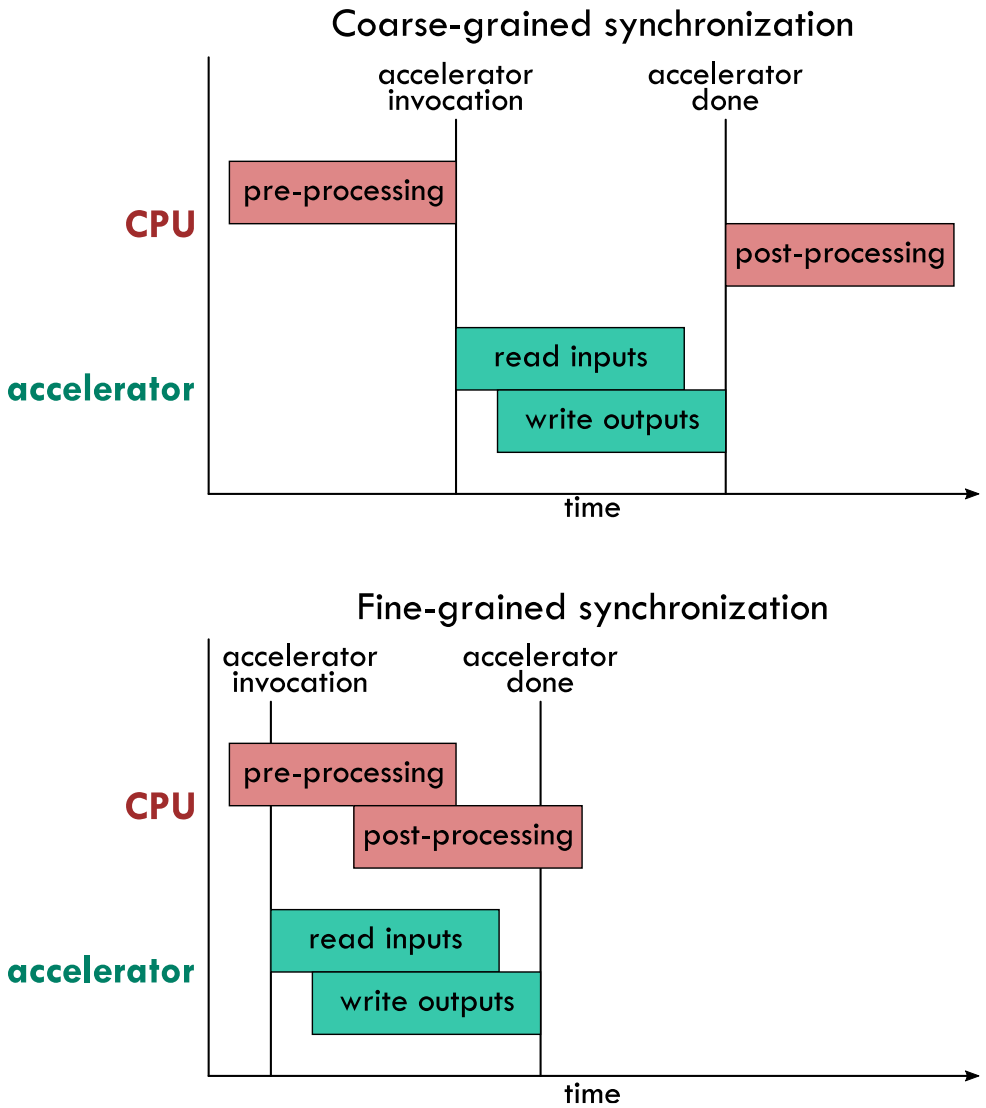
- Flushing of all caches is disruptive in large SoCs
  - Limit the flush to a few private caches and LLC partitions
    - **Coherence Domain Restriction** [Fu 2015]
      Limit the number of sharers and LLC partitions that partake in a coherence domain



INVOCATION CHALLENGES

- No fine-grain CPU-accelerator synchronization
  - All input data ready before accelerator invocation
  - Race-free accelerator access to inputs and outputs

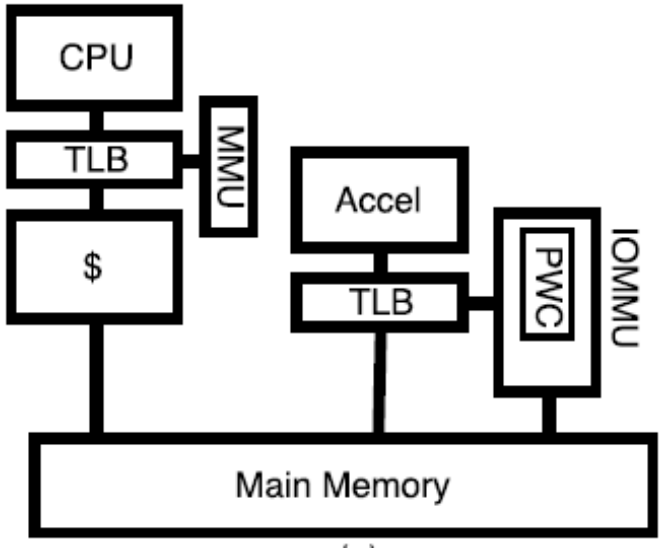
  - How to enable early accelerator launch and proactive data return with a fine-grained synchronization scheme?
    - **Full-empty bits scheme** [Lustig 2013]
      CPU and accelerator can inform each other on whether or not a region of input/output data is ready



