National Science Foundation
WHERE DISCOVERIES BEGIN

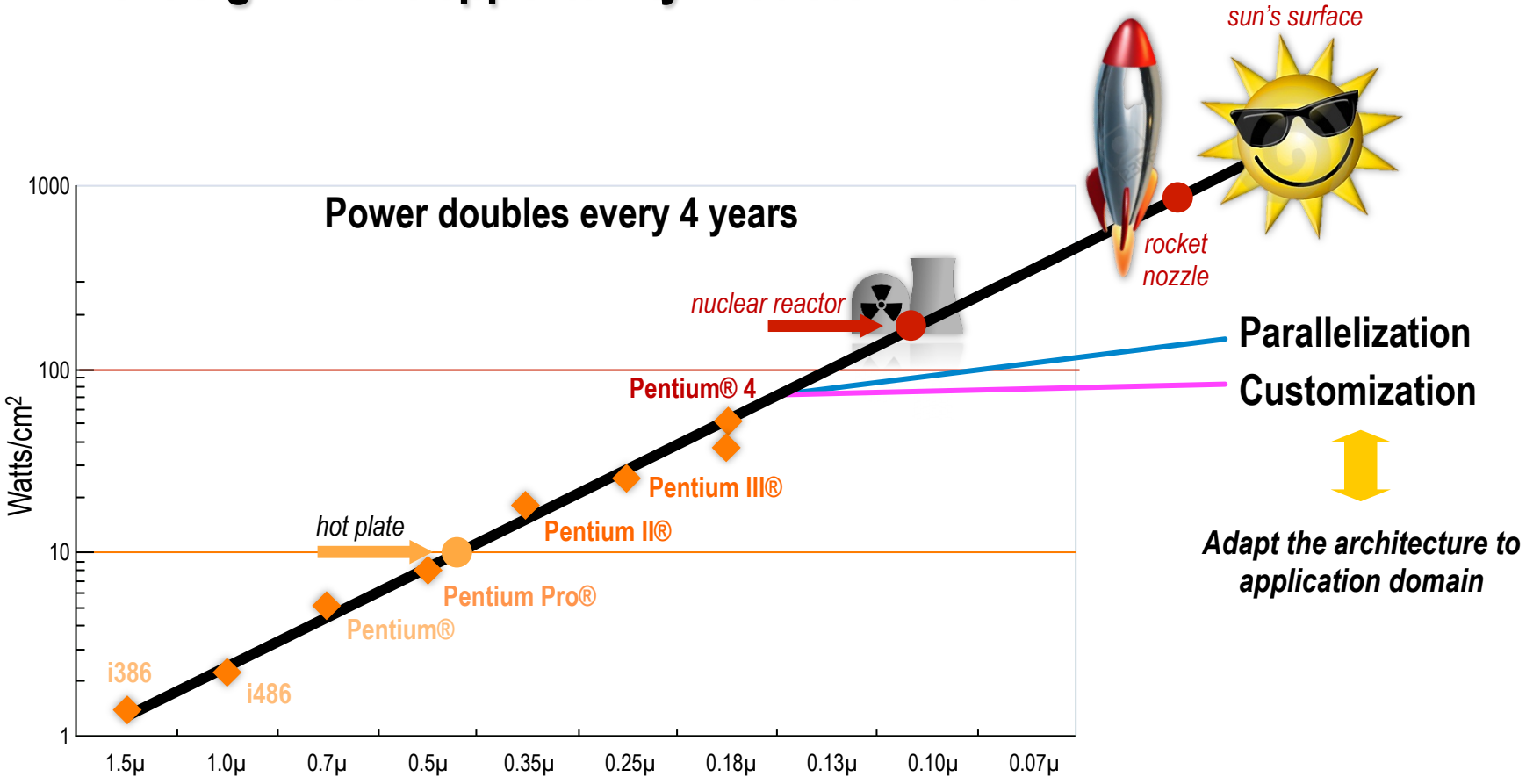
Customizable Domain-Specific Computing

Supported by NSF "Expedition in Computing" Program

Jason Cong, Director
Center for Domain-Specific Computing (CDSC)
www.cdsc.ucla.edu
cong@cs.ucla.edu

Focus: New Transformative Approach to Energy-efficient Computing

- Current solution: *Parallelization*
- Next significant opportunity – *Customization*



Based on Fred Pollack (Intel) and Michael Taylor (UCSD)

Justification: Potential of Customization

AES 128bit key 128bit data	Throughput	Power	Figure of Merit (Gb/s/W)
0.18mm CMOS	➢3.84 Gbits/sec	350 mW	11 (1/1)
FPGA [1]	1.32 Gbit/sec	490 mW	2.7 (1/4)
ASM StrongARM [2]	31 Mbit/sec	240 mW	0.13 (1/85)
ASM Pentium III [3]	648 Mbits/sec	41.4 W	0.015 (1/800)
C Emb. Sparc [4]	133 Kbits/sec	120 mW	0.0011 (1/10,000)
Java [5] Emb. Sparc	450 bits/sec	120 mW	0.0000037 (1/3,000,000)

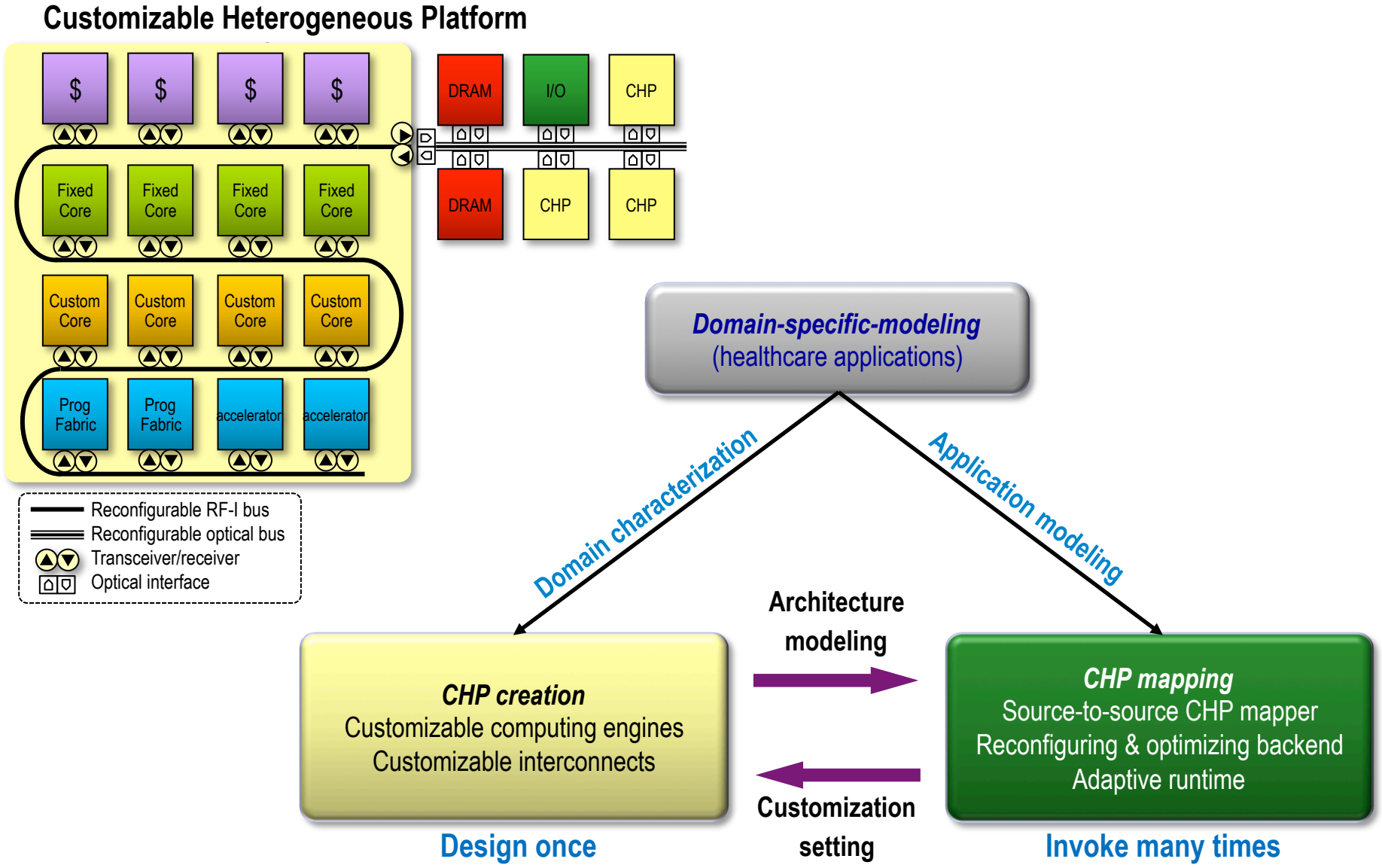
[1] Amphion CS5230 on Virtex2 + Xilinx Virtex2 Power Estimator
[2] Dag Arne Osvik: 544 cycles AES – ECB on StrongArm SA-1110
[3] Helger Lipmaa PIII assembly handcoded + Intel Pentium III (1.13 GHz) Datasheet
[4] gcc, 1 mW/MHz @ 120 Mhz Sparc – assumes 0.25 u CMOS
[5] Java on KVM (Sun J2ME, non-JIT) on 1 mW/MHz @ 120 MHz Sparc – assumes 0.25 u CMOS

