MANCHESTER
1824
The University of Manchester

Advanced Processor Technologies Group

Run-Time Reconfigurable CPU Interlays for Building Flexible ARM SoCs

Raul Garcia and **Dirk Koch**
School of Computer Science, The University of Manchester
{raul.garcia, dirk.koch}@manchester.ac.uk

MANCHESTER
1824
The University of Manchester

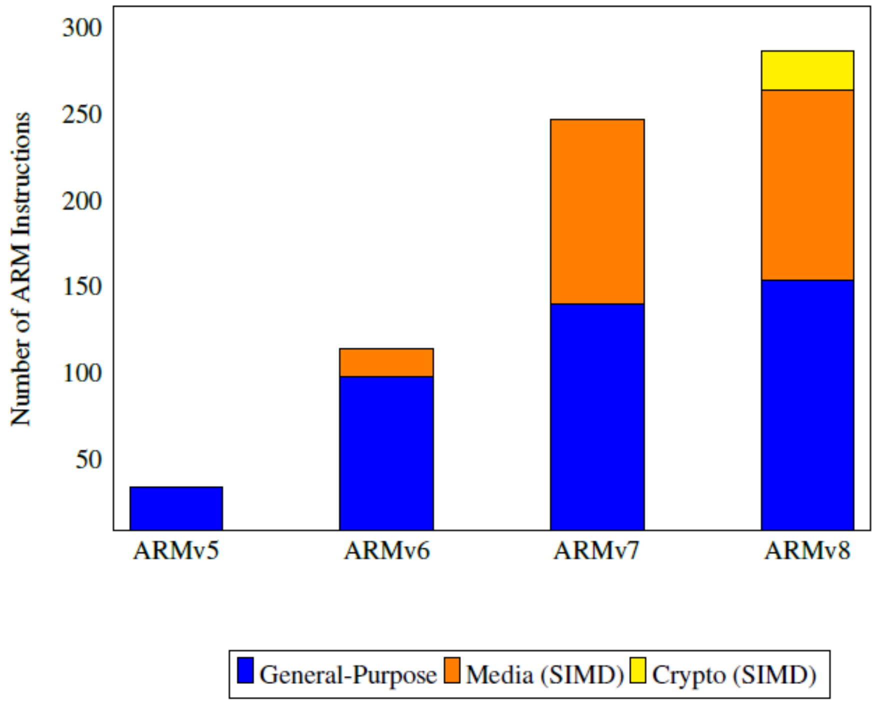
APT
Advanced Processor Technologies Group

Agenda

- Introduction
- CPU Interlays
- Custom Instructions for Interlays
- Interlays in SoC Designs
- Conclusions

Introduction

Existing general-purpose (GP) CPU architectures tend to provide feature-rich instructions sets (Instruction Set Extensions)
- ISEs introduce substantial area/energy overhead
- CPU clock is limited by power



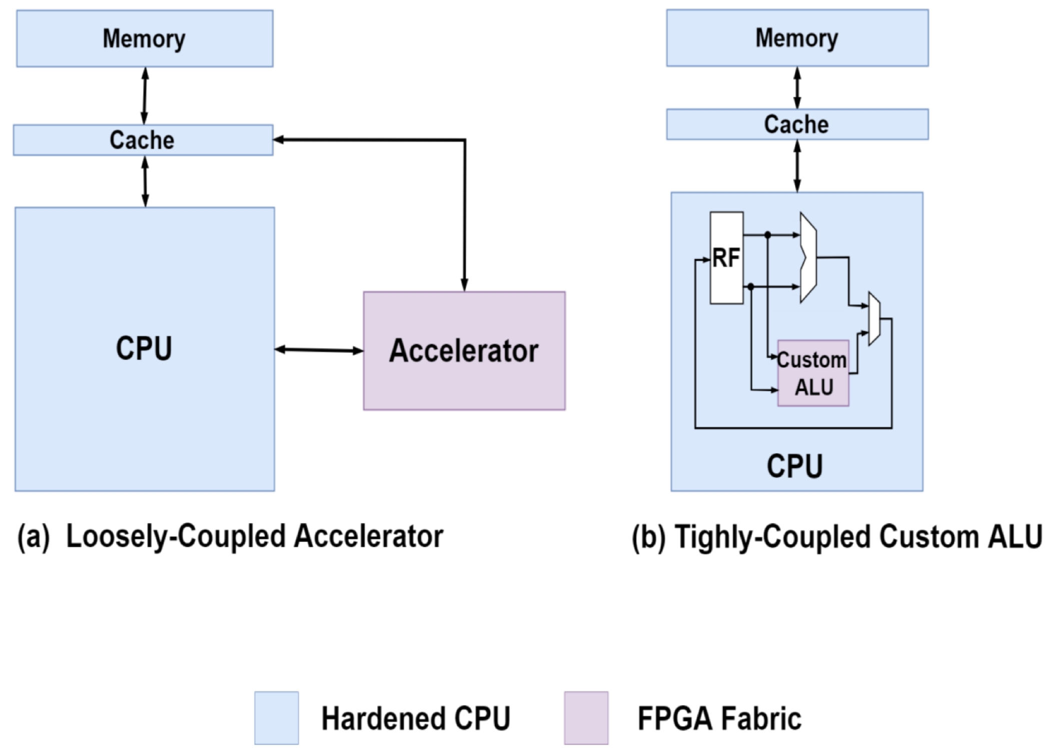
MANCHESTER 1824
The University of Manchester

APT
Advanced Processor Technologies Group

Alternatively, hardened GP CPUs can be coupled to reconfigurable fabrics to implement custom accelerators/ALUs

Advantages:

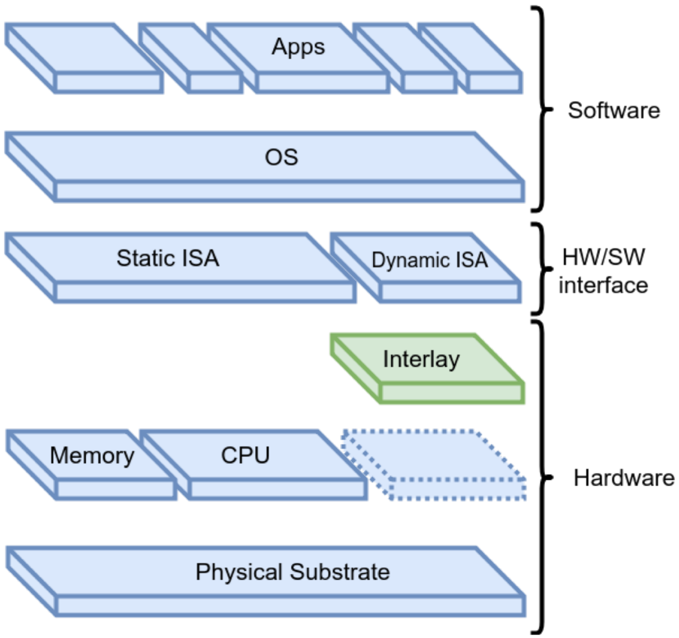
- Customization
- Resource Sharing
- In-field updates



(a) Loosely-Coupled Accelerator (b) Tighly-Coupled Custom ALU

CPU Interlays

This work explores the potential advantages that can be obtained by embedding a tiny FPGA fabric into an otherwise hardened CPU. We call this tiny reconfigurable fabric an *Interlay* as it sits between the software layer and the physical substrate



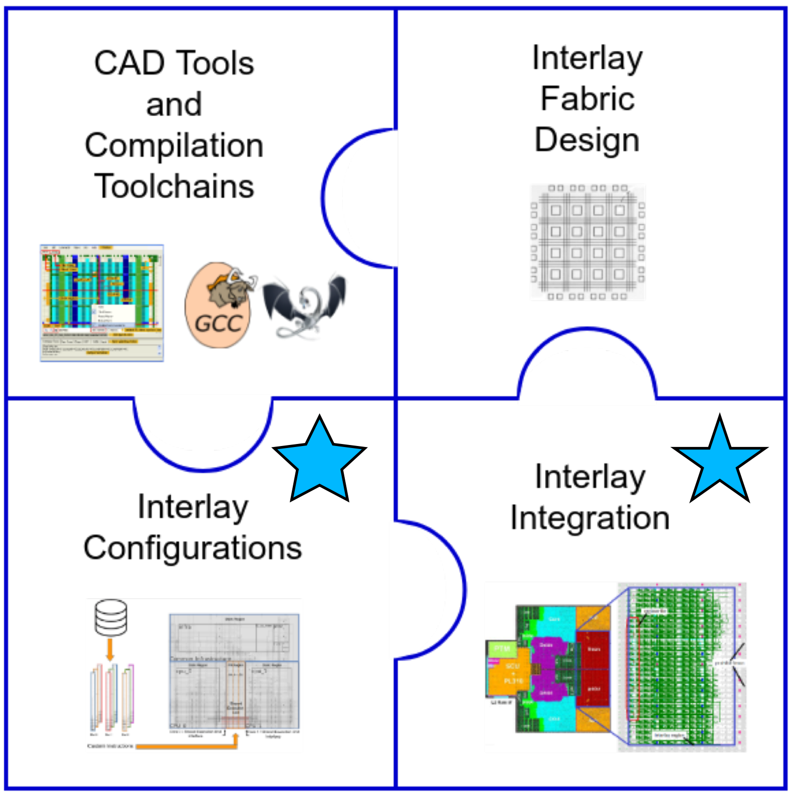
Characteristics:

- Tightly-coupled to a hardened CPU as a custom ALU
- Targeted to accelerate relatively small kernels
- Leverages the advantages of reconfigurability for an existing design
- Provides efficiency and performance through customization
- Called in user mode

The University of Manchester

Advanced Processor Technologies Group

Interlay Ecosystem and Research Scope



Research Scope

- Integration of the FPGA Interlay into an existing SoC architecture
- Generation of custom instructions (configurations) for the Interlay fabric

Research Methodology and Case Study

Top-Down Approach:

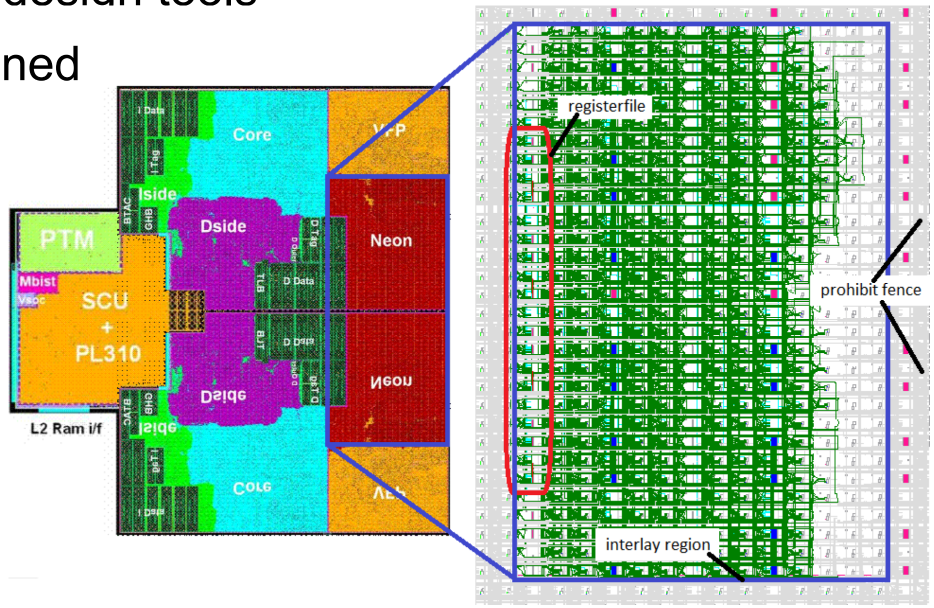
- Adding constraints to an off-the-shelf FPGA to emulate the characteristics of the Interlay

