Efficient Resource Shared RISC-V Multicore Processor

Md Ashraful Islam, Kenji Kise
Department of Computer Science
Tokyo Institute of Technology

Introduction

- Computation are the key requirement for the Edge computing to process the data near the source
- To meet the performance condition of such edge computing devices like IoT, multicore heterogeneous computing can be advantageous
- In FPGAs, the heterogeneous multicore is realized as multiple soft processor cores with custom processing elements
- In this work, we proposed the microarchitecture of a five-stage pipeline in-order scalar RISC-V processor that enables the sharing of functional units for instruction execution among the multiple cores
- We investigated the performance and hardware resource utilization for a four-core processor

What Is RISC-V

- RISC-V is the free and open-source Instruction Set Architecture (ISA) opposed to legacy ISA like x86, POWER, ARM etc.
- Rapidly growing software ecosystem for RISC-V making it more flexible for adoption and open the door for innovations in computer architecture research
- RISC-V adoption is in motion across industries such as AI, IoT, security, HPC, AR/VR, automotive, embedded, industrial, and more
- Base instruction set is RV32I (for 32-bit) and RV64I (for 64-bit) that contains only 47 instructions. Other type of instructions can be added as extensions like
  - M for multiply and divide instructions
  - A for Atomic instructions
  - F for floating point instructions
  - …..