Source: *P Schaumont and I Verbauwhede, "Domain specific codesign for embedded security," IEEE Computer 36(4), 2003*

Project Goals

- **A general, customizable platform for the given domain(s)**
  - Can be customized to a wide-range of applications in the domain
  - Can be massively produced with cost efficiency
  - Can be programmed efficiently with novel compilation and runtime systems
- **Metric of success**
  - A “supercomputer-in-a-box” with +100x performance/power improvement via customization for the intended domain(s)

Overview of CDSC Research Program



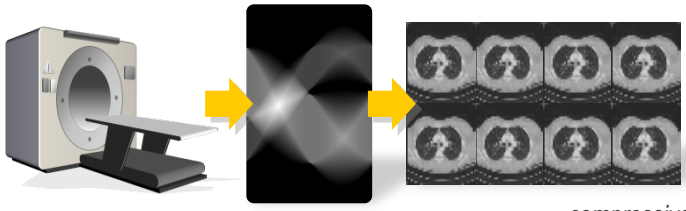
Application Domain

- **Medical imaging has changed the nature of healthcare and biomedical research**
  - Only *in vivo* method for understanding, diagnosis, and assessing treatment response for many diseases (e.g., cancer)
    - Computed tomography (CT)
    - Magnetic resonance (MR) imaging
  - Many state-of-the-art advances in medical imaging are hindered by computational runtime

Computational Challenges

medical image processing pipeline

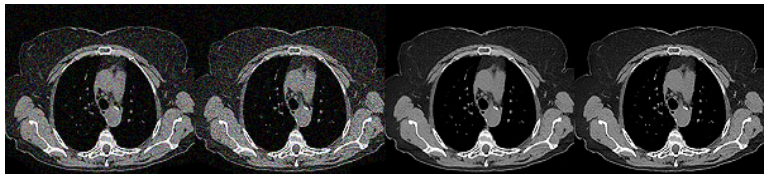
reconstruction



compressive sensing

Iterative convex optimization problem, dense and sparse matrix algebra, local and global operations

denoising



removal of noise/motion artifacts

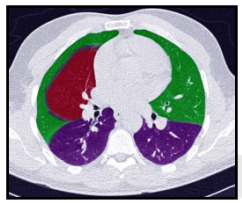
One-step explicit solution, requires non-local communication, non-iterative, grid patterns

registration

spatial remapping for comparison

Linear and nonlinear regularization steps, parallel and global communication patterns

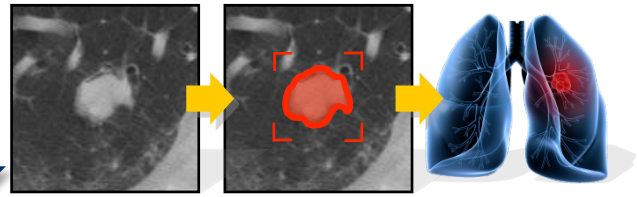
segmentation



identifying spatial regions of interest

Involves solving nonlinear PDEs to detect boundaries, dense linear algebra, local communication

analysis



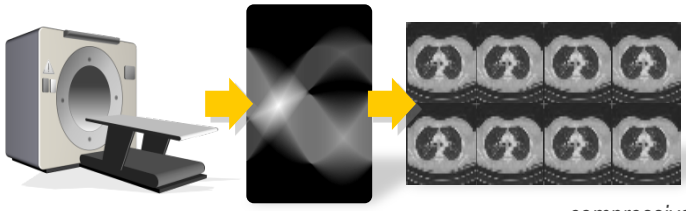
automatic quantification and classification

Machine leaning methods for classification, graphical models, local communication

Computational Challenges

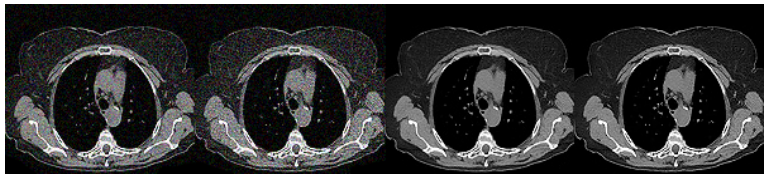
medical image processing pipeline

reconstruction



compressive sensing

denoising

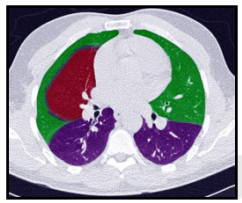


removal of noise/motion artifacts

registration

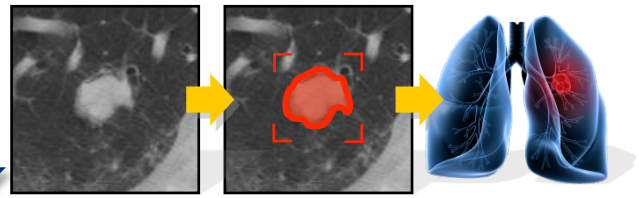
spatial remapping for comparison

segmentation



identifying spatial regions of interest

analysis

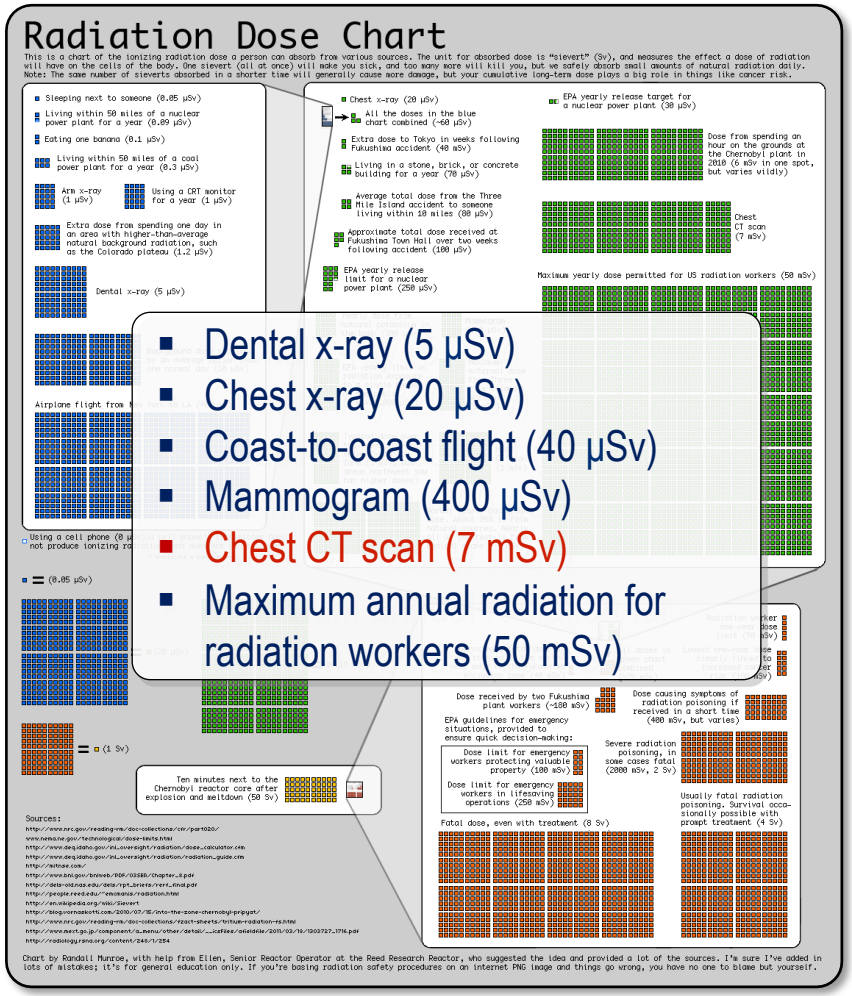


automatic quantification and classification

	Baseline	Target	Current implementation
reconstruction	1.5 min (Regular CT) 18 hours (Low-dose CT)	2 minutes	20 min (Low-dose CT)
denoising	5 minutes	1 minute	15 seconds
registration	10 minutes		2 minutes
segmentation	20 minutes		4 minutes
analysis	30 minutes		4 minutes
	~19 hours	*~3 mins.*	*~54 mins.*

What We Enable – Significant Radiation Reduction in CT

- **Medical image processing pipeline for lung cancer screening**
  - CT lung screening has been shown recently to reduce mortality via early detection
  - But there has been increased scrutiny of the use of medical imaging, cumulative lifetime radiation