  → Leverage existing design tools

Case Study: Replacing the hardened NEON with an Interlay [1]

Soft NEON allows for:

- Reuse of existing ARM code
- NEON ISA customization
- In-field updates
- Leverage SIMD interface
- Minimal architectural disruptions
- Avoid area/energy overhead

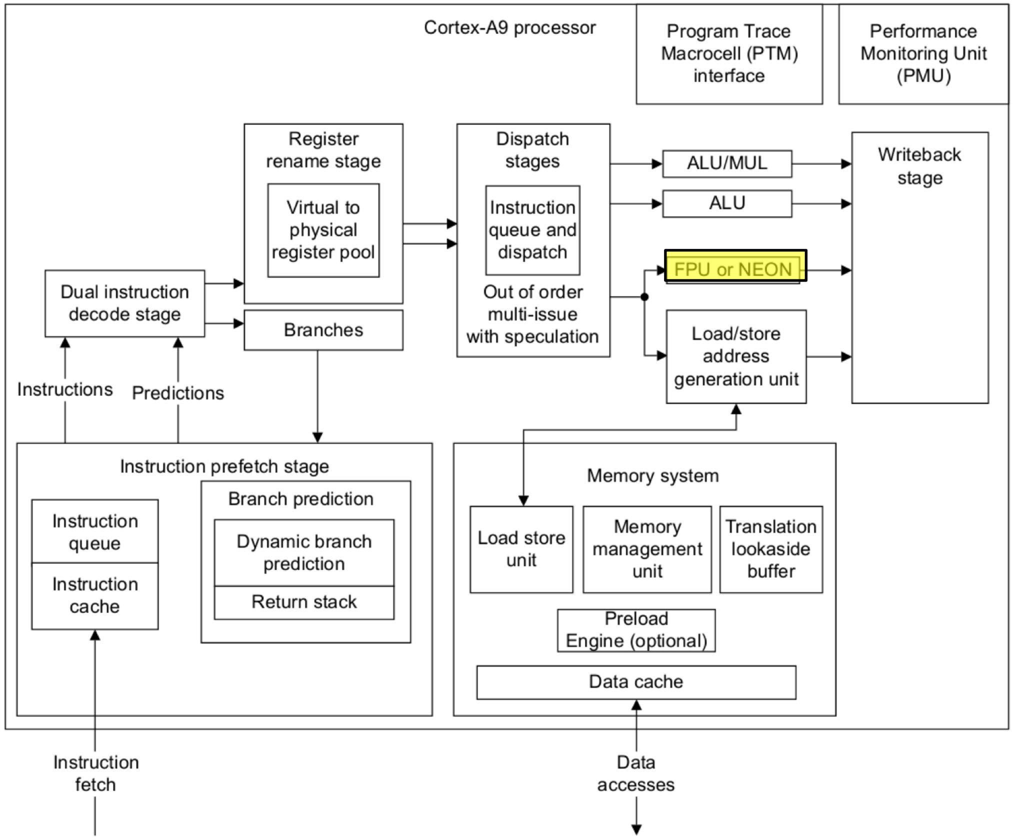


1. Garcia Ordaz, Jose Raul, and Dirk Koch. "Making a case for an ARM Cortex-A9 CPU interlay replacing the NEON SIMD unit." *International Conference on Field Programmable Logic and Applications (FPL),* IEEE, 2017.

MANCHESTER 1824
The University of Manchester

APT
Advanced Processor Technologies Group

About the NEON Engine



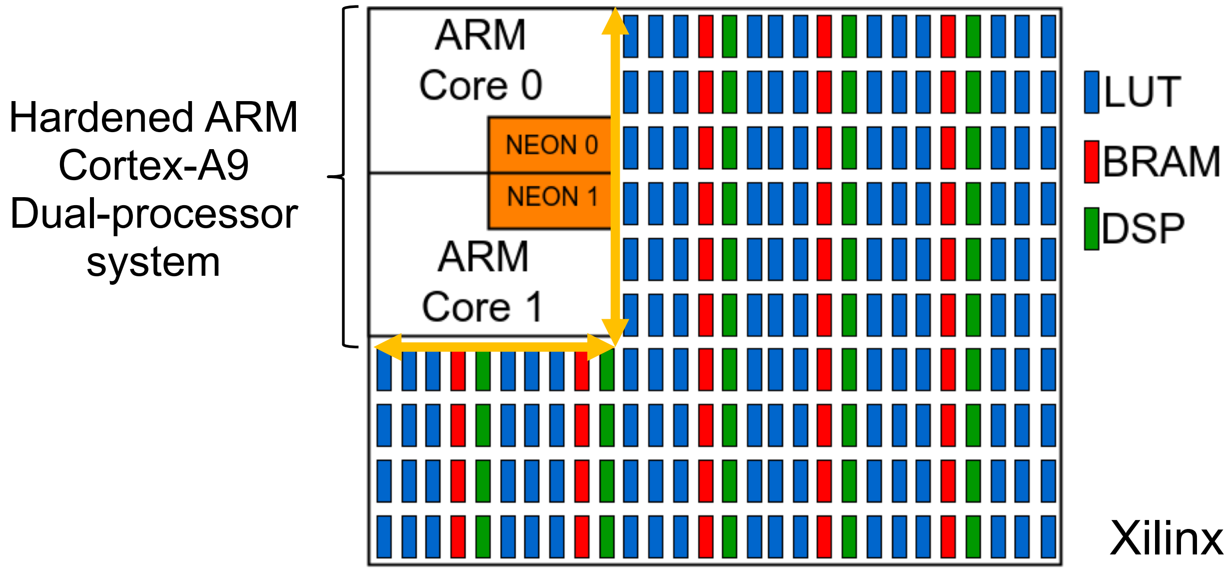
ARM Cortex-A9

NEON SIMD Engine [2]:

- Exploits data-level parallelism
- Targeted mostly to media applications
- Independent vector register file and datapath

2. ARM, Cortex-A9 Technical Reference Manual, Online: www.arm.com

Hardened NEON Area Estimation



Xilinx Zynq Chip

Functional Unit	FPGA Primitive Equivalent		
	LUT	DSP	BRAM
Dual Core ARM Cortex A-9 Processor	10400	80	40
Single Core ARM Cortex A-9 Processor	5200	40	20
Two NEON Units	2080	16	8
Single NEON Unit	1040	8	4

The University of Manchester

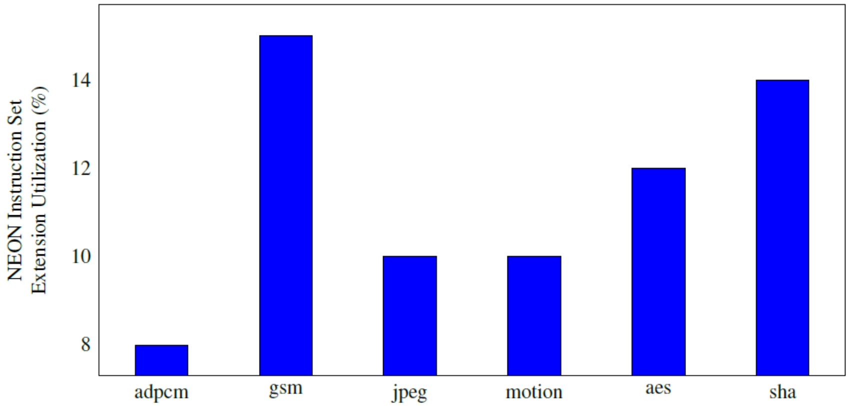
Advanced Processor Technologies Group

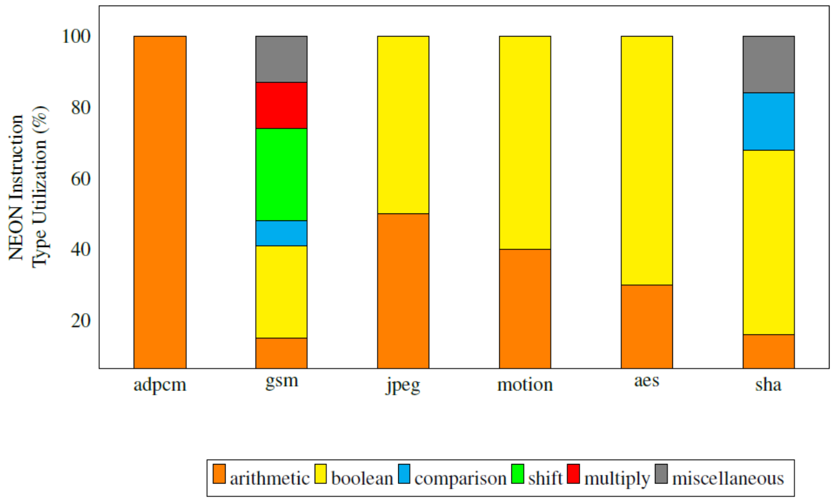
Soft NEON / Hardened NEON Gap Measurement

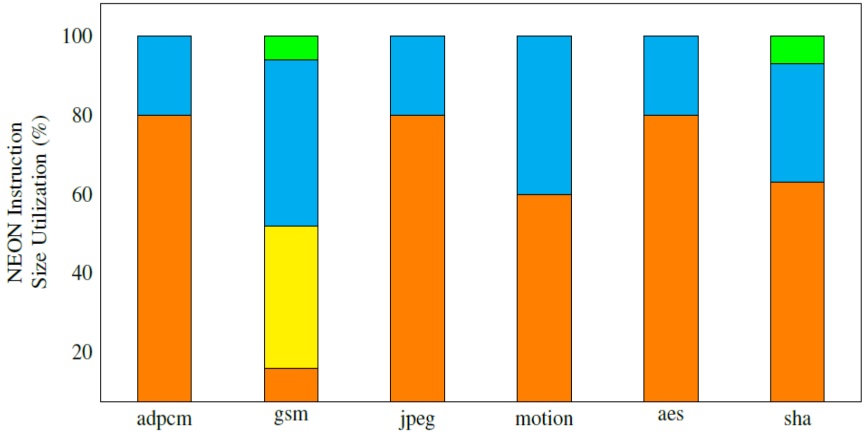
Functional Unit	FPGA Primitive		
	LUT	DSP	BRAM
– NEON ALU	10968	275	0
– arithmetic-ops	640	64	0
– boolean-ops	388	4	0
– comparison-ops	926	0	0
– shift-ops	718	0	0
– multiply-ops	819	88	0
– miscellaneous-ops	7699	119	0
– NEON Register File	0	0	4
– NEON Unit	11360	275	4

FPGA Resources				Operating Frequency	
Hardened NEON		RTL NEON		Hardened NEON	RTL NEON
LUT	DSP	LUT	DSP	MHz	MHz
1040	8	11360 (10.9×)	275 (34.4×)	650	164 (3.9×)