ADDRESSING

CONVENTIONAL ADDRESSING MODELS

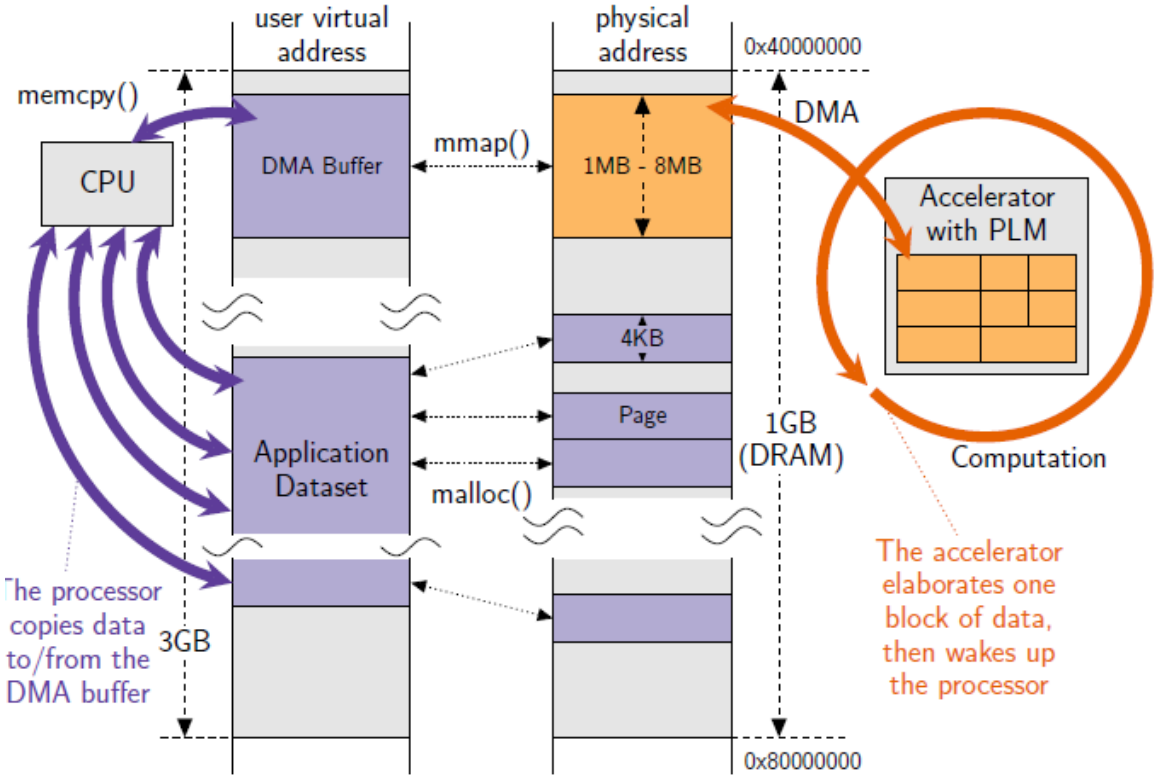
- **IOMMU**
  - High programmability
    - CPU and accelerator share virtual address space
  - Low performance
    - Latency of page table walks on the critical path
    - Area overhead



[Haria 2018]

CONVENTIONAL ADDRESSING MODELS

- **Contiguous physical memory**
  - Data is allocated in contiguous physical memory
  - Programmability
    - The accelerator works in physical address space
    - The contiguous buffer must be pinned in memory
    - Large contiguous memory may not be available
  - Performance
    - No translation needed
    - Normally requires data copies

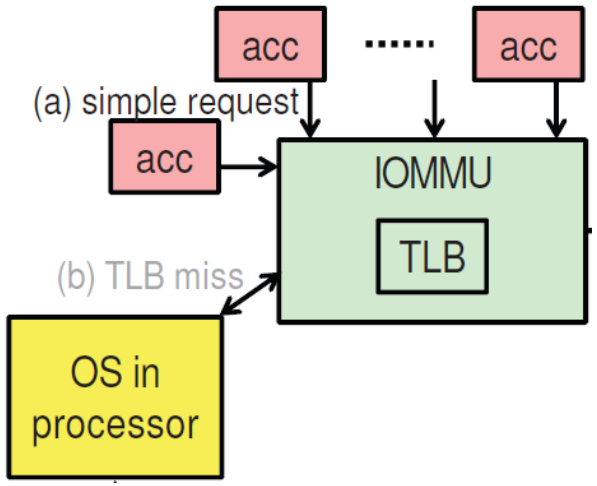


[Mantovani 2016A] [Kornaros 2018]