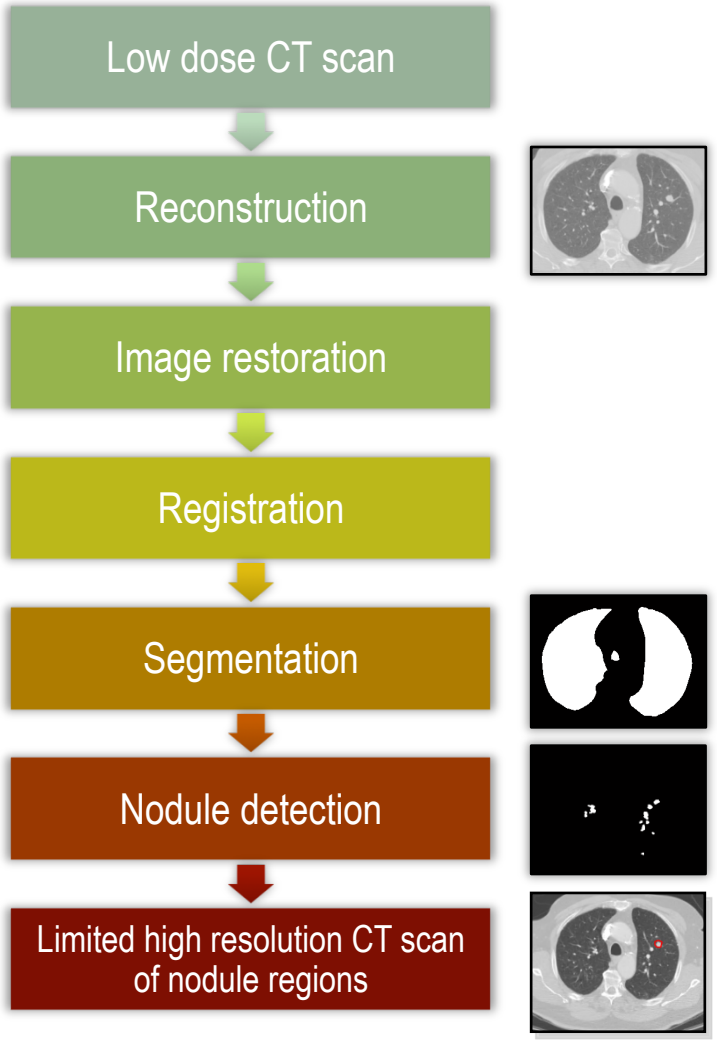


What We Enable – Significant Radiation Reduction in CT

- **Medical image processing pipeline for lung cancer screening**
  - CT lung screening has been shown recently to reduce mortality via early detection
  - But there has been increased scrutiny of the use of medical imaging, cumulative lifetime radiation

What We Enable – Significant Radiation Reduction in CT

- **Medical image processing pipeline for lung cancer screening**
  - CT lung screening has been shown recently to reduce mortality via early detection
  - But there has been increased scrutiny of the use of medical imaging, cumulative lifetime radiation
- **Implemented compressive sensing reconstruction with computer-aided diagnosis (CAD)**
  1. Low-dose CT scan is first performed
  2. EM+TV is used for reconstruction
  3. Resulting images are fed to the processing pipeline for registration, segmentation, and classification
  4. Automated detection of nodules > 2 mm
  5. Images containing these pulmonary nodules identify those regions that require subsequent higher-resolution scans and reconstruction



Current state: < 30 mins.

Customizable Heterogeneous Platform (CHP) Creation

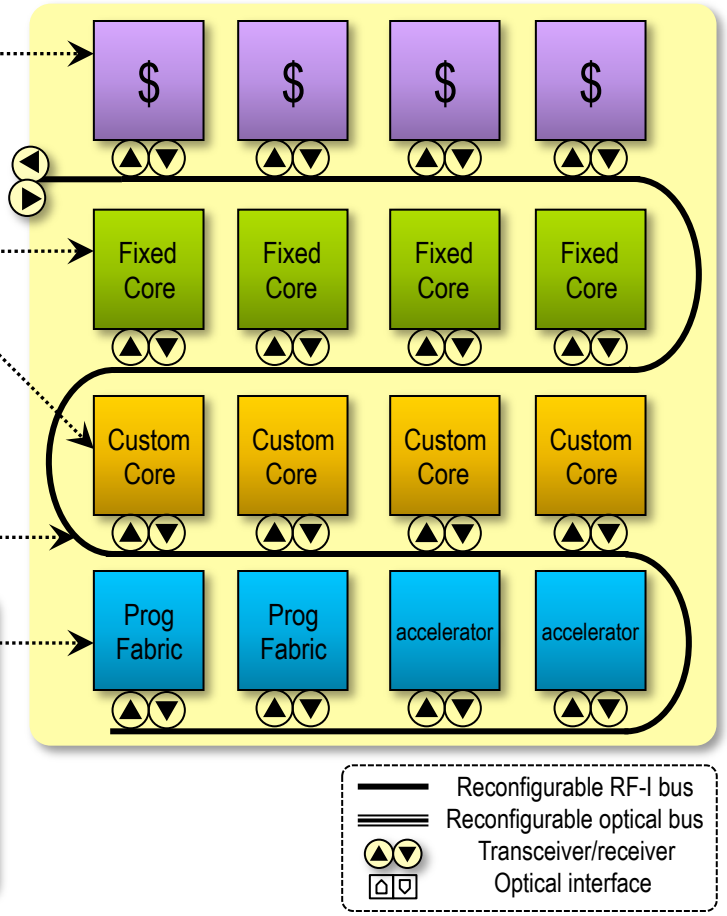
Cache parameters
- Cache size & configuration
- Cache vs. SPM
- …

Core parameters
- Frequency & voltage
- Datapath bit width
- Instruction window size
- Issue width
- Cache size & configuration
- Register file organization
- # of thread contexts
- …

NoC parameters
- Interconnect topology
- # of virtual channels
- Routing policy
- Link bandwidth
- Router pipeline depth
- Number of RF-I enabled routers
- RF-I channel and bandwidth allocation
- …

Custom instructions & accelerators
- Shared vs. private accelerators
- Choice of accelerators
- Custom instruction selection
- Amount of programmable fabric
- …

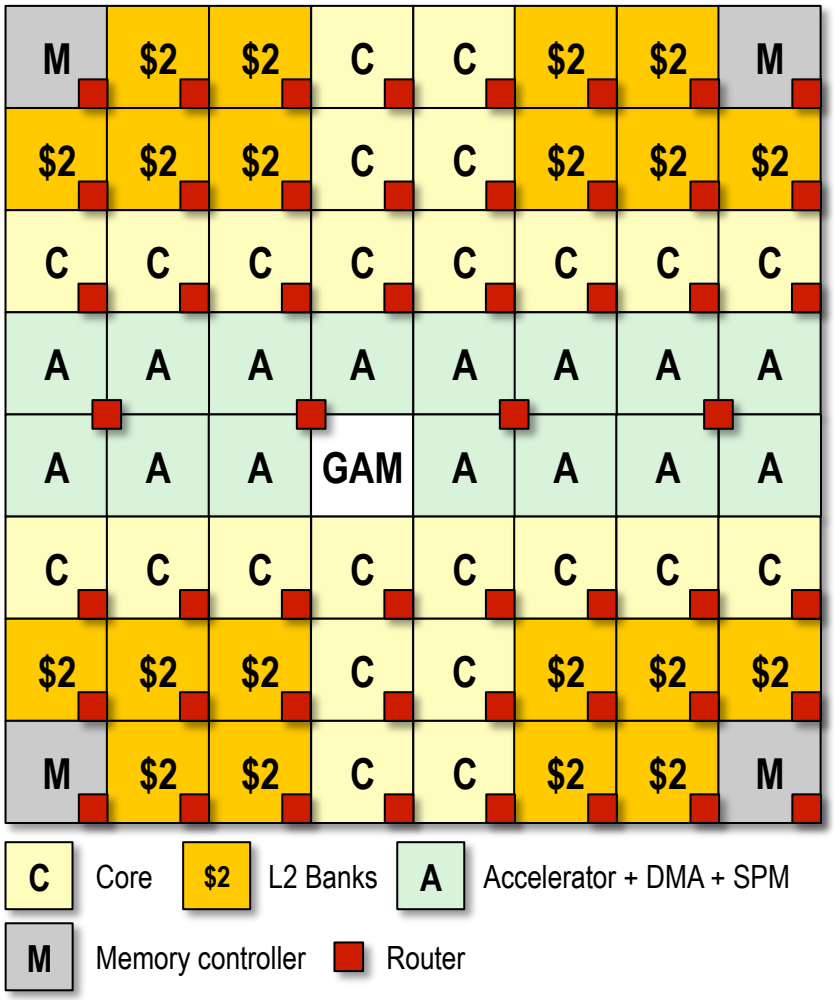
Customizable Heterogeneous Platform (CHP)



Key questions: Optimal trade-off between efficiency & customizability
Which options to fix at CHP creation? Which to be set by CHP mapper?