Media/Security Application Profiling





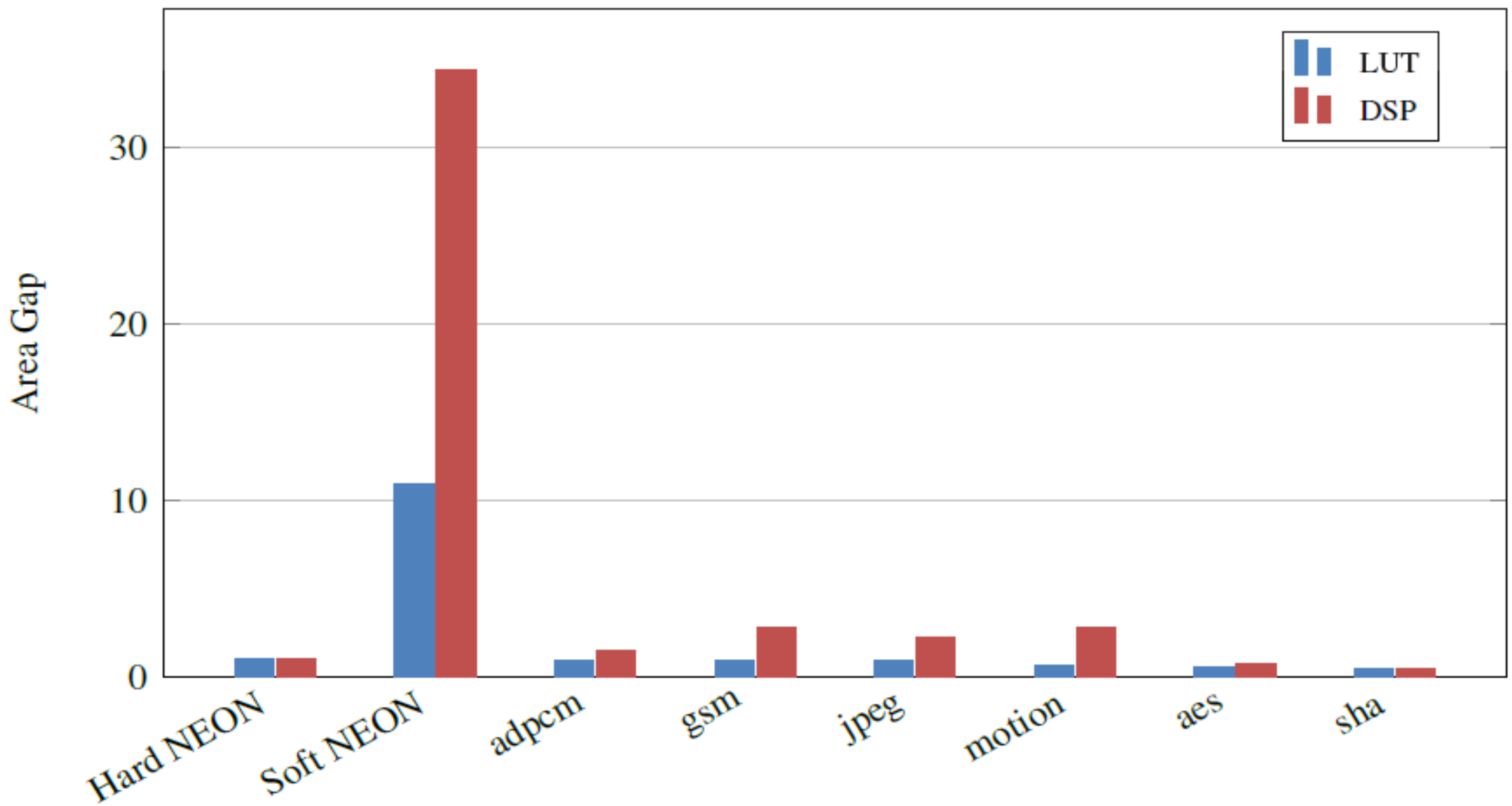


Soft NEON Optimization
- ISA Subsetting
- Vector Width Customization
- Operation Folding

MANCHESTER 1824
The University of Manchester

Advanced Processor Technologies Group

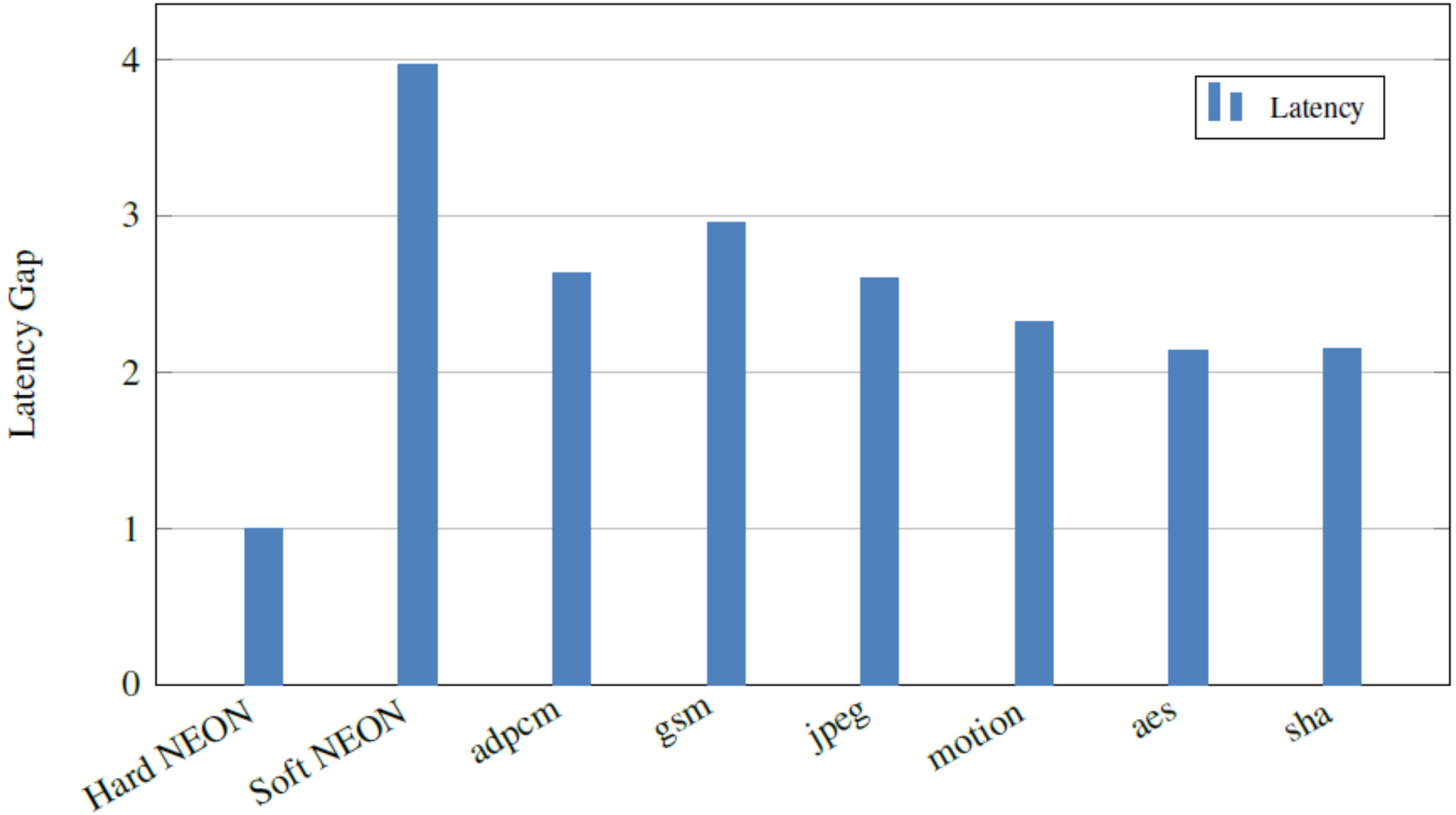
Closing the Soft NEON / Hardened NEON Gap (Area)



MANCHESTER 1824
The University of Manchester

APT
Advanced Processor Technologies Group

Closing the Soft NEON / Hardened NEON Gap (Latency)



Closing the Soft NEON / Hardened NEON Gap (Performance)

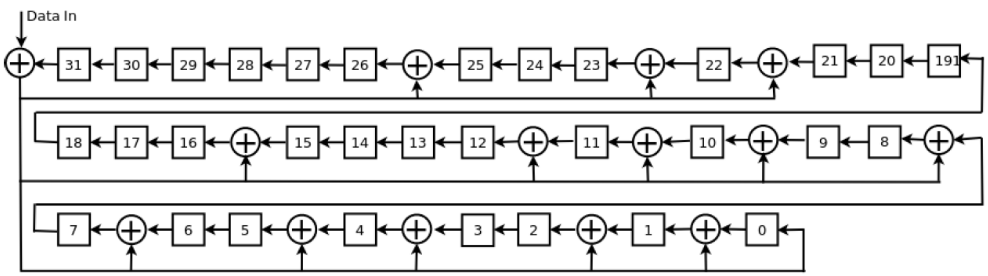
Application	Execution Time (μs)	
	Hardened CPU + Hardened NEON	Hardened CPU + CPU Interlay
adpcm	259	285 (1.10×)
gsm	65	69 (1.06×)
jpeg	5411	6096 (1.13×)
motion	72	74 (1.03×)
aes	202	213 (1.05×)
sha	597	793 (1.33×)

MANCHESTER 1824
The University of Manchester

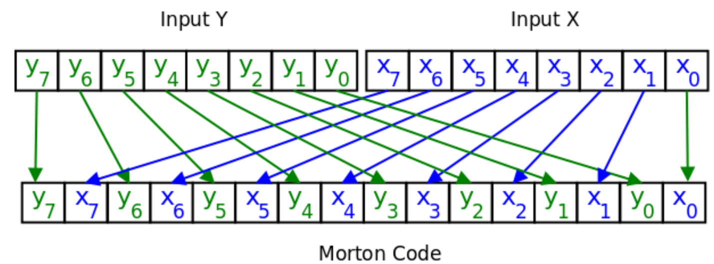
Advanced Processor Technologies Group

Custom Interlay Examples – Bit Manipulation

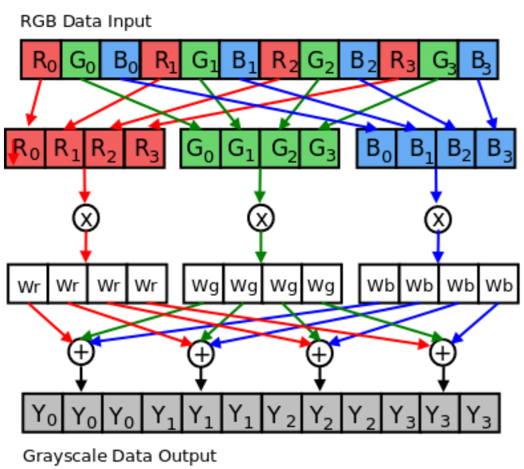
CRC-32 (121 LUTs) → 68x Speedup



Morton Code (223 LUTs → 85x Speedup



RGB2Y (136 LUTS) → 4.9x Speedup



Human Skin Colour Classifier (132 LUTS)
→ 3.6x Speedup

Algorithm 1 Human Skin Colour Classification [KPS03]

```
function SKIN_COLOUR_RGB(RGB Image)                ▷ RGB Data Input
    R > 95 AND G > 40 AND B > 20 AND
    max{R, G, B} - min{R, G, B} > 15 AND
        RGB values must not be closer together
        greyness elimination
    |R - G| > 15 AND
        also R and G values must not be close together
    R > G and R > B
        R value must be the smallest value
end function
```