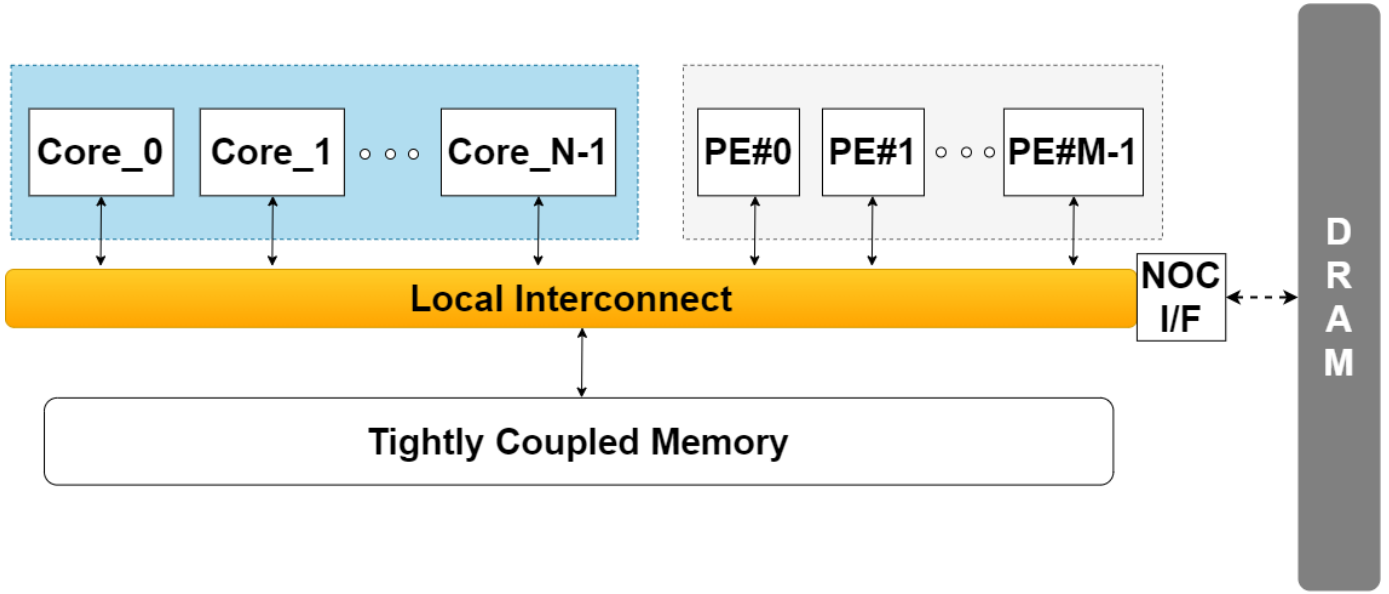
*Source: https://riscv.org/about/

FPGA Soft Processor Challenge

- FPGA is most frequently used to implement heterogeneous system when the number of devices are not good enough to adapt the expensive ASIC flow.
- Now a days several FPGAs are available with hard processors which contains 1, 2 or 4 processor cores. These hard processors have higher operating frequencies but have few drawbacks-
  - Number of processor core are fixed, which may be sub-optimal
  - Processor's configurations are fixed like cache, memory regions etc.
  - Each hard processor is placed in a fixed position, which may lead to difficulties in routing
- FPGA soft processors provide the flexibility in terms of number of cores and configuration which uses FPGA logic.
- Very limited logic resources are available for soft processor implementation given the fact accelerators utilize much logic resource.

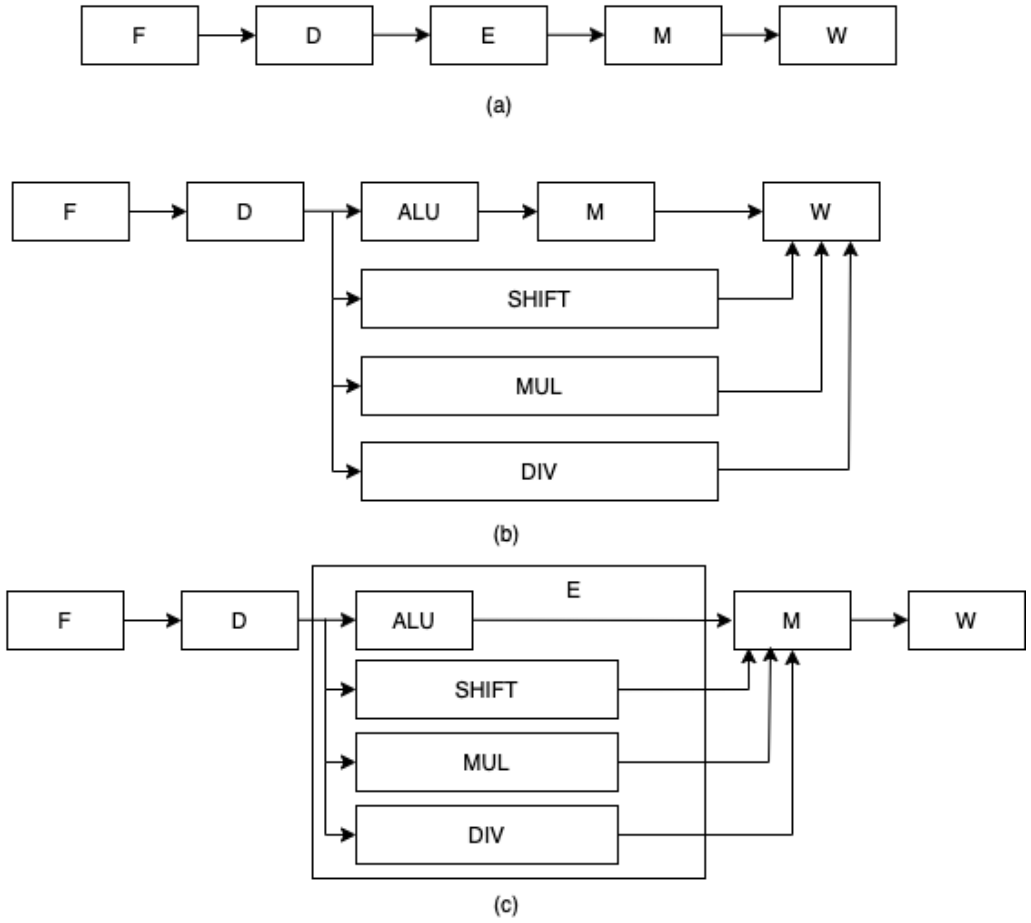
Heterogeneous Multicore Architecture

- Heterogeneous multicore incorporates-
  - N number of RISC-V cores
  - M number of custom hardware Processing Element (PE)
  - Tightly Coupled Memory (TCM) are used to avoid the complexity of cache coherency
  - Local interconnect to facilitate communication among cores, PE, and TCM
  - Network-on-Chip Interface (NOC I/F) to extend such multicores