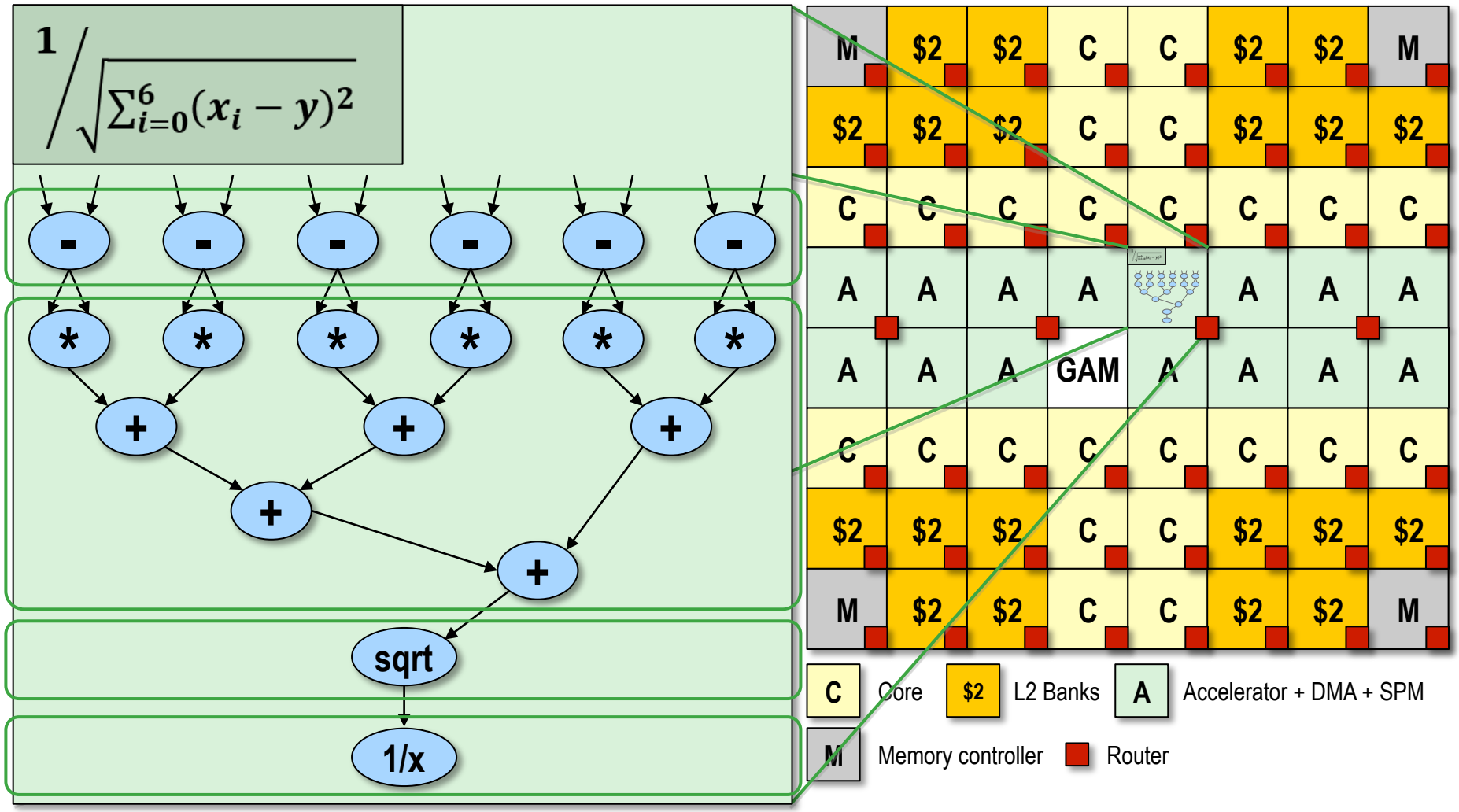
Highlight: Accelerator-Rich Architectures

- **Sea of Accelerators**
  - Accelerators deliver 100X+ performance and energy efficiency
- **Challenges**
  - Accelerators are inflexible
    - Limited use for new algorithms/domains
    - Often under-utilized
    - Many replicated structures
      - FP-ALUs, DMA engines, SPM
      - Unused when accelerator is idle
  - Need to support accelerator sharing, scheduling, management, virtualization
- **Our solutions**
  - On-chip Global Accelerator Manager (GAM)
  - Dynamic accelerator composition
  - Efficient memory and on-chip network support for accelerators

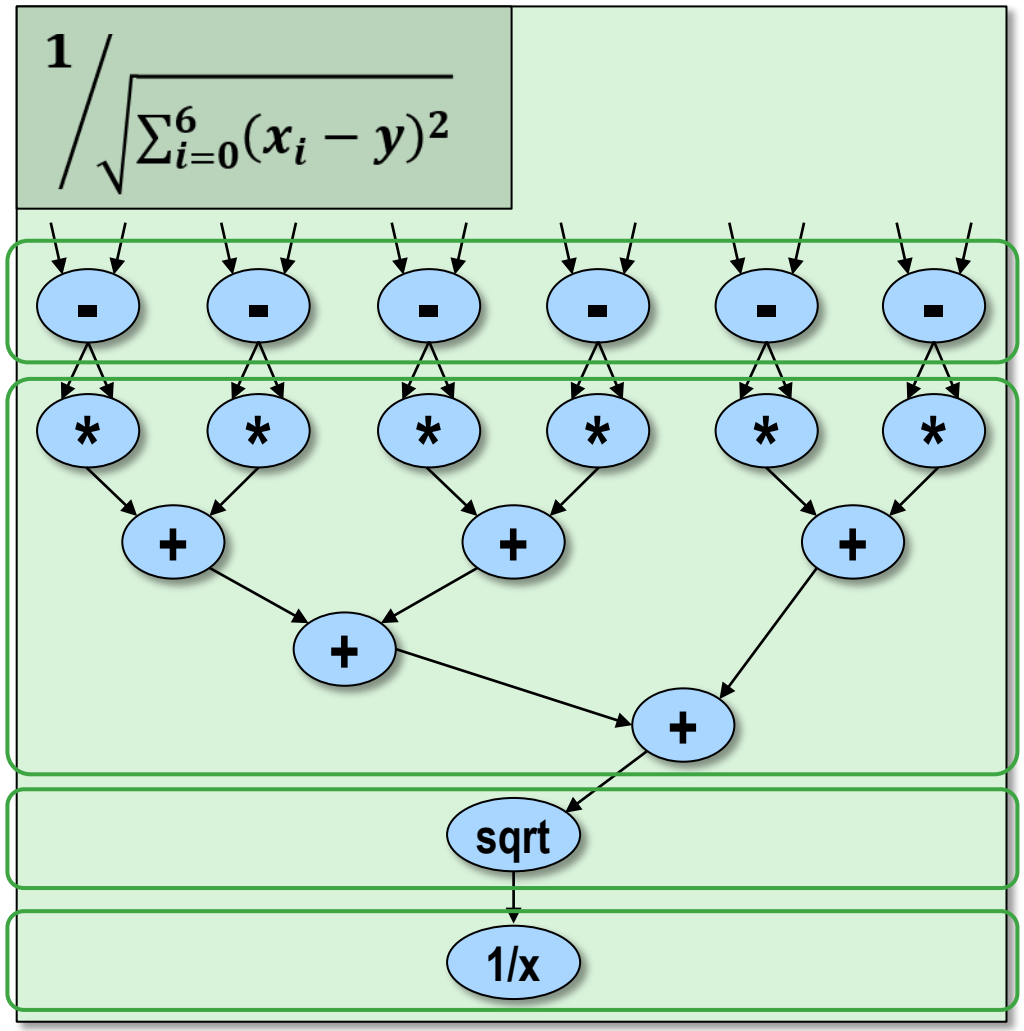


[Cong, et al, DAC'2012]