MANCHESTER 1824

The University of Manchester

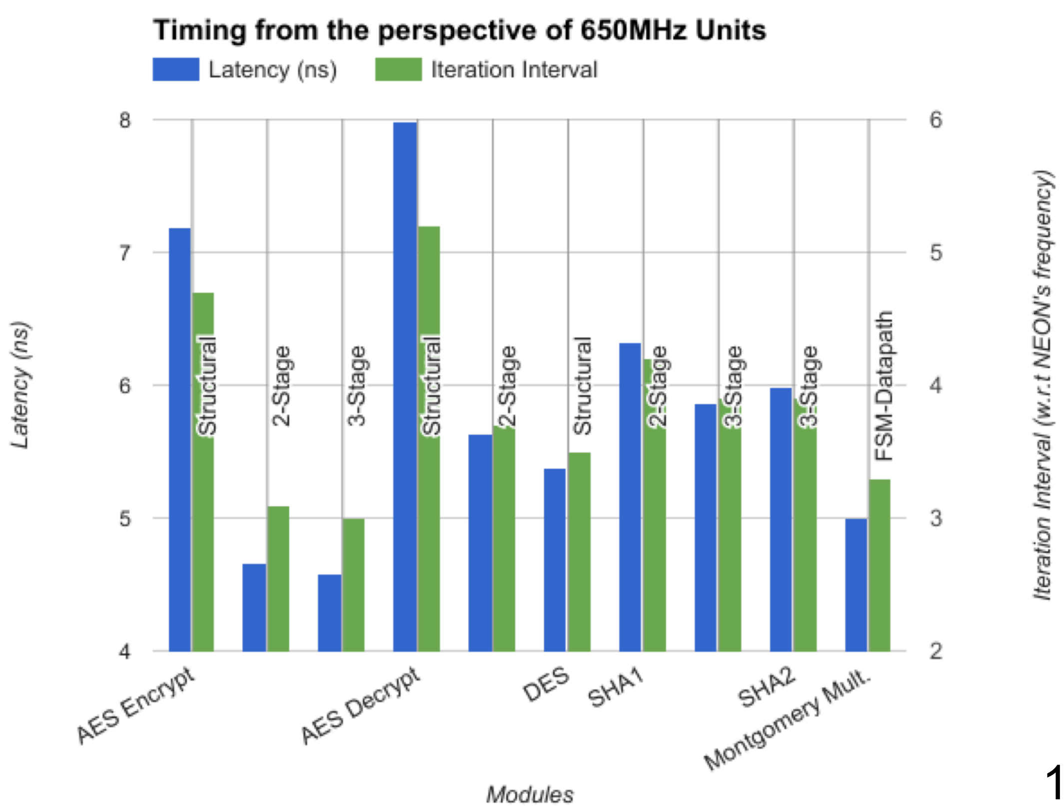
APT

Advanced Processor Technologies Group

Custom Interlay Examples - Crypto

- We implemented a library with all major crypto primitives (AES, DES, SHA, Montgomery multiplication, TRNG)

Module		LUTs	BRAMs
AES encrypt	2-stage-L	809	0
	2-stage-B	172	4
	3-stage-L	899	0
	3-stage-B	397	4
AES decrypt	2-stage-L	1079	0
	2-stage-B	630	4
DES	structural	96	0
SHA1	2-stage	235	0
	3-stage	295	0
SHA2	3-stage	365	0
Montg. mult.	FSM datapath	1014	0
TRNG	structural	544	0

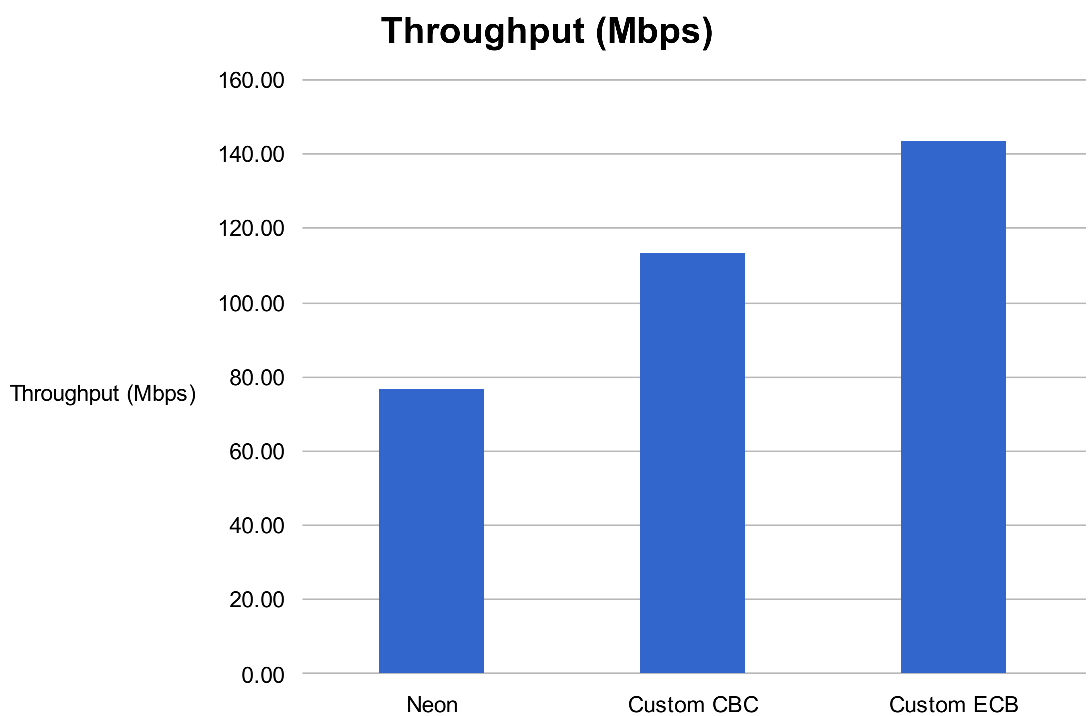


MANCHESTER 1824
The University of Manchester

APT Advanced Processor Technologies Group

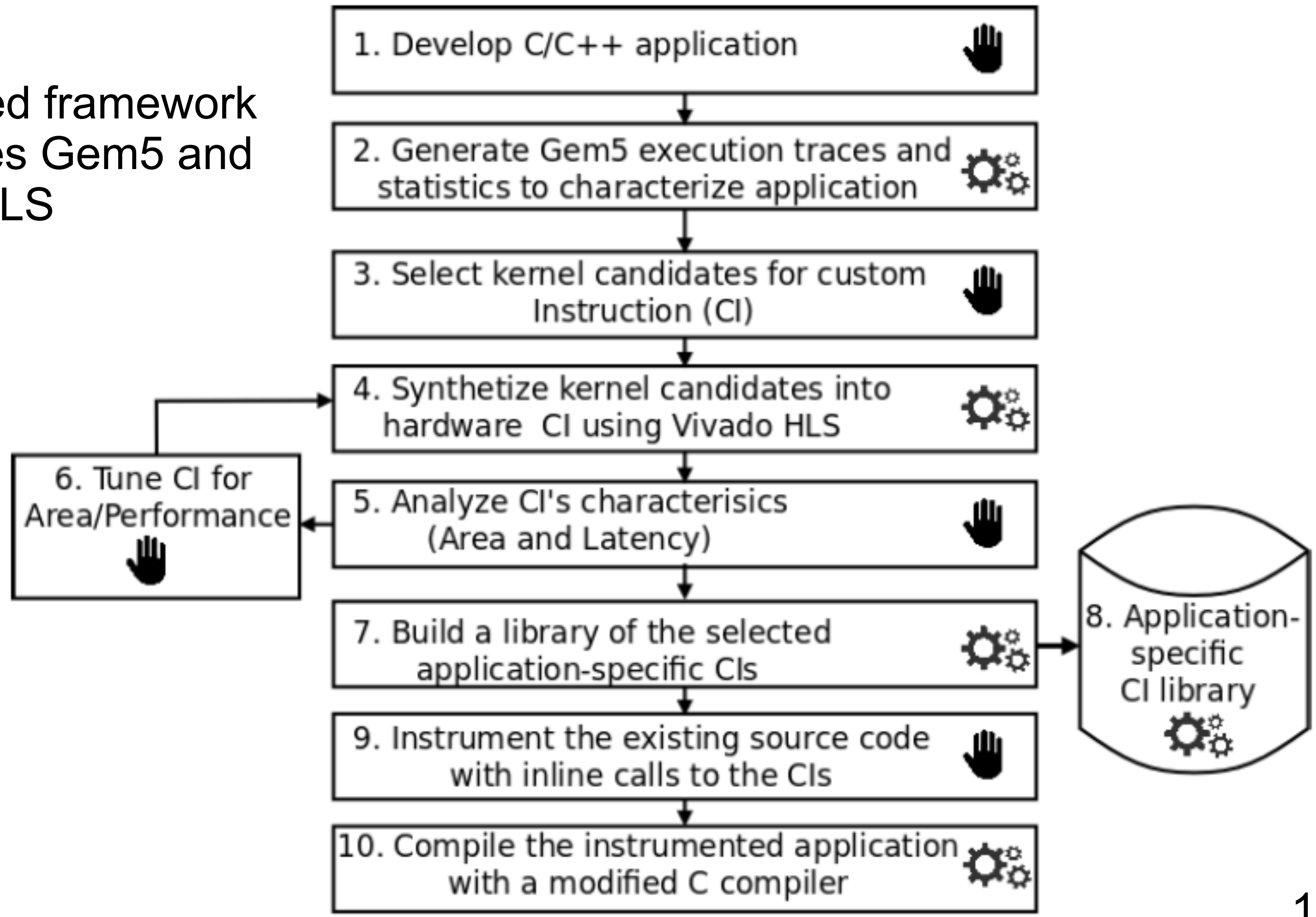
Custom Interlay Examples - Crypto

- Throughput evaluation for Rijndael implementation of MiBench (ARM at 650 MHz coupled with a Zynq-fabric interlay)
- Based on cycle accurate simulation in Gem5



Automatically Generated Interlay CIs

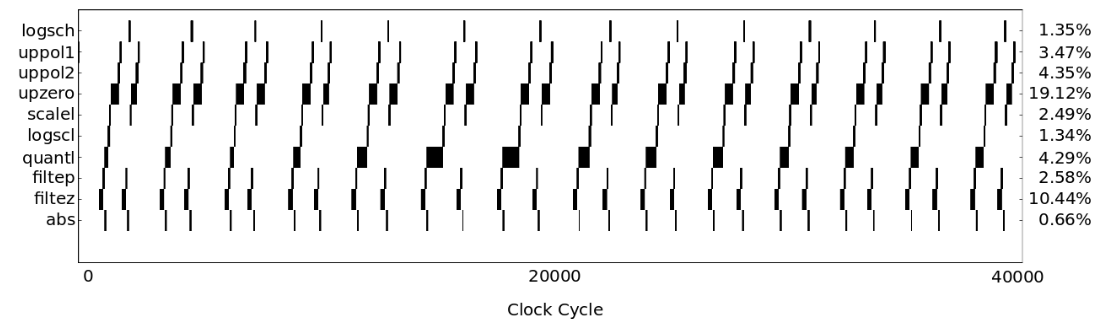
- Function-Oriented framework
  - Leverages Gem5 and Vivado HLS