NOVEL ADDRESSING MODELS

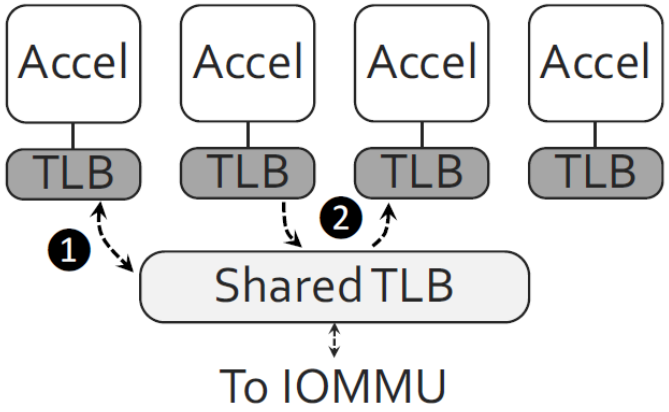
[Chen 2013] [Hao 2017]

- Leverage the CPU MMU to serve TLB misses
- Share one IOMMU and TLB among accelerators



[Hao 2017]

- Add a small per-accelerator private TLB



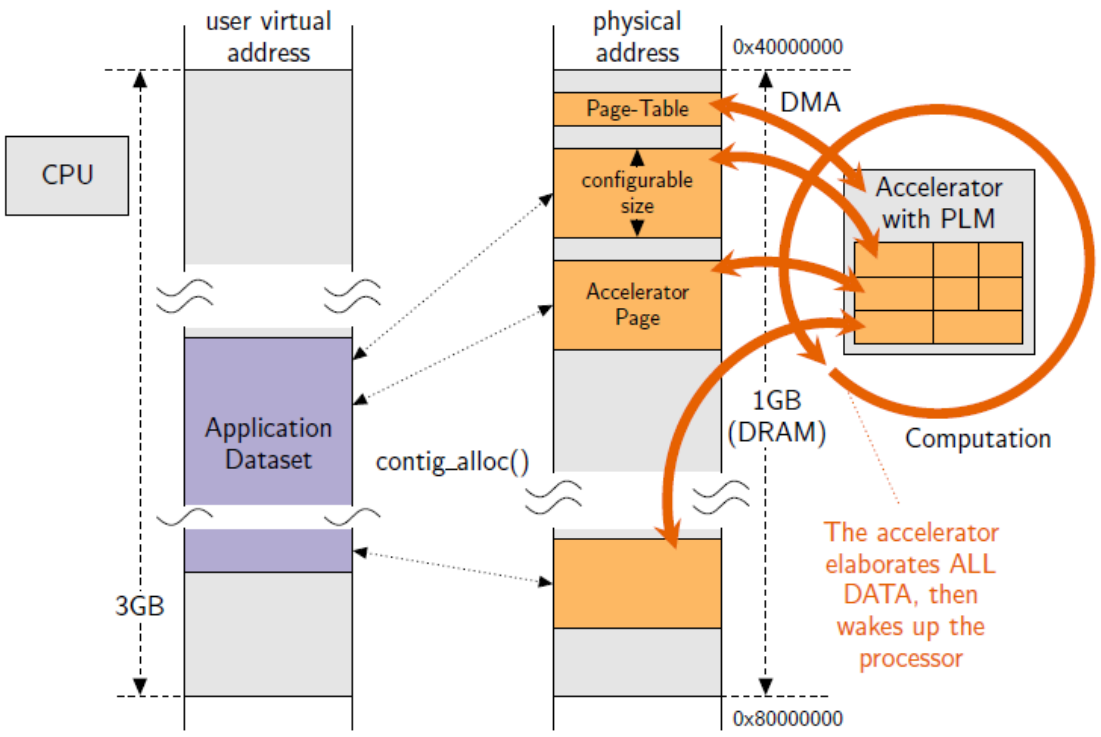
NOVEL ADDRESSING MODELS

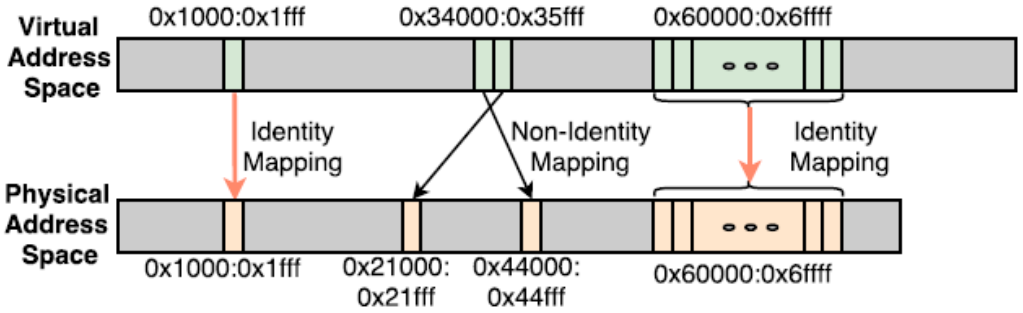
[Mantovani 2016A]