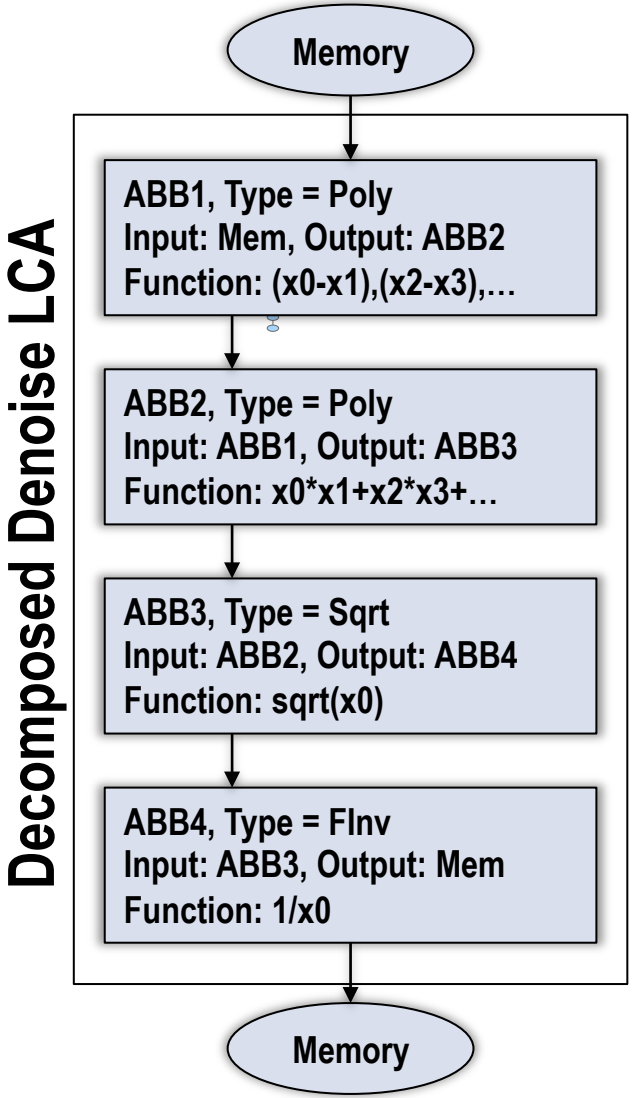
Composable Accelerators



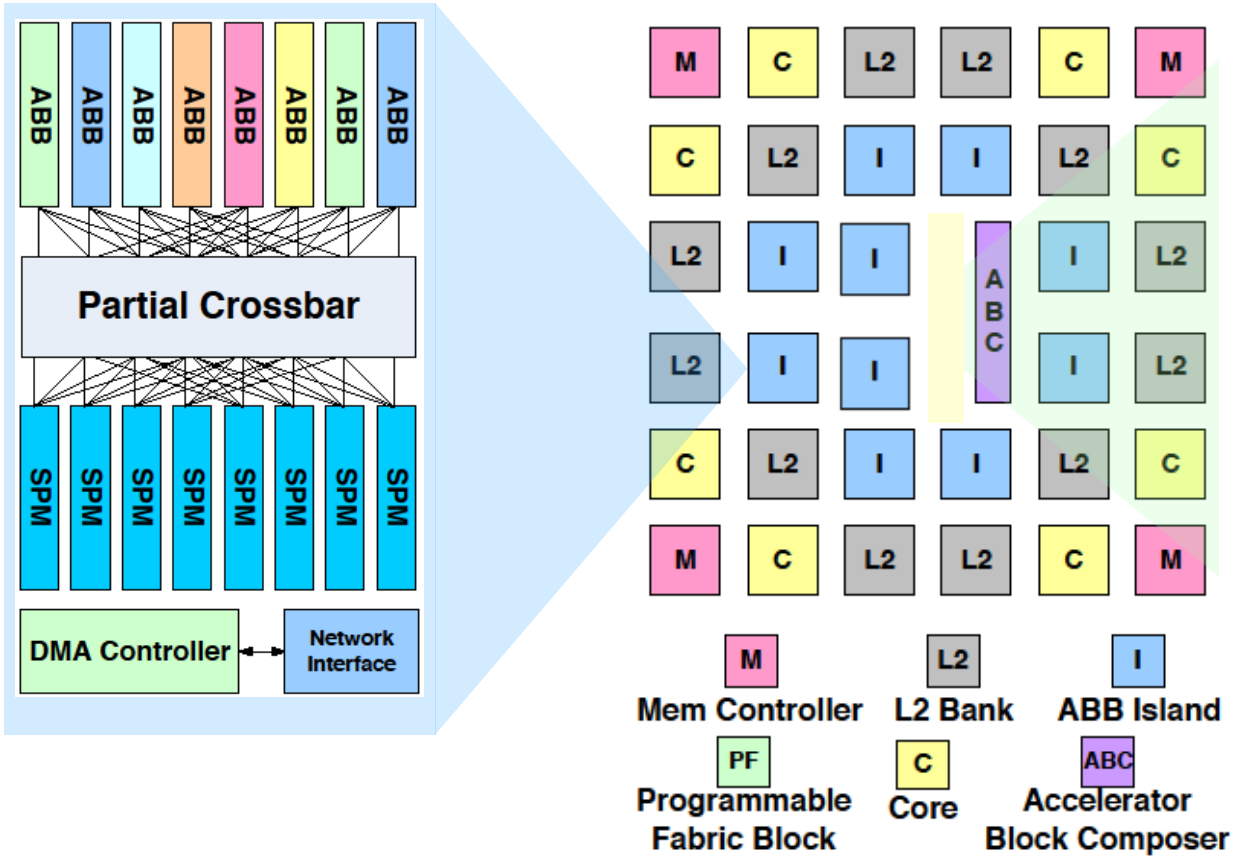
Composable Accelerators



Static Decomposition into ABBs



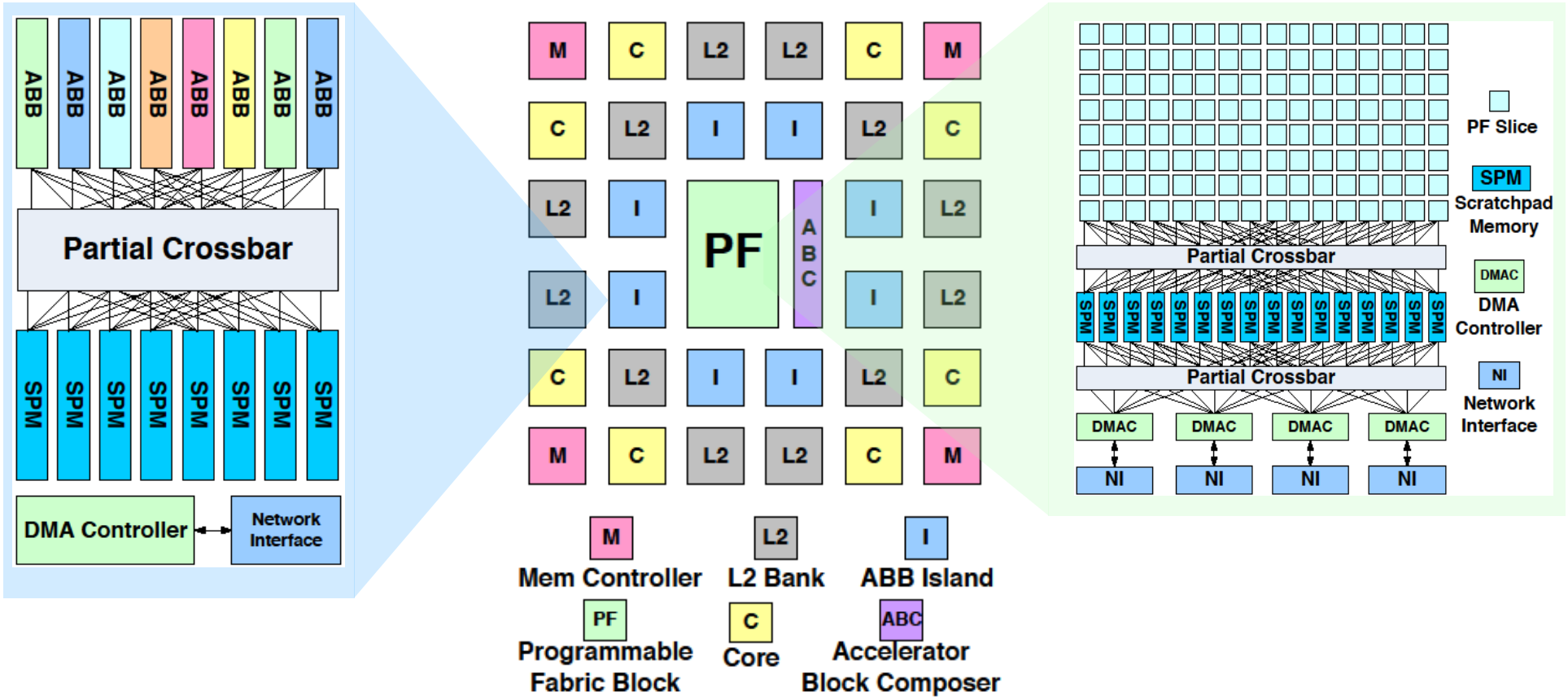
Composable Accelerators



Dynamic Resource Allocation of ABBs

Cong, Ghodrat, Gill, Grigorian and Reinman. "CHARM: A Composable Heterogeneous Accelerator-Rich Microprocessor." ISLPED 2012

Composable Accelerators



Cong, Ghodrat, Gill, Grigorian and Reinman. "CHARM: A Composable Heterogeneous Accelerator-Rich Microprocessor." ISLPED 2012