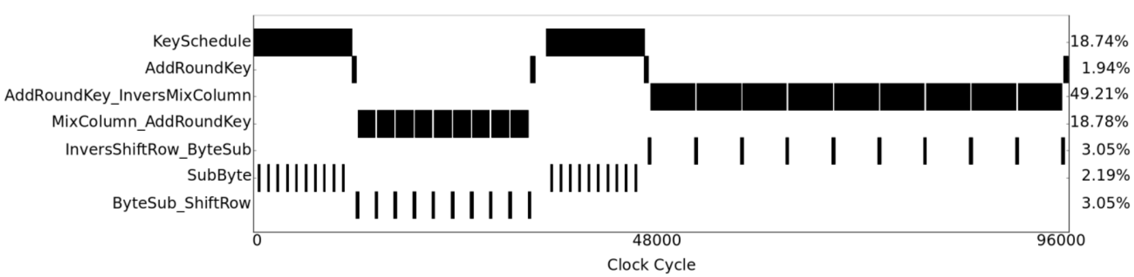
MANCHESTER 1824
The University of Manchester

APT
Advanced Processor Technologies Group

ADPCM



AES

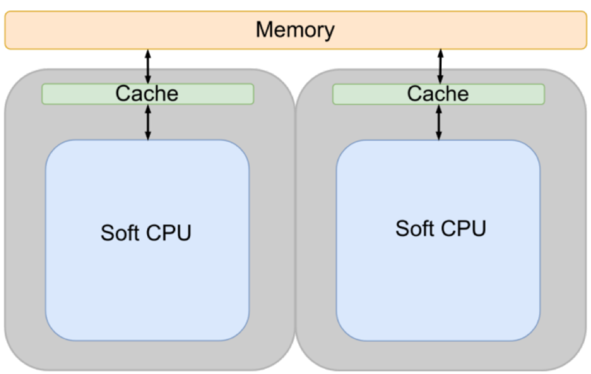


CI	Primitives						Speedup		Exec
	OA			OP			OA	OP	%
	LUT	DSP	Fit	LUT	DSP	Fit			
upzero	337	4	✓	1087	24	✗	1.4	3.3	19.1
filtez	281	8	✓	143	24	✗	1.1	3.8	10.4
uppol2	217	8	✓	300	8	✓	2.8	3.1	4.3
quantl	108	2	✓	1097	54	✗	1.5	14.6	4.2
uppol1	234	4	✓	296	4	✓	2.2	2.9	3.4

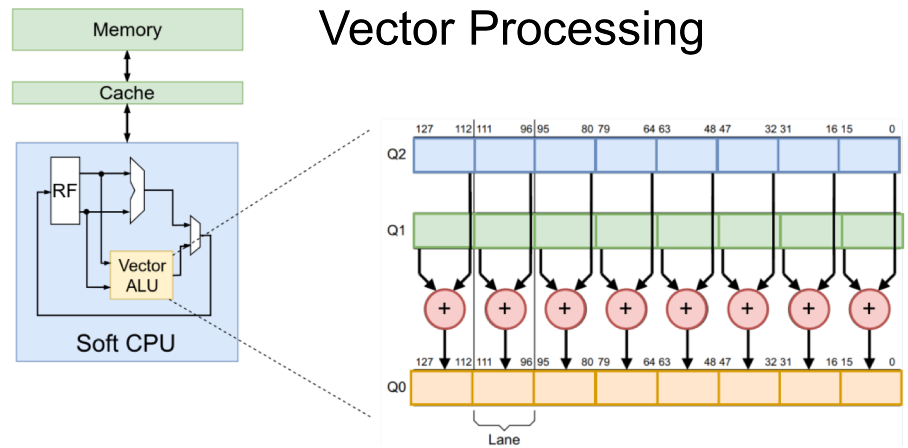
CI	Primitives						Speedup		Exec
	OA			OP			OA	OP	%
	LUT	DSP	Fit	LUT	DSP	Fit			
AddRound Key_Invers MixColumn	2276	4	✓	4971	4	✗	8.9	18.1	49.2
MixColumn_ AddRound Key	2156	4	✓	2156	4	✓	10.3	10.3	18.8
KeySchedule	2310	1	✓	4284	1	✗	3.9	4.6	18.7
ByteSub_ ShiftRow	1291	0	✓	1291	0	✓	20.4	20.4	3.1
Invers ShiftRow_ ByteSub	1197	0	✓	1197	0	✓	20.4	20.4	3.1

Interlays in Dual-Processor Systems

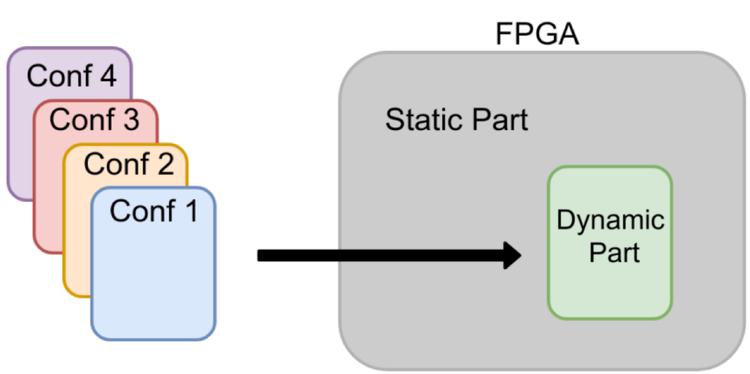
Multi-Processing



Vector Processing



Partial Reconfiguration



Prototype: Partially Run-Time Reconfigurable Shared RISC-V Processing System

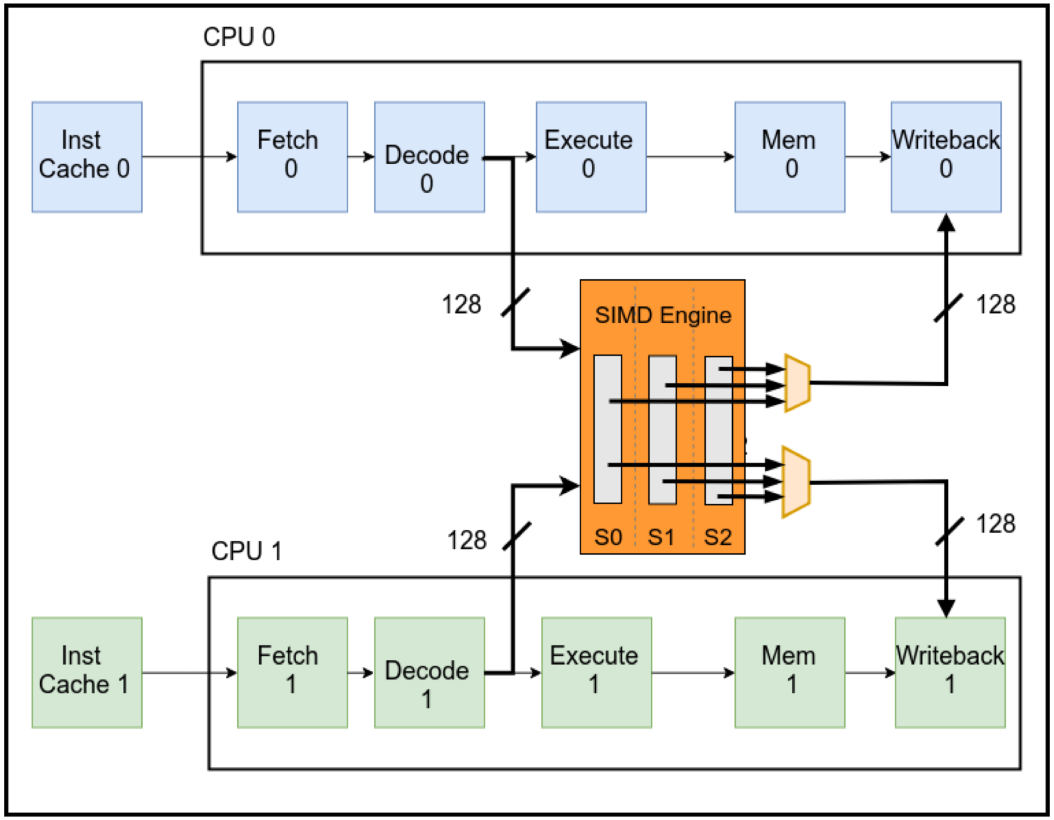
SIMD engine:

– 128-Bit SIMD ALU engine
– Placed logically inline with the scalar ALUs
– Shared amongst both CPUs

Additionally:

– Includes its own 16-entry Vector RF
– Shares the decoder unit with scalar CPUs
– Vector LD/ST are executed serially in 4 (32-bit) operations

Dual-Processor System with a Shared 128-bit SIMD Engine

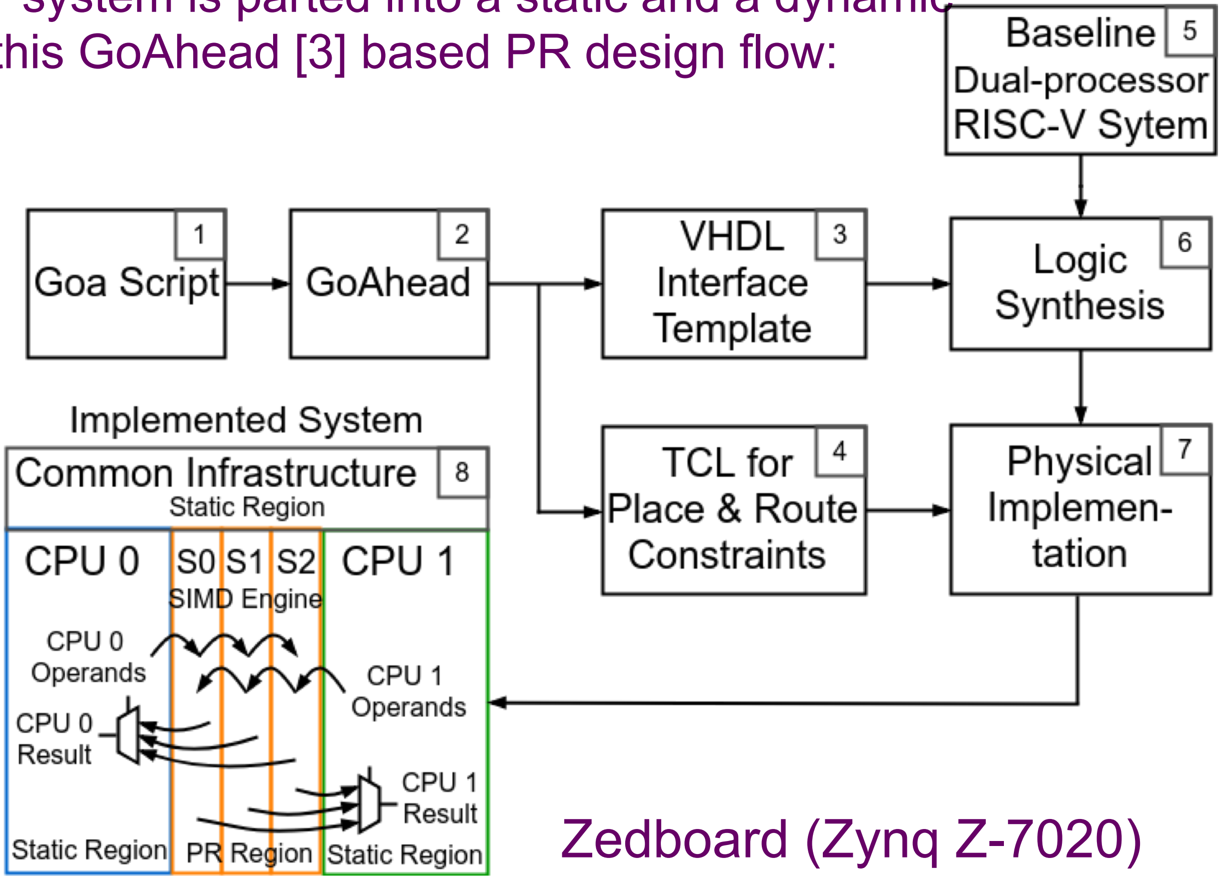


The University of Manchester

Advanced Processor Technologies Group

Prototype System Implementation

The dual-processor system is parted into a static and a dynamic subsystems using this GoAhead [3] based PR design flow:



Zedboard (Zynq Z-7020)

3. Beckhoff, Christian, Dirk Koch, and Jim Torresen. "Go ahead: A partial reconfiguration framework." *International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, IEEE, 2012.