- Allocate memory in big pages and prepare a small page table of physical addresses that the accelerator will fetch and store locally
  - No contiguous buffer in physical memory
  - No TLB misses
  - No need for data copies

[Haria 2018]

- Allocate memory such that physical and virtual addresses are almost always identical
  - No translation needed in most cases





DATA MOVEMENT

DATA MOVEMENT CHALLENGES

- Accelerators have custom private scratchpads built to minimize the memory accesses [Peemen 2013] [Parashar 2017]
- From a system perspective an accelerator can be characterized by its memory access pattern [Lyons 2011] [Cota 2015]

System integration challenges

- Interaction with the **memory hierarchy**
- Solutions to increase the **scratchpads utilization** since they occupy most of the accelerator area

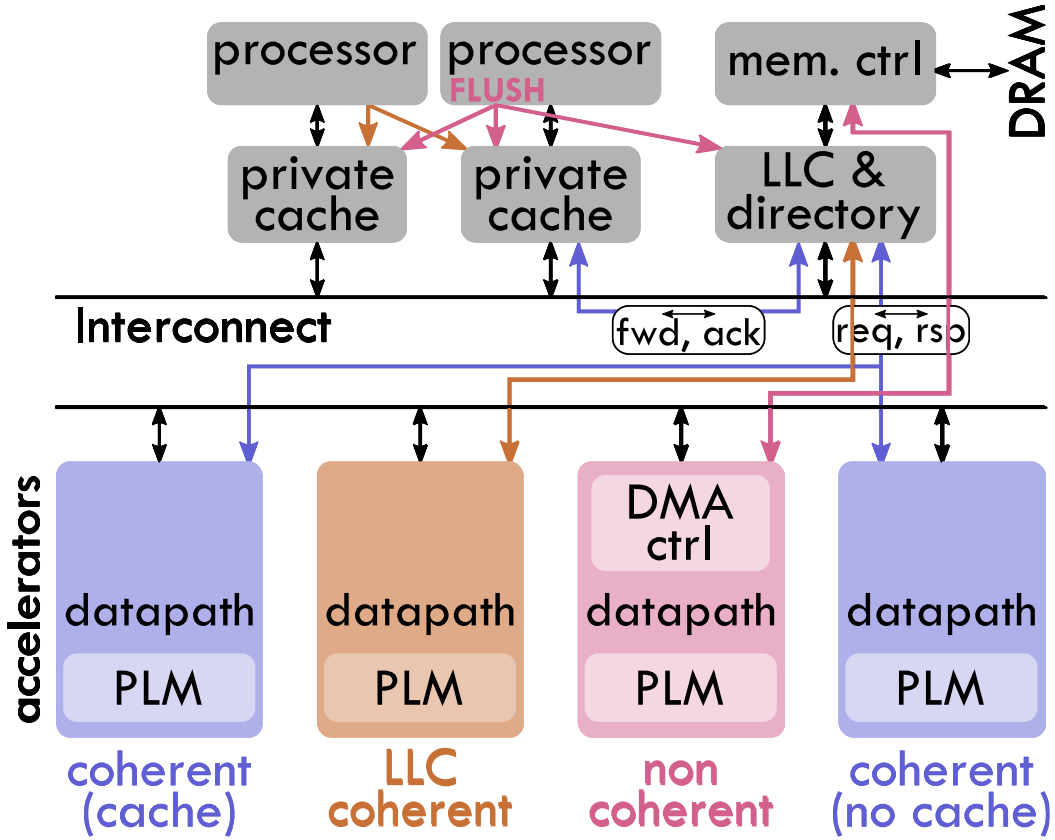
CACHE COHERENCE MODELS

Main models

- Non-coherent [Cong 2012] [Cota 2015] [Shao 2016]
- LLC-coherent [Cota 2015]
- Coherent [Lyons 2012] [Shao 2016]

Novel solutions

- **Cohesion** [Kelm 2011]
  - Hybrid hardware- and software-managed coherence
  - Fine-grained temporal and spatial reassignment between the two coherence models
  - Save cache coherence overheads when not needed