Results

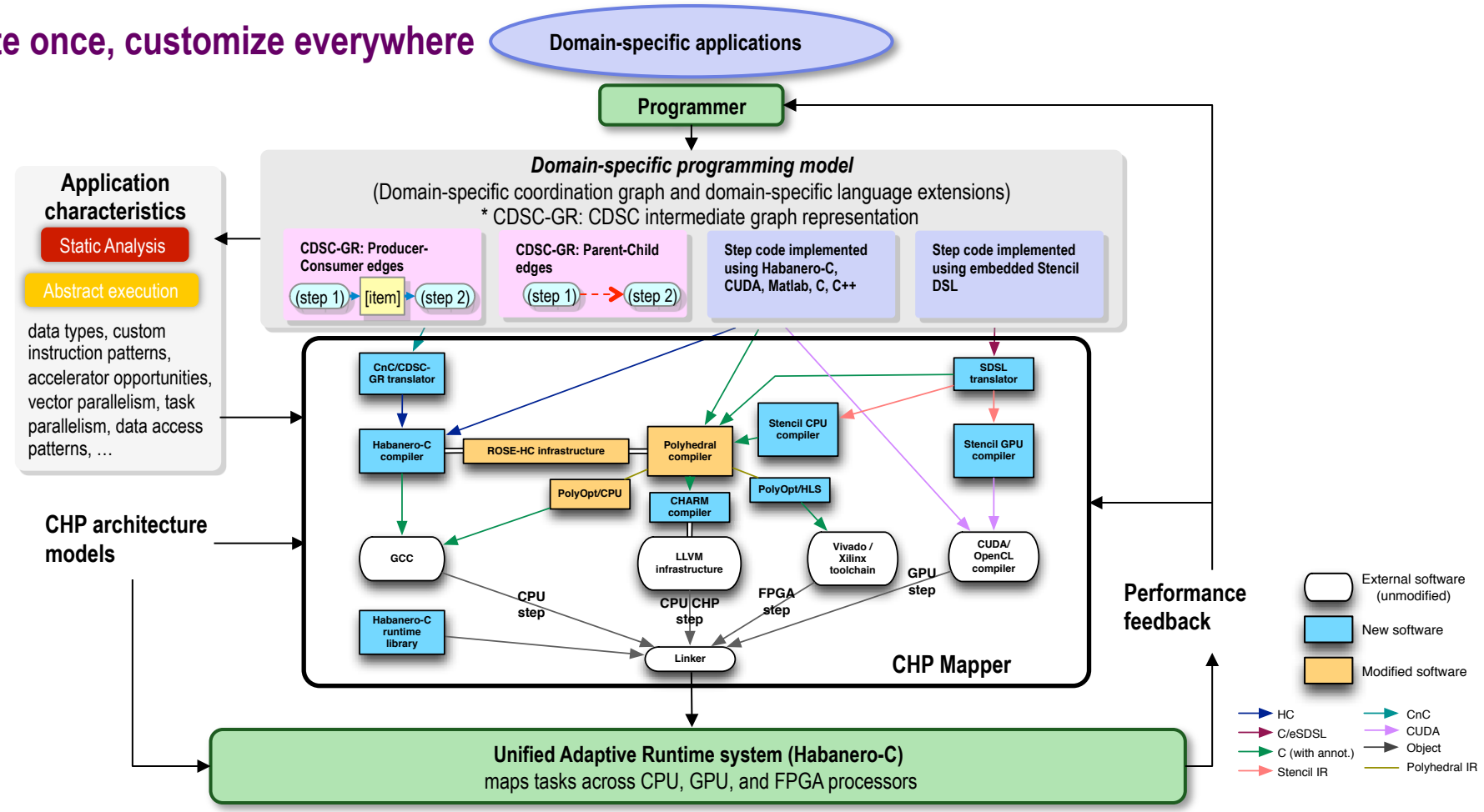
		GPU (NVIDIA Tesla M2075)	FPGA (Xilinx V6)	Monolithic Accelerators	Composable Accelerators
Deblur	Performance	97X	25X	58X	107X
	Energy	19X	130X	369X	261X
Denoise	Performance	38X	12X	26X	37X
	Energy	7.5X	89X	327X	308X
Segmentation	Performance	52X	78X	79X	155X
	Energy	2.4X	371X	201X	149X
Registration	Performance	32X	24X	53X	109X
	Energy	27.8X	31X	854X	1102X
Average	**Performance**	**50X**	**27X**	**50X**	**90X**
	Energy	**10X**	**107X**	**379X**	**338X**

Results relative to an Intel Core i7 (L5640 @ 2.27 GHz)
Accelerators are synthesized in 32nm technology

- **Also, with 20% of the chip area dedicated to programmable fabric, we can achieve more:**
  - **Flexibility:** An average 12x (up to 146x) speedup in other domains, such as commercial, vision and navigation
  - **Longevity:** 22x speedup on a new application within the medical imaging domain

Modeling & Mapping for Customizable Heterogeneous Architecture

Write once, customize everywhere



Key innovations:

- Embedded domain-specific language for automatic mapping to heterogeneous hardware [ICS'13]
- Automatic compilation support for composable accelerators
- Polyhedral compilation techniques for FPGA/HLS [FPGA'13]
- Runtime system for heterogeneous systems (mCPU+GPU+FPGA) [LCTES'12]

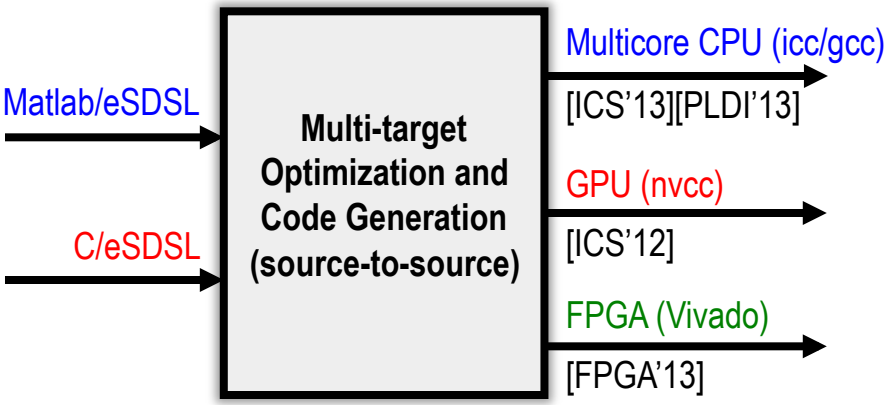
Example: Medical imaging pipeline results on Convey HC1-ex (baseline: Intel ICC)

- Fully automated CDSC Mapper flow using Stencil-DSL (no auto-tuning)
  Improvements: CPU: 1.2x to 2.8x GPU: 1.1x to 6.1x FPGA: 3.2x to 3.8x
- Best manual implementation (with tuning):
  Improvements: CPU: 1.2x to 2.8x GPU: 22x to 38x FPGA: 3.8x to 26x

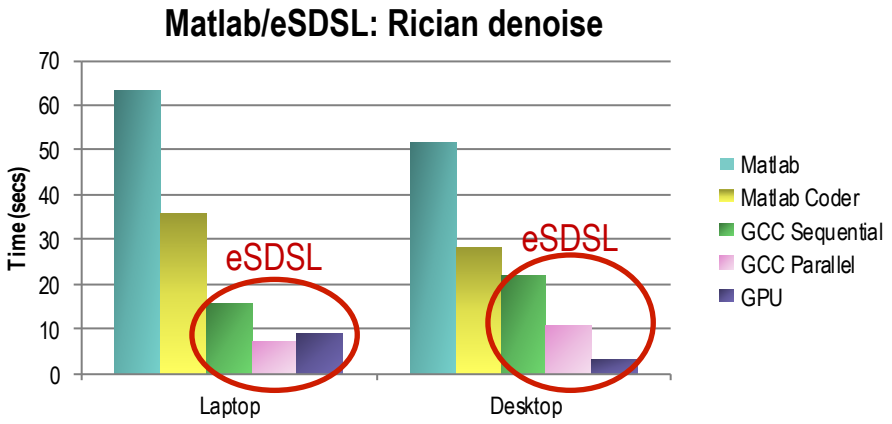
Highlight: Embedded Domain-Specific Language

Benefits of high-level specification of computations using domain-specific languages:

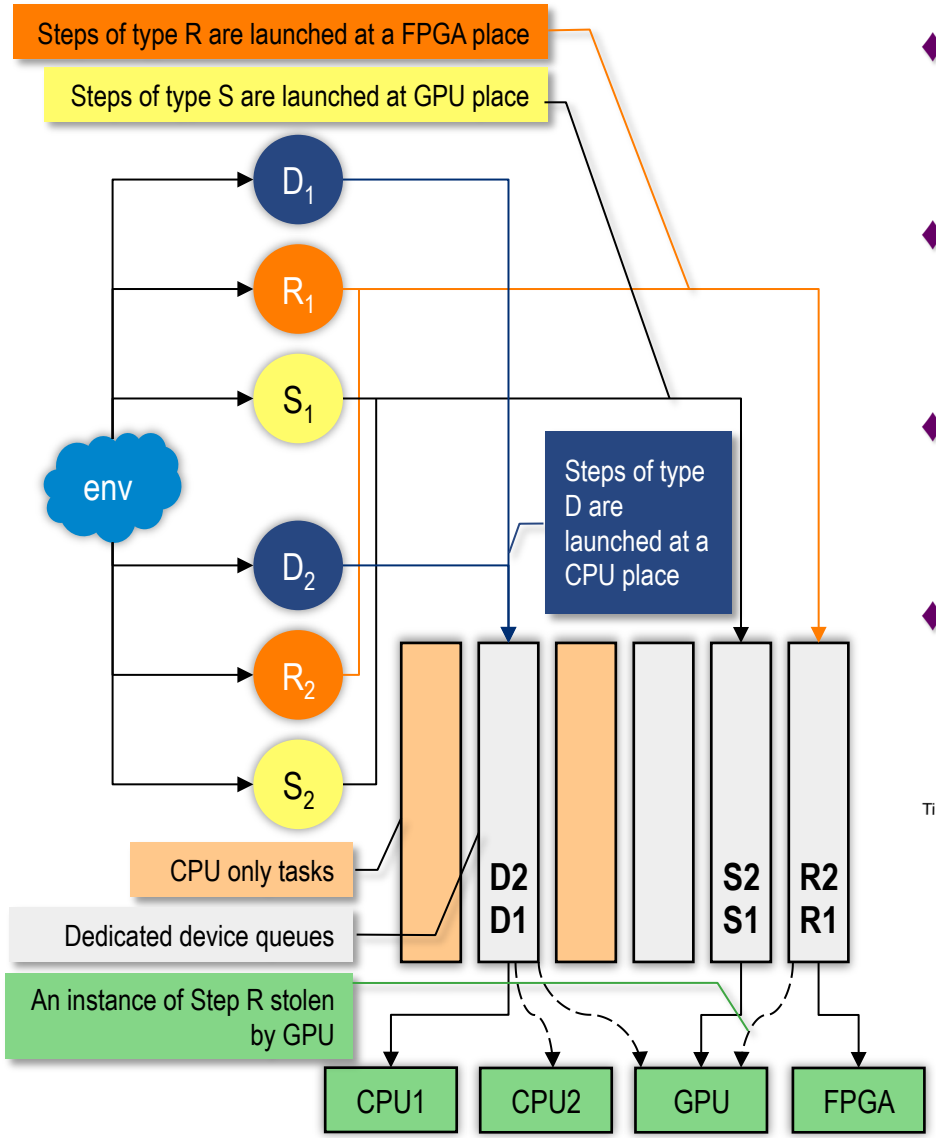
- Ease of use (for mathematicians/scientists creating the code)
- Ease of optimization (facilitate loop and data transformations)
- Embedded DSL provides flexibility:
- Generality of standard programming language
- Automated transformation of embedded DSL region