MANCHESTER 1824
The University of Manchester

Advanced Processor Technologies Group

Static/Dynamic Interface Wiring Arrangement

The Interface:

- **Vertically aligned MSB (top)**

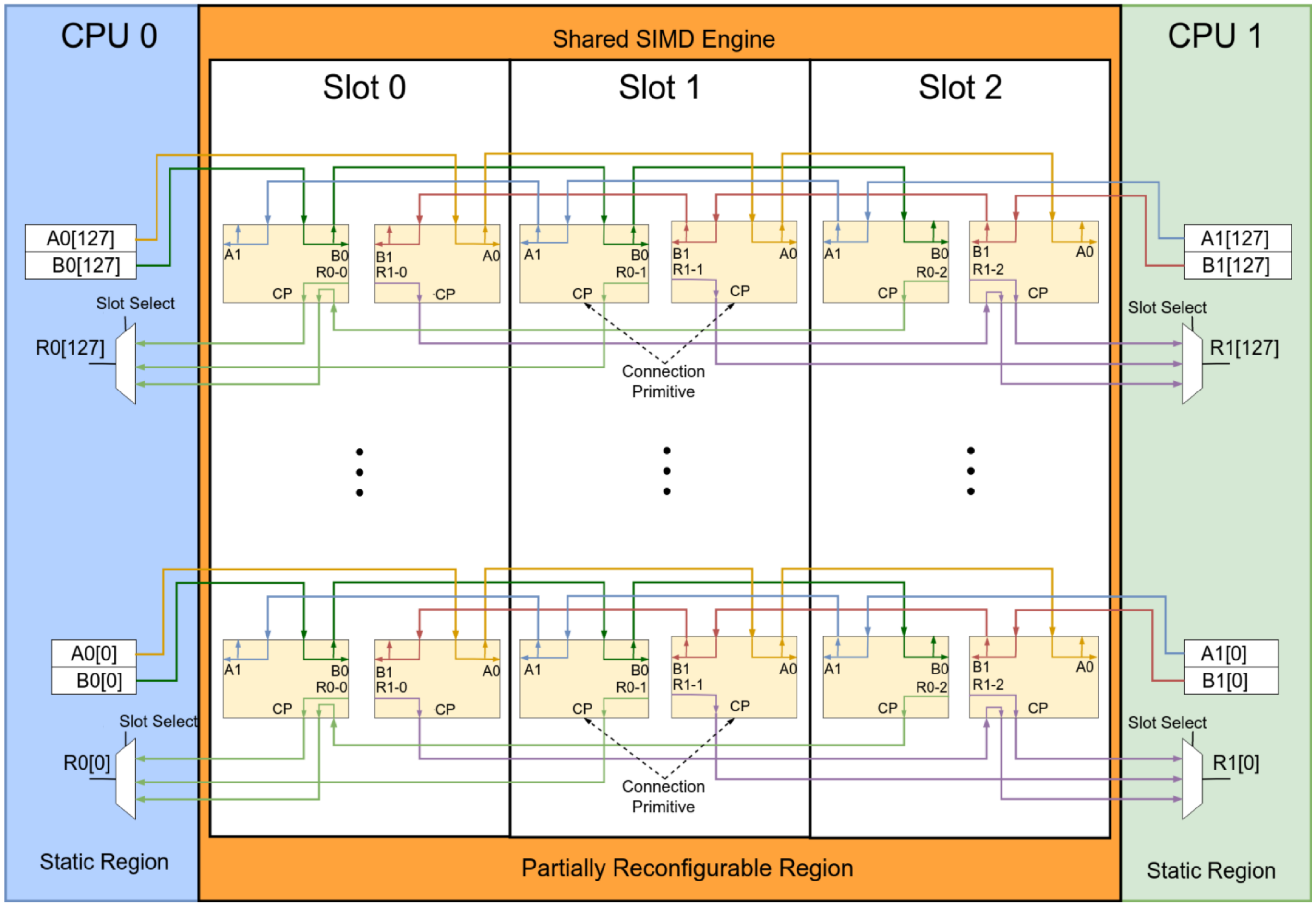
  ↑

  LSB (bottom)

- **Low overhead**

- **Remains consistent**

- **Signals can be split into 2, 4,8, 16 lanes**



Partially Run-Time Reconfigurable Subsystem

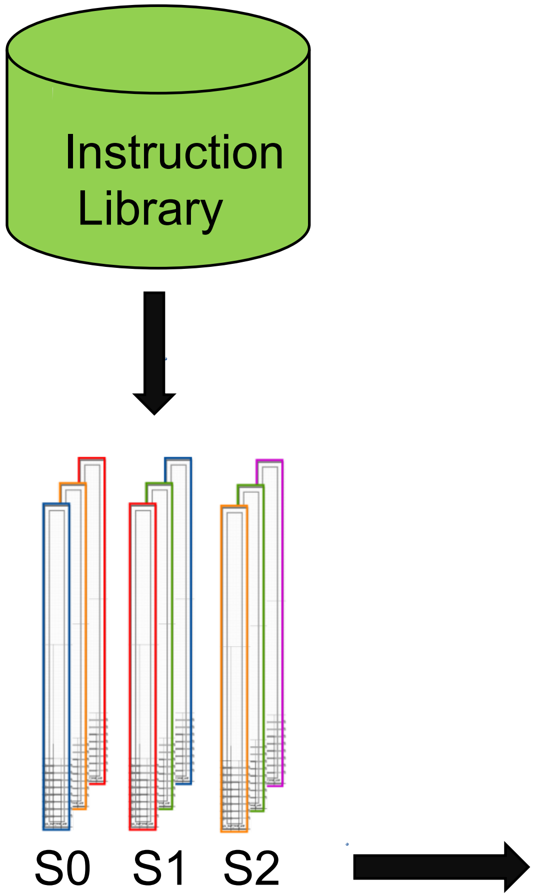
- PR region set to 2082 LUTs ➡ 694 LUTs per slot

- Each slot is 2 CLB columns wide

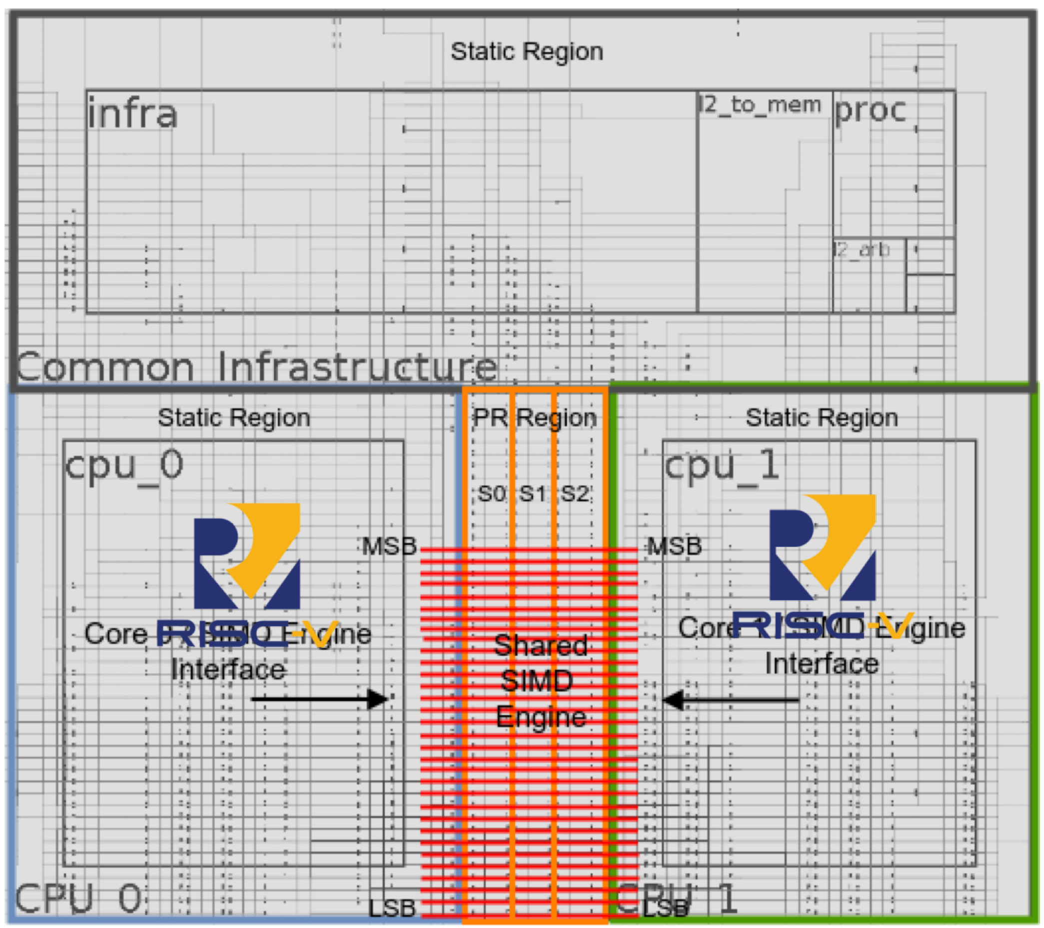
- PR region is reconfigured through the Zynq Internal Configuration Access Port (ICAP)
    - Considering ICAP maximum throughput (400 MB/s):
        ➡ Estimated slot reconfiguration time: 295 μs

- Module configuration prefetching can be used to hide reconf. Latency
    ➡ Prefetching instructions inserted in the code

System Implementation



System Floorplan

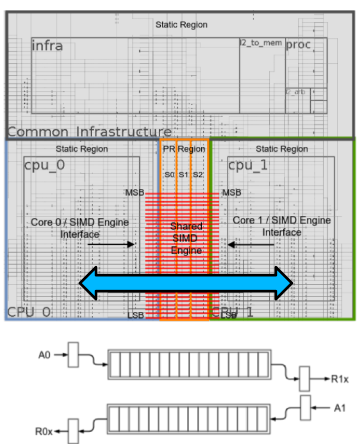


Zynq Z-7020 (Artix-7 FPGA Fabric)