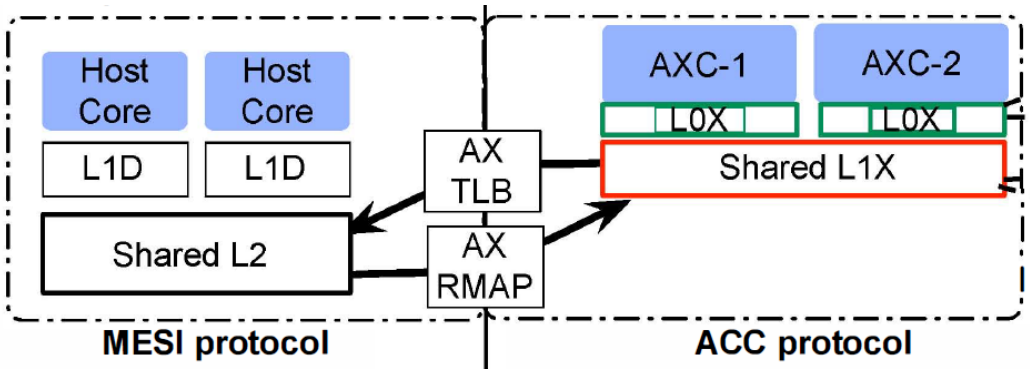


CACHE COHERENCE MODELS

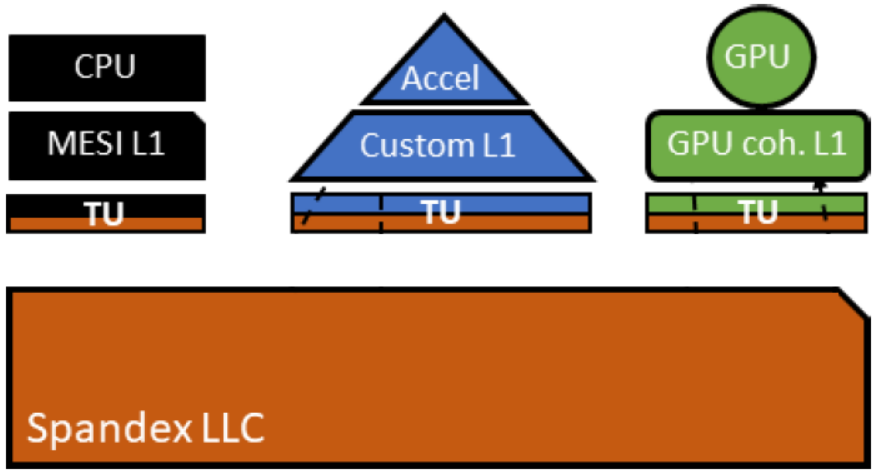
Novel solutions

- **Fusion** [Kumar 2015]
  - Two levels private cache hierarchy for accelerators
  - ACC: a lightweight timestamp-based coherence
- **Spandex** [Alsop 2018]
  - Flexible interface for heterogeneous coherence because different components require different coherence
  - Support for CPU coherence, GPU coherence and DeNovo coherence
  - LLC based on DeNovo coherence protocol

Fusion



Spandex



SCRATCHPAD OPPORTUNITIES

- Two size thresholds for the accelerator scratchpad
  - Minimum size to support the parallelism of the datapath
    - Must be tightly coupled with the datapath!
  - Minimum size to maximize reuse and minimize memory accesses
    - Must be on-chip

Many system-level solutions for this

Goal

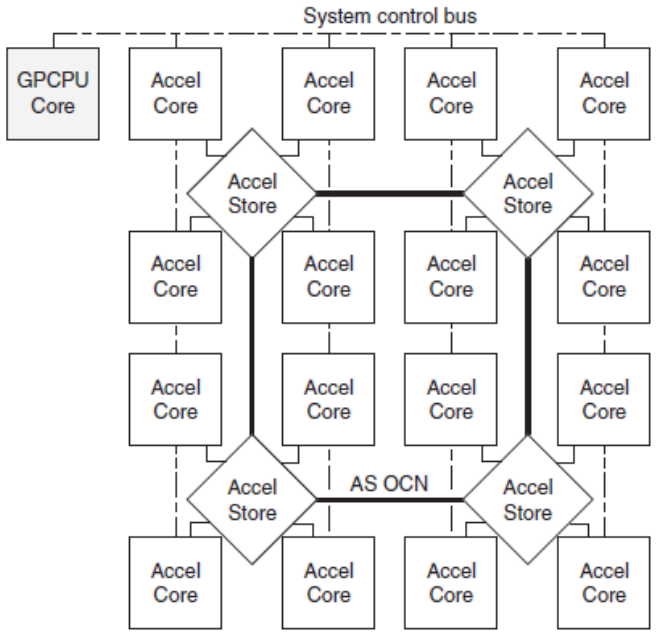
- Increase scratchpad utilization and reduce memory accesses

[Lyons 2012] [Pellauer 2019]

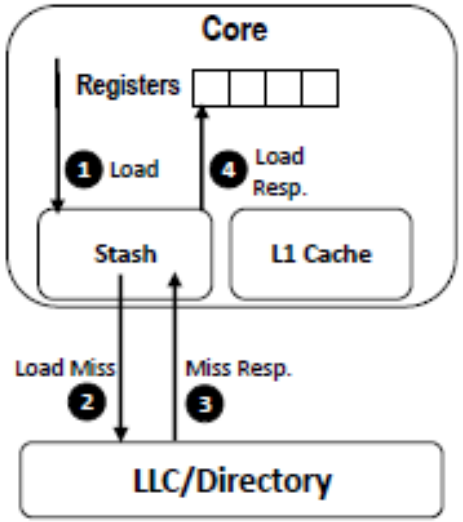
SCRATCHPAD OPTIMIZATIONS

- Accelerator chaining [Cong 2012]
- Shared scratchpads [Chen 2013]
  - **Accelerator store** [Lyons 2012]
  - **Buffets** [Pellauer 2019]
    - Shared hierarchy of intelligent scratchpads
  - **Buffer-integrated cache** [Fajardo 2011]
    - Scratchpad of reconfigurable size integrated in the LLC
    - Larger scratchpad means less LLC ways
- **Stash** [Komuravelly 2015]
  - Combines the benefits of caches and scratchpads
    - Like a scratchpad: explicit access and compact storage
    - Like a cache: globally addressable and visible (support implicit data movement)