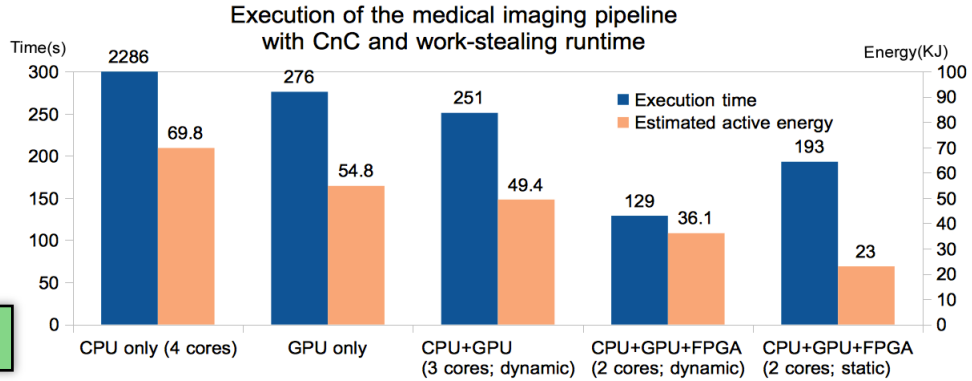

```
int Nr; int Nc;
grid g [Nr][Nc];
double griddata a on g at 0,1;

pointfunction five_point_avg(p) {
  double ONE_FIFTH = 0.2;
  [1]p[0][0] = ONE_FIFTH*([0]p[-1][0] + [0]p[0][-1]
                 + [0]p[0][0] + [0]p[0][1] + [0]p[1][0]);
}
iterate 1000 {
  stencil jacobi_2d {
    [0     ][0:Nc-1] : [1]a[0][0] = [0]a[0][0];
    [Nr-1  ][0:Nc-1] : [1]a[0][0] = [0]a[0][0];
    [0:Nr-1][0     ] : [1]a[0][0] = [0]a[0][0];
    [0:Nr-1][Nc-1  ] : [1]a[0][0] = [0]a[0][0];
    [1:Nr-2][1:Nc-2] : five_point_avg(a);
  }
  reduction max_diff max {
    [0:Nr-1][0:Nc-1] : fabs([1]a[0][0] - [0]a[0][0]);
  }
} check (max_diff < .00001) every 4 iterations
```



Highlight: Use of CDSC Unified Adaptive Runtime System for Heterogeneous Scheduling



- CDSC-GR supports a dynamic dataflow model, without requiring that an underlying sequential program be provided
- Each task in a CDSC-GR program can be compiled for execution on multiple heterogeneous processors
- An adaptive runtime system dynamically decides which processor should execute a given task
- To the best of our knowledge, this is the first system with the above characteristics



Sbirlea, Zou, Budimlic, Cong, Sarkar. "Mapping a Data-Flow Programming Model onto Heterogeneous Platforms." LCTES 2012

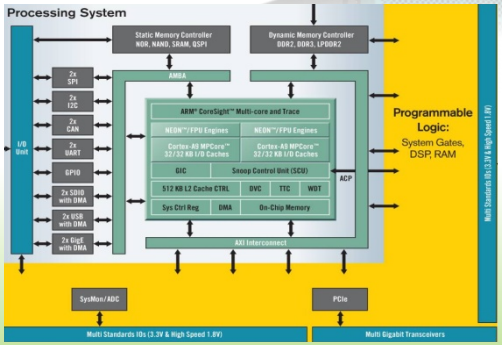
Experimental Platform Thrust: Progress Overview

Server-class platform
CPU, GPU, FPGA, etc.
(e.g., Convey HC-1)
High performance; good energy efficiency

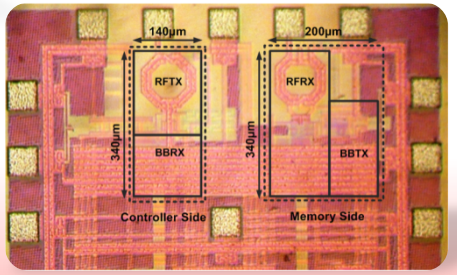
Server/client model
Computation vs. display

Mobile platform
SOC, etc.
(e.g., Tegra 2)
Low power, good energy efficiency

Mobile components in server platforms

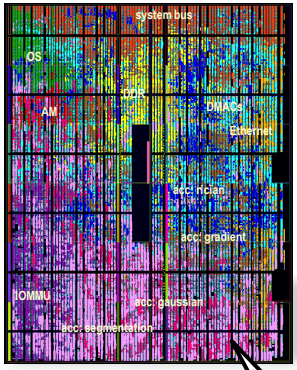


Field Programmable SoC
Zynq SoC (courtesy of Xilinx)