Evaluation

Implemented Custom SIMD instructions

- Resource utilization and latency obtained from Vivado synthesis report
- Kernel speedup is calculated from assembly code analysis and execution traces generated from simulation

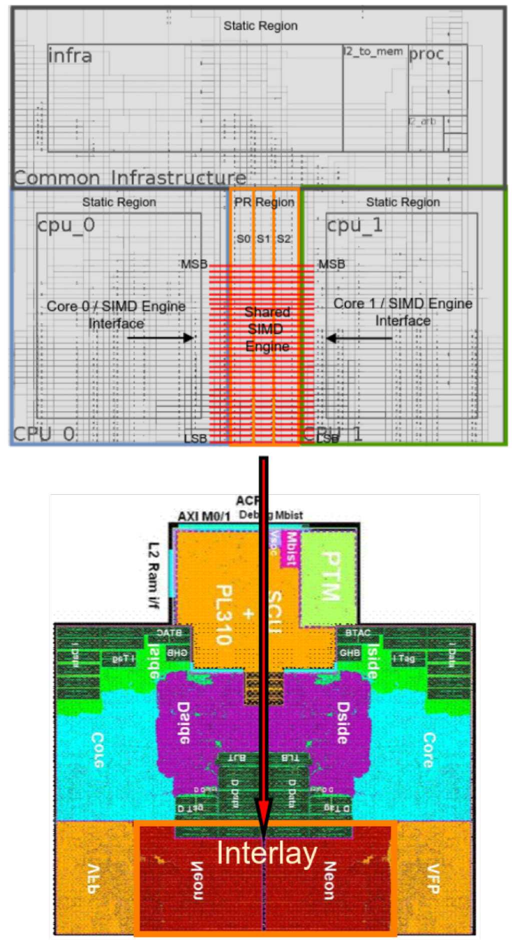


Mailbox instruction for core-to-core communication

Kernel Benhmark	Custom Vector Instruction	Resource Footprint	CI Latency	Folding Factor	# Instances per Slot	Kernel Speedup
MIX	VADD16	513 LUTs	5.7 ns	$\lambda = 1\times$	1	8×
ACS	VADD8	513 LUTs	5.7 ns	$\lambda = 1\times$	1	16×
	VSUB8	513 LUTs	5.7 ns	$\lambda = 1\times$	1	
	VSEL8	129 LUTs	4.5 ns	$\lambda = 1\times$	2	
SAD	VSAD8	1940 LUTs	14.2 ns	$\lambda = 4\times$	1 (Folded)	4×
SORT	VSORT8	312 LUTs	8.4 ns	$\lambda = 1\times$	2	448×
	VMBOX	128 LUTs	7.1 ns	$\lambda = 1\times$	2	

Systems Floorplan Comparison

Soft Dual Core RISC-V Prototype



ARM Cortex-A9 SoC

- Realistic approach
- Fits existing SoC architecture
- Enables further research

MANCHESTER 1824
The University of Manchester

APT
Advanced Processor Technologies Group

Conclusions

- **Interlays** can be a solution to provide enhanced flexibility and performance in SoCs
- **We described** implementation details (Interlay integration/management) and examples of CIs
- **With this work** we aim to stimulate research in the field of hybrid systems that can meet the challenges faced by existing SoC architectures

Future Work

- Customization of the Interlay fabric
  - Coarse/Fine grain mix, enhancing DSP blocks
- Enhancement of the emulation system
  - Scaling vector interface, number of Interlay slots
- Exploration of further application domains
  - Machine learning (customized precision)
- Building additional ecosystem components
  - Custom compilers


MANCHESTER
1824
The University of Manchester
APT
Advanced Processor Technologies Group
Thank You !
Questions?

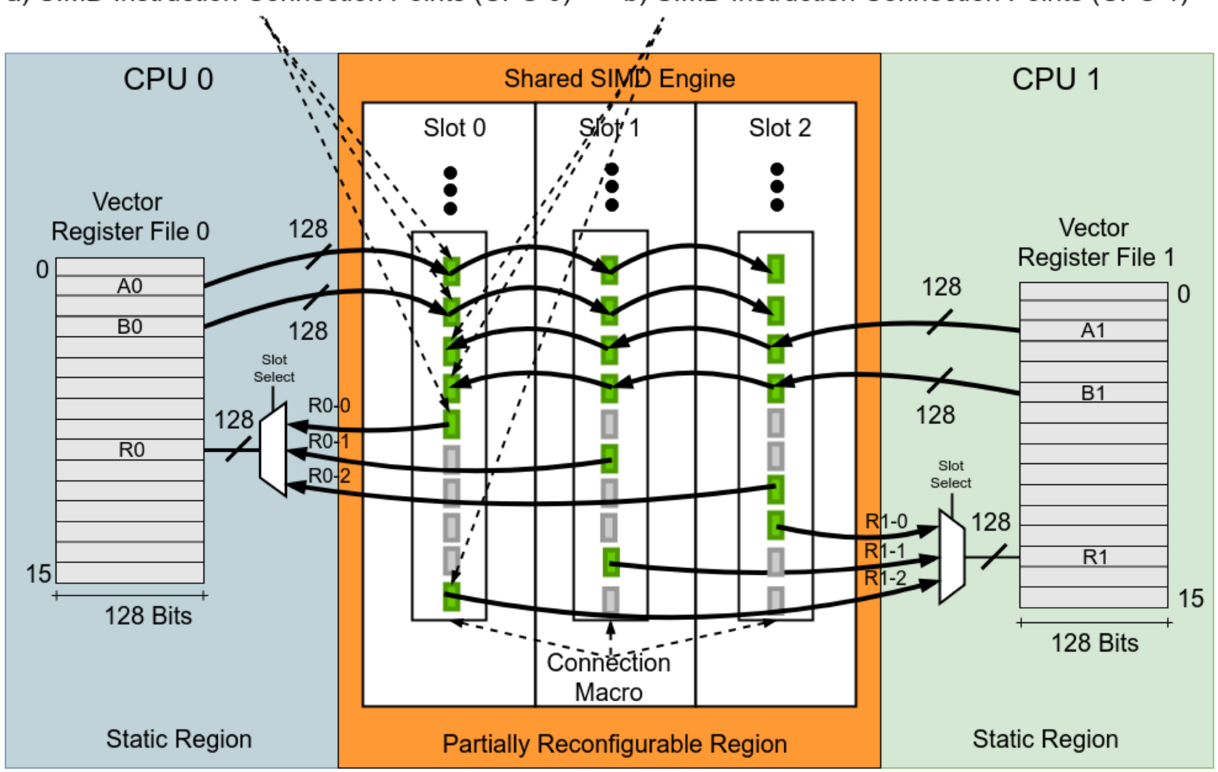
Backup

MANCHESTER
1824
The University of Manchester

Advanced Processor Technologies Group

Adding architectural support for the SIMD engine

Slot-Based Resource Sharing Infrastructure: Propagates operands from the CPUs to the slots and results from individual slots back to the CPUs



MANCHESTER 1824
The University of Manchester

APT
Advanced Processor Technologies Group

SIMD Engine Configuration Controller: Controls (re)configuration of the slots

- Used to time-share the SIMD engine amongst both CPUs
- Controls the integration of vector instructions at run-time
- Leverages vector compute and I/O bursts in each CPU

SIMD Instructions

- Identified offline through profiling
- Implemented only if it provides substantial kernel speedups

$$S_k = \frac{t_{kISA}}{t_{kSIMD}}$$

- And, it is invoked a significant number of consecutive times to amortize reconfiguration overhead

$$t_{SIMD} * n > t_{RCFG}$$

MANCHESTER 1824

The University of Manchester

Advanced Processor Technologies Group

Programmable Instruction Decoder: Enables changing instruction settings at run-time

- An extension of the Decode Unit
- Handles SIMD instruction information:
  1) Target slot
  2) SIMD instr. Cycles
  3) Operation folding factor
- These settings can be overwritten at run-time
- Setting configuration as an OS task

CPU 0		CPU 1	
SIMD Instr Encoding	Instruction Settings	SIMD Instr Encoding	Instruction Settings
0	Trap	0	Slot 0, 2 Cycles, λ = 1x
1	Slot 1, 1 Cycle, λ = 4x	1	Trap
2	Slot 2, 2 Cycles, λ = 2x	2	Slot 2, 2 Cycles, λ = 2x
3	Trap	3	Trap

MANCHESTER 1824
The University of Manchester

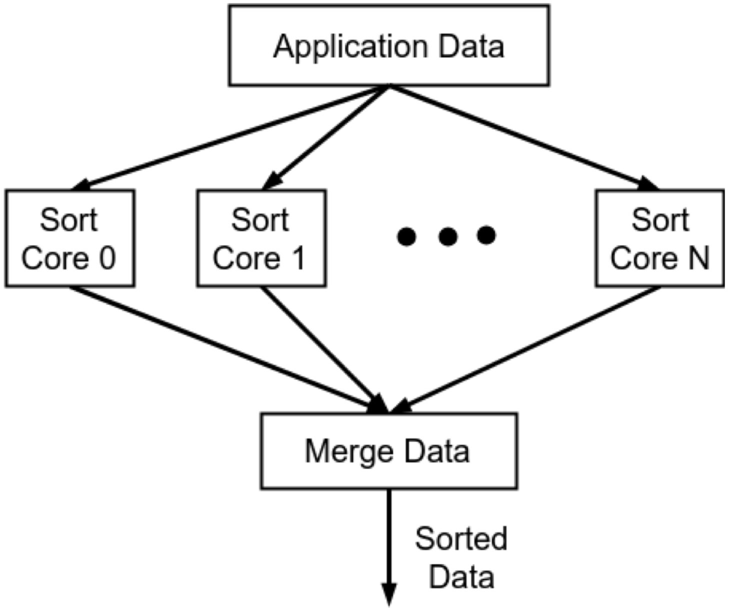
APT
Advanced Processor Technologies Group

Case Study

- Divided in 2 parts:

1) Each CPU executes its own program using its corresponding memory space
   - Kernel for mixing 16-bit PCM audio signals
   - Kernel for computing Sum of Absolute Differences values (motion estimation)
   - Kernel for computing 8-bit Add-Compare-Select values (Viterbi decoder)

2) Both CPUs work together to complete a task
   - Sorting an array of pixels by hue
   - Application data is split and equally allocated to each CPU data memory space
   - Each CPU sorts its corresponding sub-array
   - After sorting a final merge step combines both results