Accelerator store



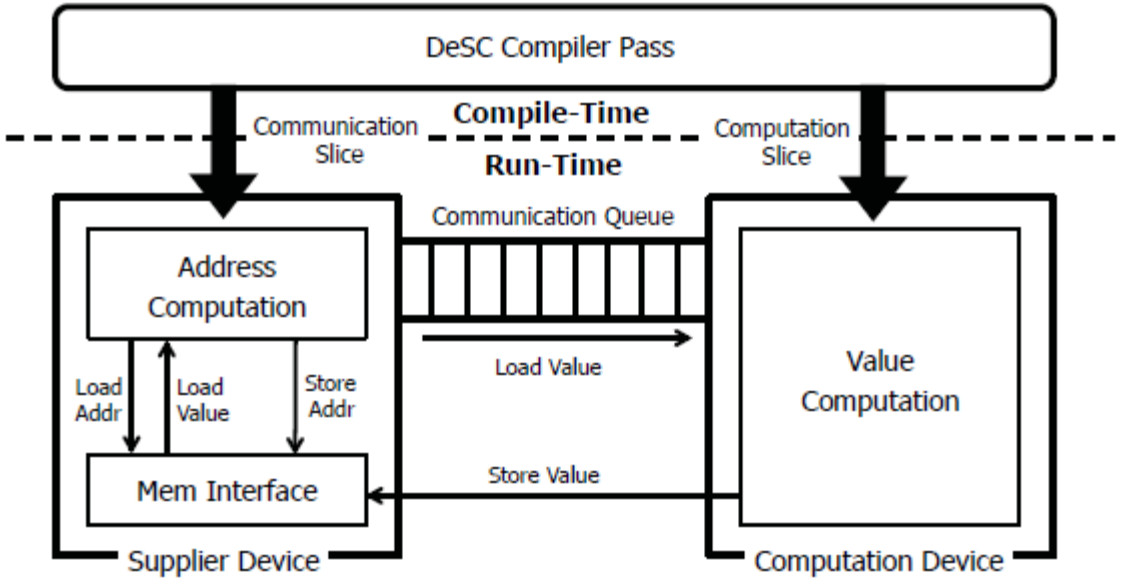
Stash



DECOUPLING OF SUPPLY AND COMPUTE

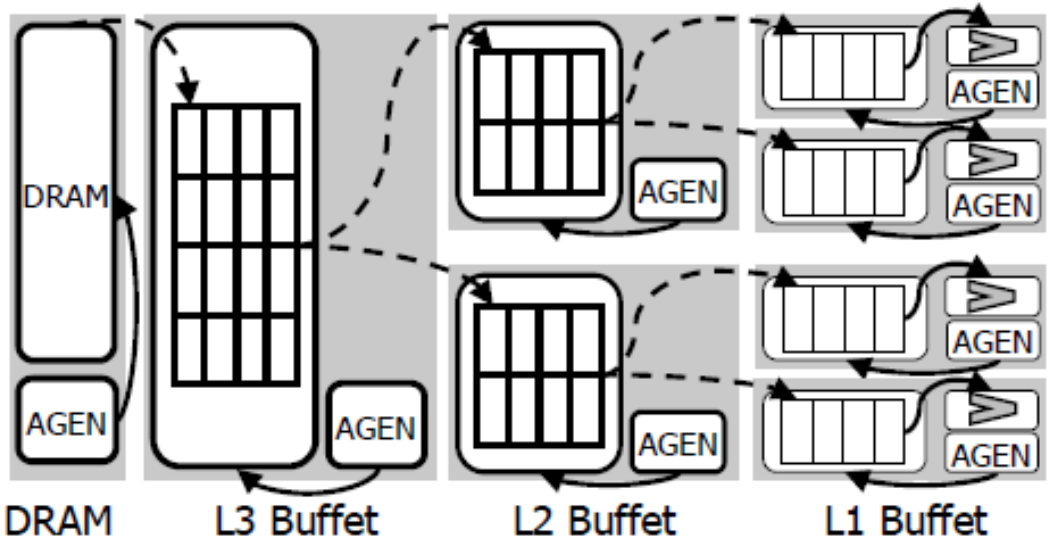
DeSC [Ham 2015]

- Decouple supply and compute parts of a program
- Offload to either CPUs or accelerators