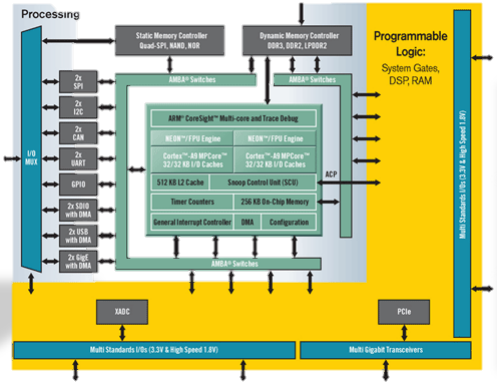


RF Interconnects
High bandwidth, programmable interface

Highlight: Prototyping of Accelerator-Rich Platform in FPGA



Xilinx Virtex6 FPGA chip in ML605 board

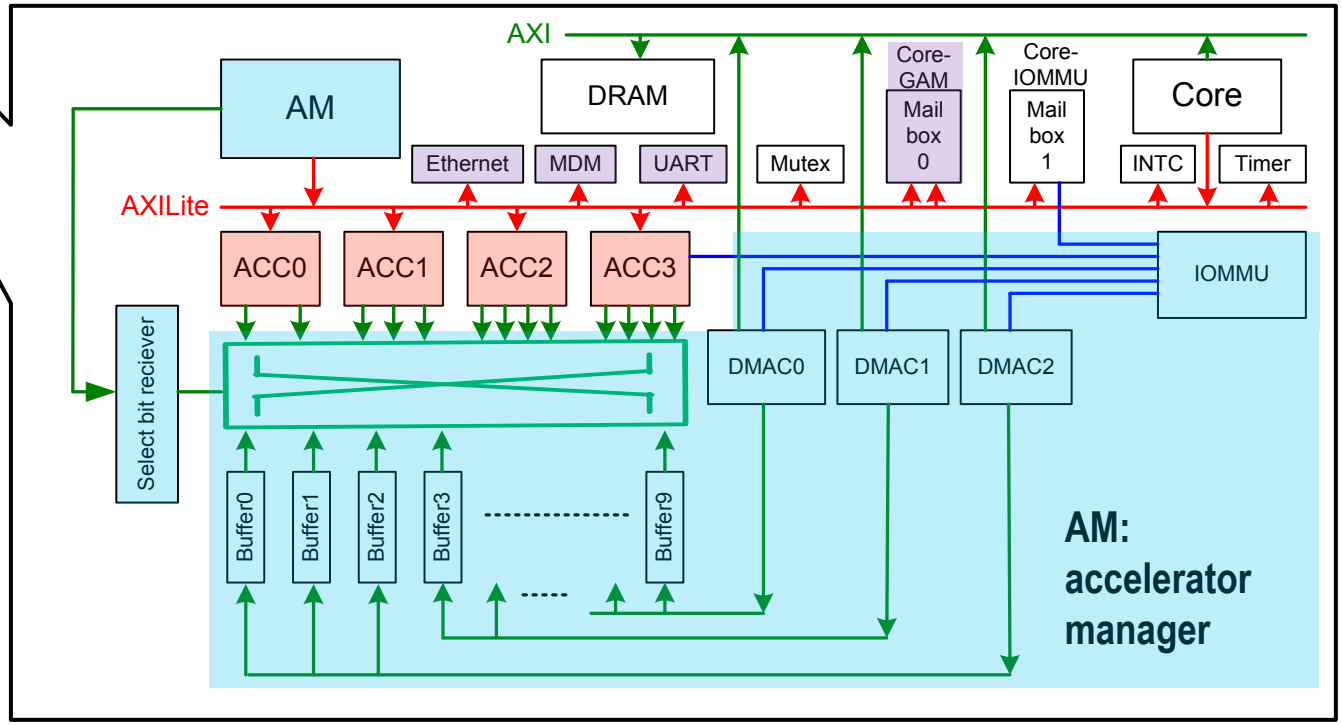


Zynq chip with dual-core A9 and programmable logics

Domain-Specific Accelerators +

Platform Independent Modules +

Platform Specific Modules



Design Flow of Accelerator-Rich Platform

Creation of CHP prototype

targeting an application domain with common kernels to be accelerated

↓

design accelerators purely in C which is enabled by system support

↓

modify platform configuration file (cfg.xml) and platform generated by automation flow

↓

write host applications with accelerators accessible by object-oriented library

↓

run host applications in OS and accelerators scheduled for QoS

Mapping for CHP prototype

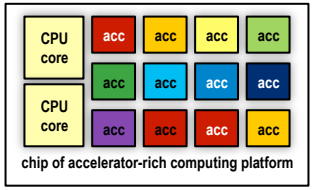
original code (C or openCL) of an application to be accelerated

↓

kernel identification, and performance/power evaluation of HW/SW implementation

↓

global optimization of computation, shared memory and interconnect among kernels



↓

generation of optimized code and application-specific IPs

Example: Add 'denoise' kernel to CHP

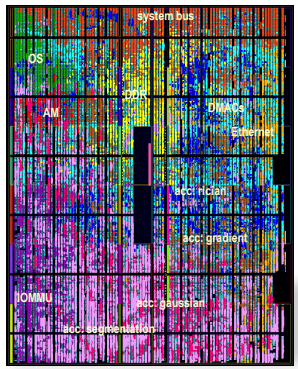
input	# of code lines to write
original kernel code	52
platform cfg.xml	4

→

	Automatically-generated component	# of code lines to write	# of lines in RTL after HLS synthesis
domain-specific	accelerator (.c)	544	11,153
platform modules	accelerator manager (.c)	113	--
	IOMMU (.c)	200	4,096
	crossbar (.v)	2240	--
	system interconnects (.mhs)	542	--
	total	3,095	→ 6,991
total		**3,639 (65x)**	**→ 18,144 (324x)**

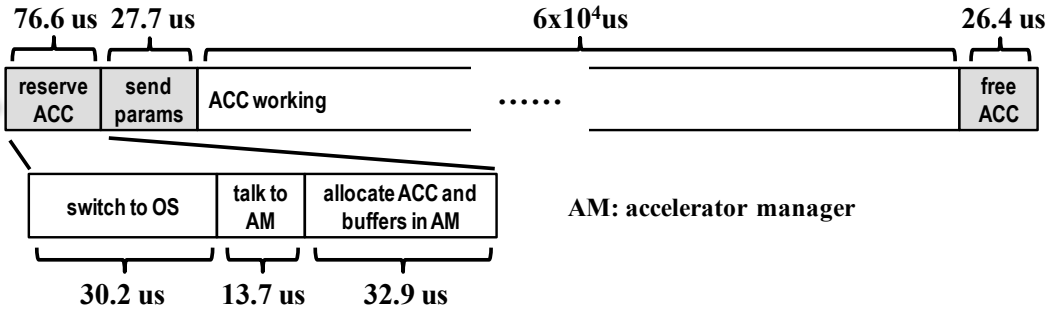
Experimental Results in FPGA Prototyping

- 4 MI kernels on chip (gradient, Rician, Gaussian, segmentation)
- 3D image size: 128 x 128 x 128



		Segmentation	Gaussian	Gradient + Rician
8-core Xeon Server E5405 @ 2GHz	runtime (s)	0.405	0.109	0.106
	energy (J)	4.056	1.064	0.992
Dual-Core ARM Cortex-A9 MPCore @ 800MHz (28nm)	runtime (s)	0.597 (0.67x)	0.301 (0.36x)	0.862 (0.12x)
	energy (J)	0.299 (13x)	0.150 (7.1x)	0.431 (2.3x)
Accelerator in our platform @ 100MHz	runtime (s)	0.056 (7.2x)	0.066 (1.7x)	0.060 (1.8x)
	energy (J)	0.028 (144x)	0.034 (31x)	0.030 (33x)

Little overhead imposed by our platform on accelerators



Some Key Statistics of CDSC

- **People**
  - Faculty: 13 (UCLA – 9; Rice – 2; Ohio-State – 1; UC Santa Barbara – 1)
  - Graduate students: 41
  - Postdocs, research scientists, associate faculty: 17
- **Publications: 179**
  - 2009 -10: 34
  - 2011: 75
  - 2012: 55
  - 2013: 15
- **Keynote/invited talks: 56**
  - 2009-10: 24
  - 2011: 23
  - 2012: 25
  - 2013: 5
- **New courses: 11**
- **PhD students graduated from CDSC: 12**
- **Industry advisory board – Broadcom, HP, IBM, Intel, Siemens, and Xilinx**
  - Center-wide reviews twice a year with good participation from the industry

Selected Awards

- Cong, Jiang, Liu and Zou, "Automatic memory partitioning and scheduling for throughput and power optimization", **TODAES'2013 Best Paper Award.**
- Pouchet, Zhang, Sadayappan and Cong, "Polyhedral-Based Data Reuse Optimization for Configurable Computing", **FPGA'2013 Best Paper Award.**
- Murphy, Darabi, Abidi, Hafez, Mirzaei, Mikhemar and Chang, "A Blocker-Tolerant Wideband Noise-Cancelling Receiver with a 2dB Noise Figure", **2012 IEEE ISSCC Distinguished Technical Paper Award and Jack Kilby Best Student Paper Award**
- Cong, Liu, Majumdar and Zhang, "Behavior-Level Observability Analysis for Operation Gating in Low-Power Behavioral Synthesis", **TODAES'2012 Best Paper Award.**
- **Outstanding Masters Graduate Award** for Professor Miodrag Potkonjak's PhD student, Saro Meguerdichian, 2012
- Papakonstantinou, Liang, Stratton, Gururaj, Chen, Hwu and Cong, "Multilevel Granularity Parallelism Synthesis on FPGAs", **FCCM'2011 Best Paper Award.**
- Shamshiri, Ghofrani and Cheng, "End-toEnd Error Correction and Online Diagnosis for On-Chip Networks", **ITC'2011 Best Student Paper Award.**
- **…**

What Does Expedition Project Enables

- **Taking novel, transformative approach as opposed to incremental improvements**
  - Need substantial new infrastructure development
  - Example: Accelerator-centric architecture
- **Multi-disciplinary collaboration, e.g.**
  - Real applications, real targets
  - SW + HW – From modeling (CDSC-GR) to implementation (FPGA)
  - CS + EE – Use of RF-I as customizable interconnects
- **Impact to the application domain**
  - "*New real-time clinical applications can begin to be realized via CDSC's effort*" – collaborators in UCLA radiology department

Concluding Remarks

"In this project we look beyond parallelization and focus on *domain-specific customization* as the *next disruptive technology* to bring orders-of-magnitude power-performance efficiency improvement to important application domains."

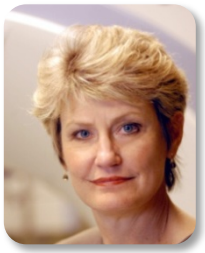
– CDSC proposal (2009)

We are making significant progress in achieving this goal with advancements in the following directions:

- Novel customizable heterogeneous computing platforms
- Unified modeling, compilation, and runtime system
- Demonstration in the medical imaging application domain

Center for Domain-Specific Computing (CDSC) Organization

A diversified & highly accomplished team: 8 in CS&E; 1 in EE; 3 in medical school; 1 in applied math


Aberle


Baraniuk


Bui


Chang


Chien


Cheng


Cong (Director)

	UCLA	Rice	UCSB	Ohio State
Domain-specific modeling	Bui, Reinman, Potkonjak	**Sarkar,** Baraniuk		Sadayappan
CHP creation	Chang, Cong, **Reinman**		Cheng	
CHP mapping	Cong, Palsberg, Potkonjak	**Sarkar**	Cheng	Sadayappan
Application drivers	Aberle, **Bui,** Chien, Vese	Baraniuk		
Experimental systems	All (led by Cong & Bui)	All	All	All


Palsberg


Potkonjak


Reinman


Sadayappan


Sarkar (Associate Dir)


Vese