MANCHESTER 1824
The University of Manchester

APT
Advanced Processor Technologies Group

Static Subsystem

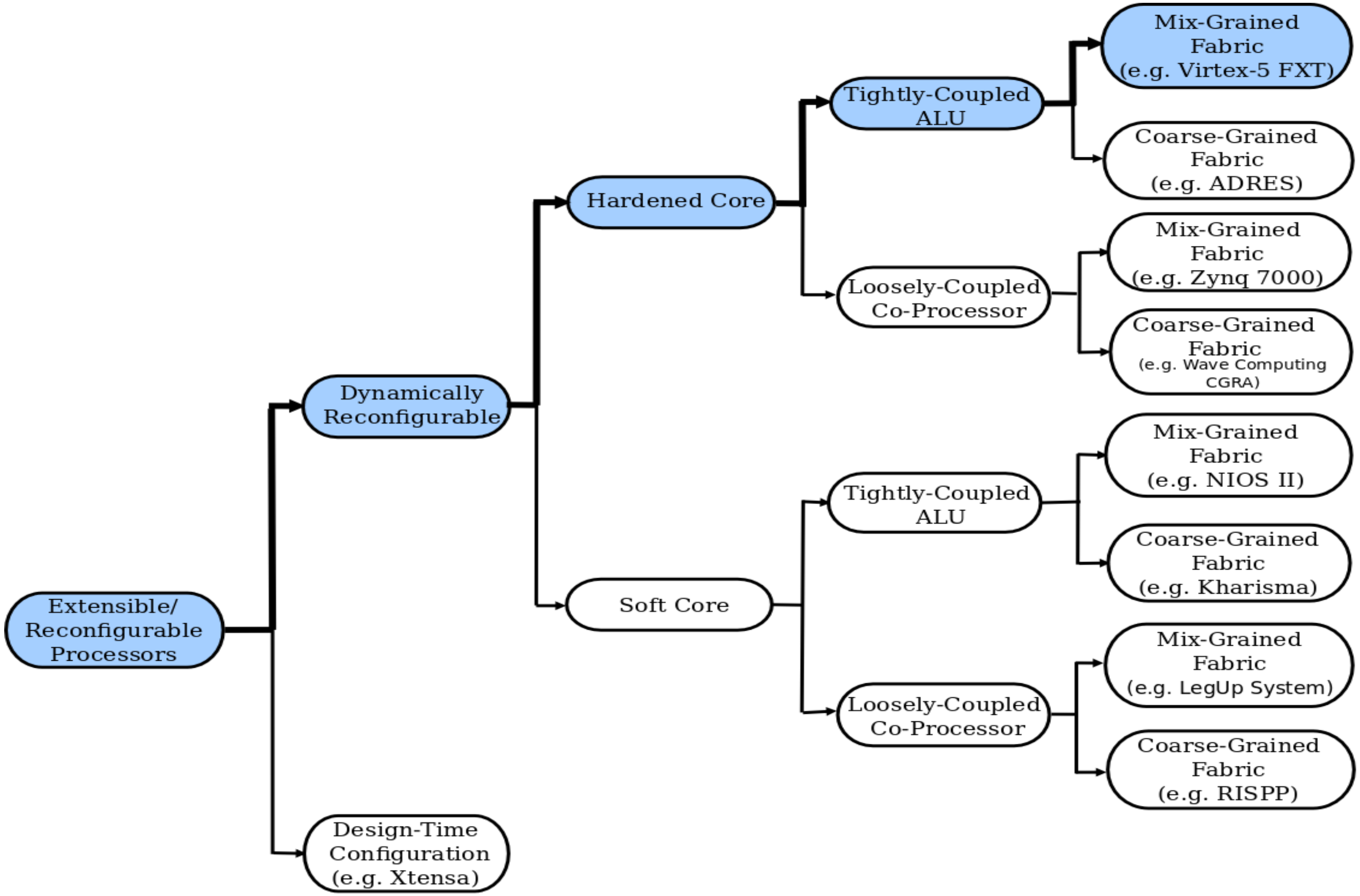
- Includes scalar CPUs, memory subsystem
- Operating frequency 96 MHz
- Component resource breakdown:

System Component	FPGA Primitive		
	LUT	DSP	BRAM
Soft Dual Processor RISC-V System	6917	8	36
RISC-V CPU 0	2728	4	10
RISC-V CPU 1	2728	4	10
Common Infrastructure	1461	0	16

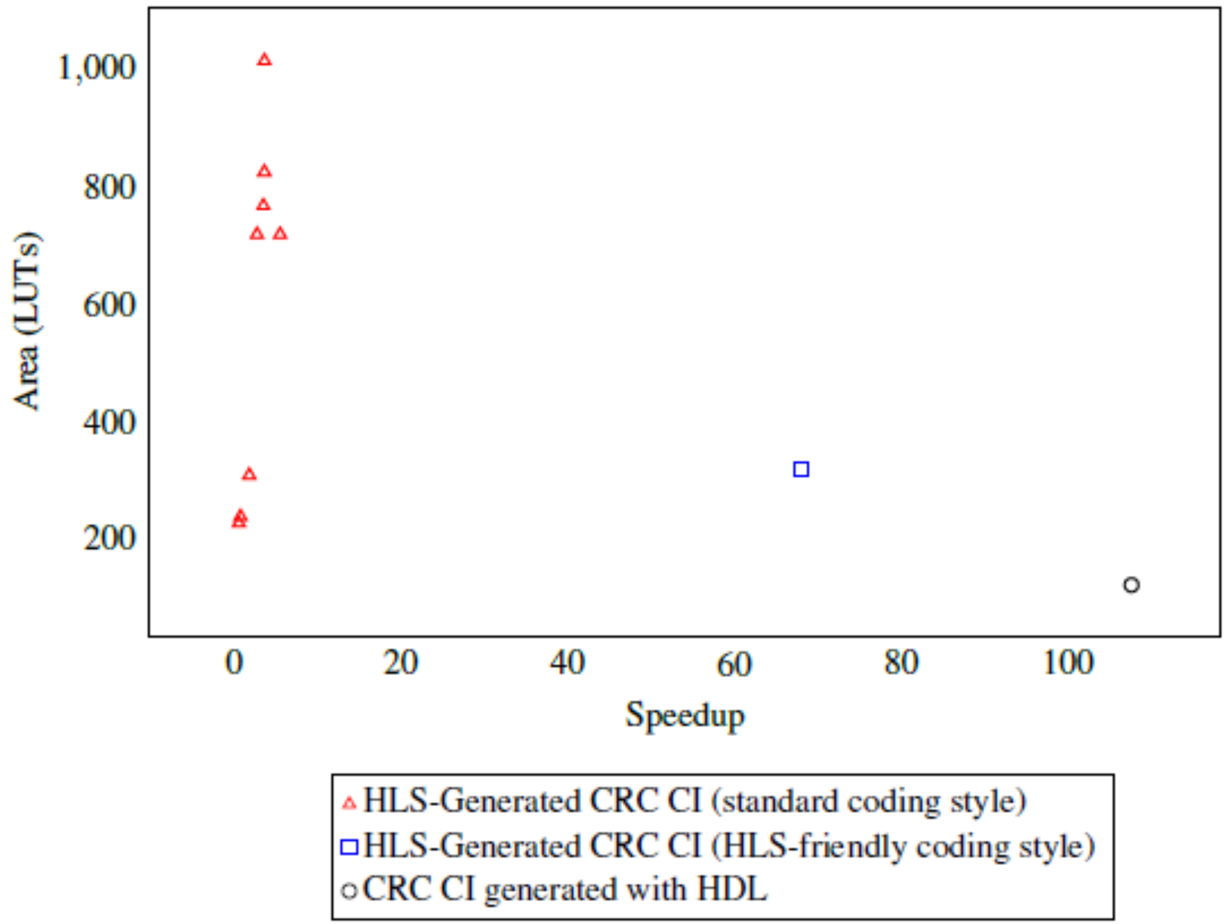
The University of Manchester

Advanced Processor Technologies Group

Classification of Reconfigurable Processor (RP) Architectures

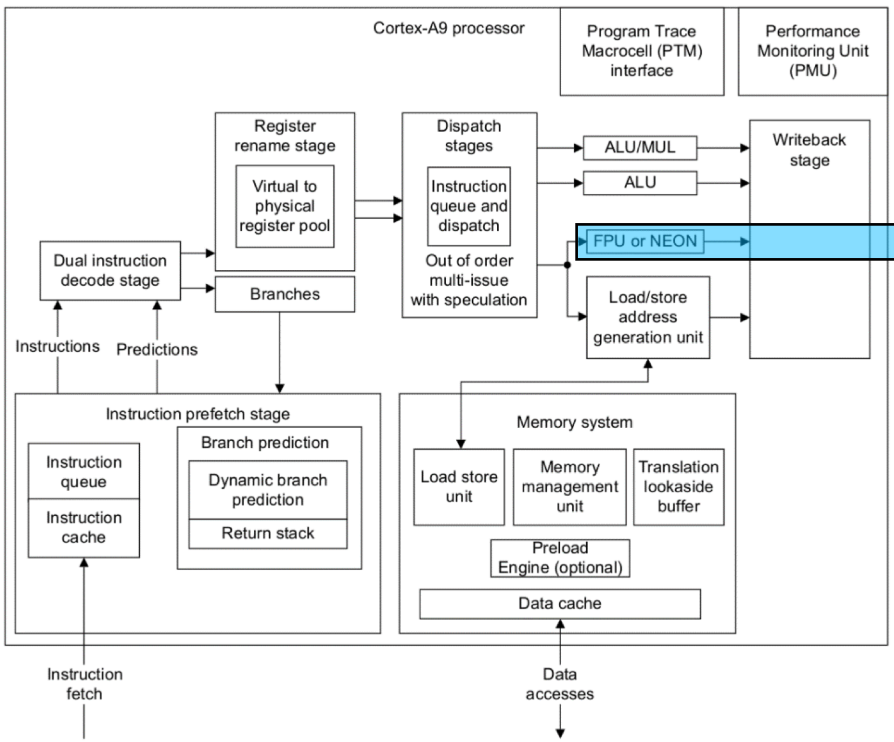


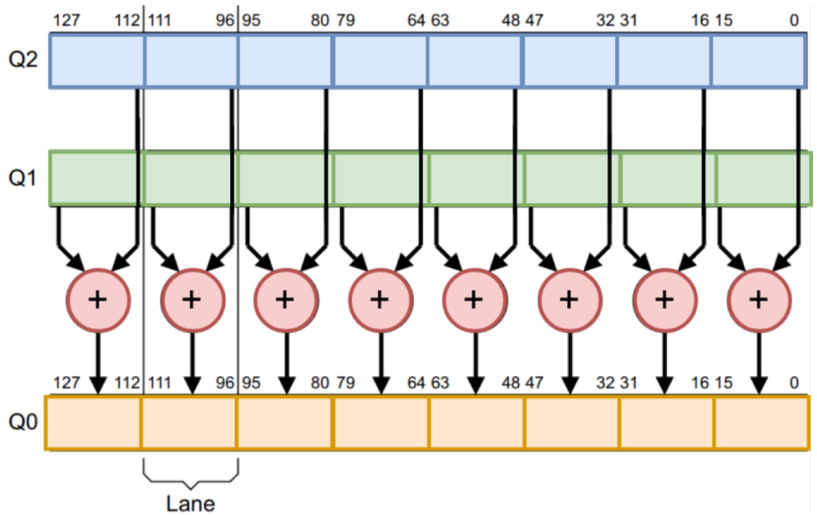
Leverage built-in custom data type libraries for developing HLS-friendly C code



Vector Accelerators

→ NEON





Exploits data-level parallelism

The University of Manchester

Advanced Processor Technologies Group

Custom Instructions for CPU Interlays

Design Aspects:

- Interface
- Area
- Energy
- Granularity

Instruction Set Extension Problem

Explored Approaches

Manual CI Generation	Automated CI Generation
Relatively slow development time	Enhanced design productivity
Full control over the generated RTL code	The generated RTL code can be influenced through C coding style
Most suitable for relatively small kernels	Suitable for relatively large functions
Hardware design expertise is required	A hardware design background is not necessarily required