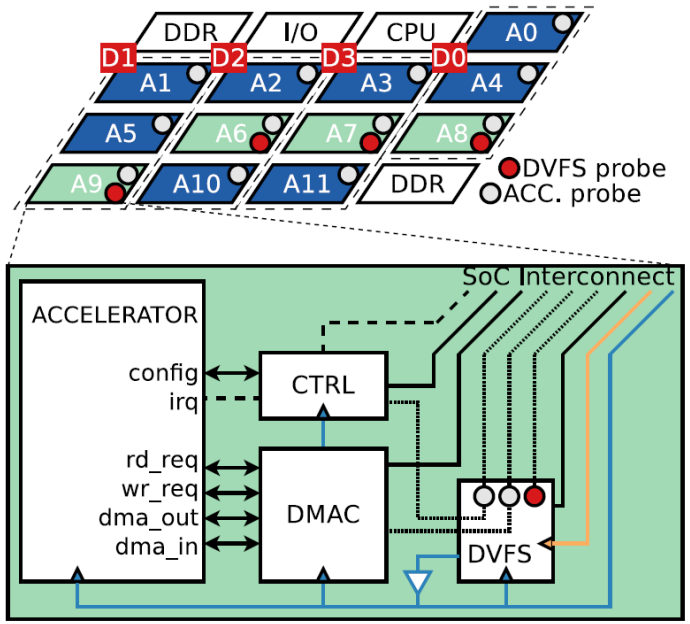
Buffets [Pellauer 2019]

- Hierarchy of intelligent scratchpads with load/store capabilities
- Efficient multi-casting
- Fine-grained supply-compute synchronization



POWER

POWER MANAGEMENT



Limited power budgets and growing number of on-chip IPs

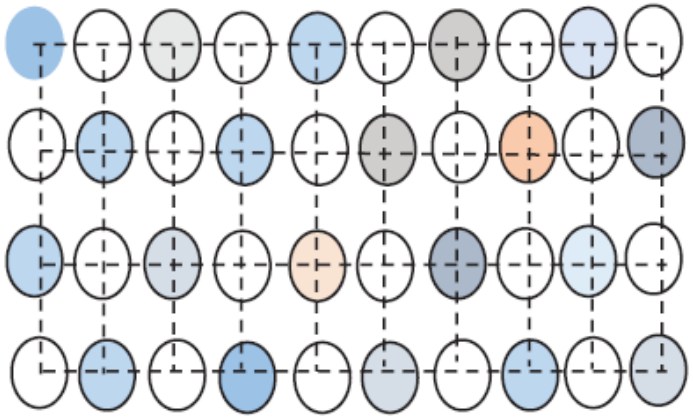
- There is an energy saving opportunity at accelerator granularity

[Mantovani 2016B]

- Fine-grained dynamic voltage-frequency scaling on accelerators
- Each VF domain has a voltage regulator and a PLL
- Various DVFS policies
  - fixed VF
  - tuning based on NoC congestion and communication-computation ratio
  - limit overall on chip power budget

[Vega 2017]

- The maximum power budget is represented by an amount of tokens
- Decentralized management
  - cores exchange tokens with neighbors according to their power needs



CONCLUSION

CONCLUSION

- The accelerator integration choices affect the accelerator performance
  - Invocation, addressing, data movement and power management.
- Most integration solutions are completely decoupled from the accelerator design
  - The system around the accelerator should take care of all the integration aspects
  - The accelerator designer should not worry about integration aspects, but it should take them into account to realistically evaluate the accelerator

THANK YOU!

SYLLABUS

Syllabus organized by topic with paper links

[Alsop 2018] J. Alsop, M. Sinclair, S. Adve, "**Spandex: A Flexible Interface for Efficient Heterogeneous Coherence,**" *International Symposium on Computer Architecture (ISCA)*, 2018.

[Cascaval 2010] C. Cascaval, S. Chatterjee, H. Franke, K. J. Gildea, P. Pattnaik, "**A Taxonomy of Accelerator Architectures and their Programming Models,**" *IBM Journal of Research and Development*, 2010.

[Chen 2013] Y. Chen, J. Cong, M. A. Ghodrat, M. Huang, C. Liu, B. Xiao, Y. Zou, "**Accelerator-rich CMPs: From Concept to Real Hardware,**" *International Conference on Computer Design (ICCD)*, 2013.

[Cong 2012] J. Cong, M. A. Ghodrat, M. Gill, B. Grigorian, G. Reinman, "**Architecture Support for Accelerator-Rich CMPs,**" *Design Automation Conference (DAC)*, 2012.

[Cota 2015] E. G. Cota, P. Mantovani, G. Di Guglielmo, L. P. Carloni, "**An Analysis of Accelerator Coupling in Heterogeneous Architectures,**" *Design Automation Conference (DAC)*, 2015.

[Fajardo 2011] C. F. Fajardo, Z. Fang, R. Iyer, G. F. Garcia, S. E. Lee L. Zhao, "**Buffer-Integrated-Cache: A Cost-effective SRAM Architecture for Handheld and Embedded Platforms,**" *Design Automation Conference (DAC)*, 2011.

[Fu 2015] Y. Fu, T. M. Nguyen, D. Wentzlaff, "**Coherence Domain Restriction on Large Scale Systems,**" *International Symposium on Microarchitecture (MICRO)*, 2015.

[Ham 2015] T. J. Ham, J. L. Aragón, M. Martonosi, "**DeSC: Decoupled Supply-compute Communication Management for Heterogeneous Architectures,**" *International Symposium on Microarchitecture (MICRO)*, 2015.

[Hao 2017] Y. Hao, Z. Fang, G. Reinman, J. Cong, "**Supporting Address Translation for Accelerator-Centric Architectures,**" *International Symposium on Computer Architecture (ISCA)*, 2017.

[Haria 2018] S. Haria, M. D. Hill, M. M. Swift, "**Devirtualizing Memory in Heterogeneous Systems,**" *Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2018.

[Kelm 2011] J. H. Kelm, D. R. Johnson, W. Tuohy, S. S. Lumetta, S. J. Patel, "**Cohesion: An Adaptive Hybrid Memory Model for Accelerators**," *IEEE Micro*, 2011.

[Komuravelli 2015] R. Komuravelli, M. D. Sinclair, J. Alsop, M. Huzaifa, M. Kotsifakou, P. Srivastava, S. V. Adve, V. S. Adve, "**Stash: Have Your Scratchpad and Cache It Too**," *International Symposium on Computer Architecture (ISCA)*, 2015.

[Kornaros 2018] G. Kornaros, M. Coppola, "**Enabling Efficient Job Dispatching in Accelerator-extended Heterogeneous Systems with Unified Address Space**," *International Symposium on Computer Architecture (ISCA)*, 2018.

[Kumar 2015] S. Kumar, A. Shriraman, N. Vedula, "**Fusion: Design Tradeoffs in Coherent Cache Hierarchies for Accelerators**," *International Symposium on Computer Architecture (ISCA)*, 2015.

[Lottarini 2019] A. Lottarini, J. P. Cerqueira, T. J. Repetti, S. A. Edwards, K. A. Ross, M. Seok, M. A. Kim, "**Master of None Acceleration: a Comparison of Accelerator Architectures for Analytical Query Processing**," *International Symposium on Computer Architecture (ISCA)*, 2019.

[Lustig 2013] D. Lustig and M. Martonosi, "**Reducing GPU Offload Latency Via Fine-grained CPU-GPU Synchronization**," *International Symposium on High-Performance Computer Architecture (HPCA)*, 2013.

[Lyons 2012] M. Lyons, M. Hempstead, G. Wei, D. Brooks, "**The Accelerator Store: A Shared Memory Framework for Accelerator-based Systems**," *ACM Transactions on Architecture and Code Optimization (TACO)*, 2012.

[Mantovani 2016A] P. Mantovani, E. G. Cota, C. Pilato, G. Di Guglielmo, L. P. Carloni, "**Handling Large Data Sets for High-performance Embedded Applications in Heterogeneous Systems-on-Chip**," *International Conference on Compilers, Architectures, and Synthesis of Embedded Systems (CASES)*, 2016.

[Mantovani 2016B] Paolo Mantovani, Emilio G. Cota, Kevin Tien, Christian Pilato, Giuseppe Di Guglielmo, Ken Shepard, and Luca P. Carloni, "**An FPGA-based Infrastructure for Fine-grained DVFS Analysis in High-performance Embedded Systems**," *Design Automation Conference (DAC)*, 2016.

[Parashar 2017] A. Parashar, M. Rhu, A. Mukkara, A. Puglielli, R. Venkatesan, B. Khailany, J. Emer, S. W. Keckler, W. J. Dally, "**SCNN: An Accelerator for Compressed-sparse Convolutional Neural Networks,**" *International Symposium on Computer Architecture (ISCA)*, 2017.

[Peemen 2013] M. Peemen, A. A. A. Setio, B. Mesman, H. Corporaal, "**Memory-centric Accelerator Design for Convolutional Neural Networks,**" *International Conference on Computer Design (ICCD)*, 2013.

[Pellauer 2019] M. Pellauer, Y. S. Shao, J. Clemons, N. Crago, K. Hegde, R. Venkatesan, S. W. Keckler, C. W. Fletcher, J. Emer, "**Buffets: An Efficient and Composable Storage Idiom for Explicit Decoupled Data Orchestration**," *Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2019.

[Shao 2015] Y. S. Shao, D. Brooks, "**Research Infrastructures for Hardware Accelerators,**" *Synthesis Lectures on Computer Architecture, Morgan & Claypool*, 2015, chapters 1-2.

[Shao 2016] Y. S. Shao, S. L. Xi, V. Srinivasan, G. Wei, D. Brooks, "**Co-designing Accelerators and SoC Interfaces using Gem5-Aladdin,**" *International Symposium on Microarchitecture (MICRO)*, 2016.

[Wu 2014] L. Wu, A. Lottarini, T. K. Paine, M. A. Kim, K. A. Ross, "**Q100: The Architecture and Design of a Database Processing Unit,**" *Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2014.

[Vega 2017] A. Vega, A. Buyuktosunoglu, P. Bose, "**Invited paper: Secure swarm intelligence: A new approach to many-core power management,**" *International Symposium on Low Power Electronics and Design (ISLPED)*, 2017.