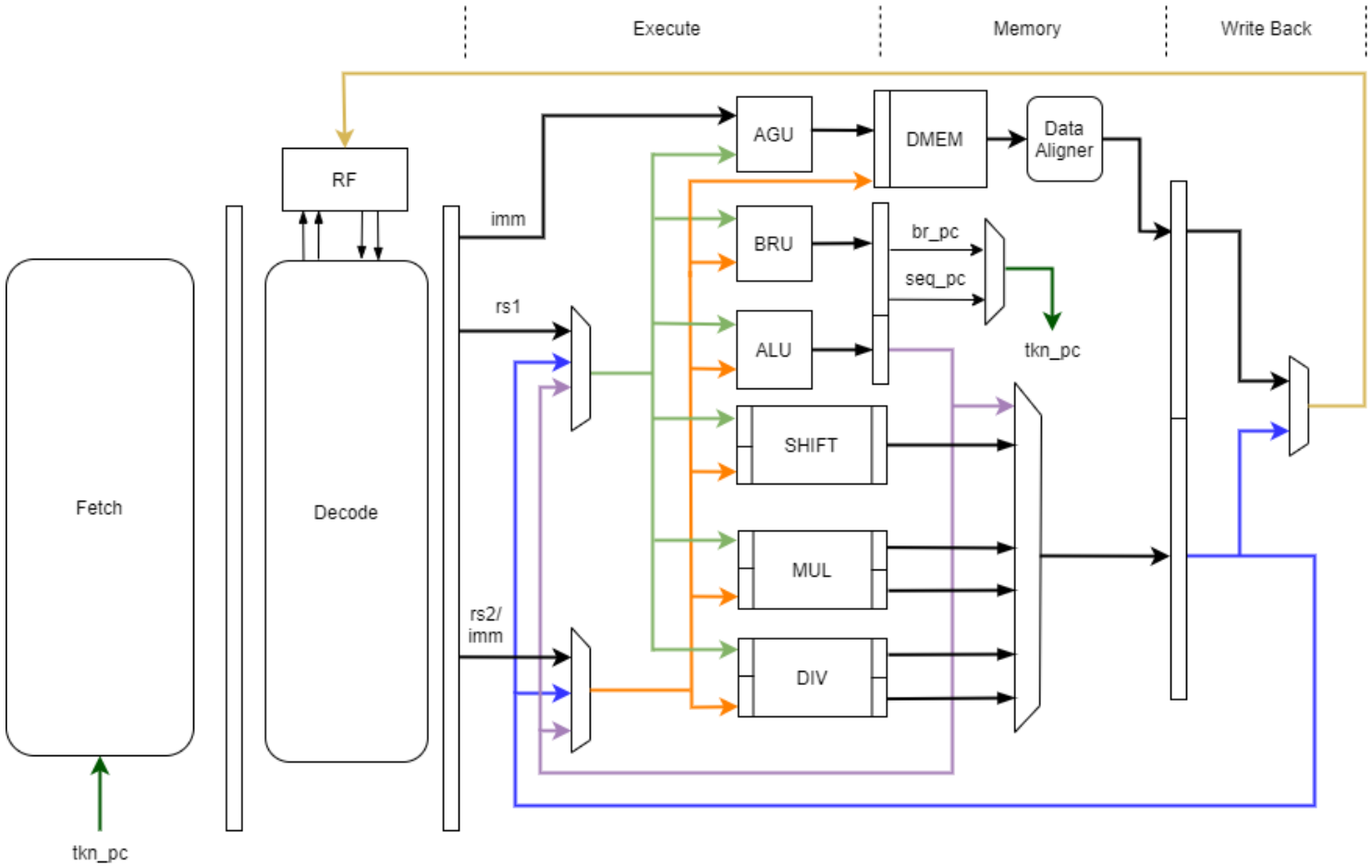


Pipeline Configuration

- (a) Simple 5-stage pipeline configuration where each stage of pipeline have fixed cycle latency
- (b) VexRiscv is an optimized RISC-V core implementation for FPGA soft processor. It's 5-stage pipeline is optimized for higher achievable operating frequency.
- (c) RVCorep-32IM is proposed 5-stage processor microarchitecture which achieves better performance than VexRiscv



RVCoreP-32IM Microarchitecture



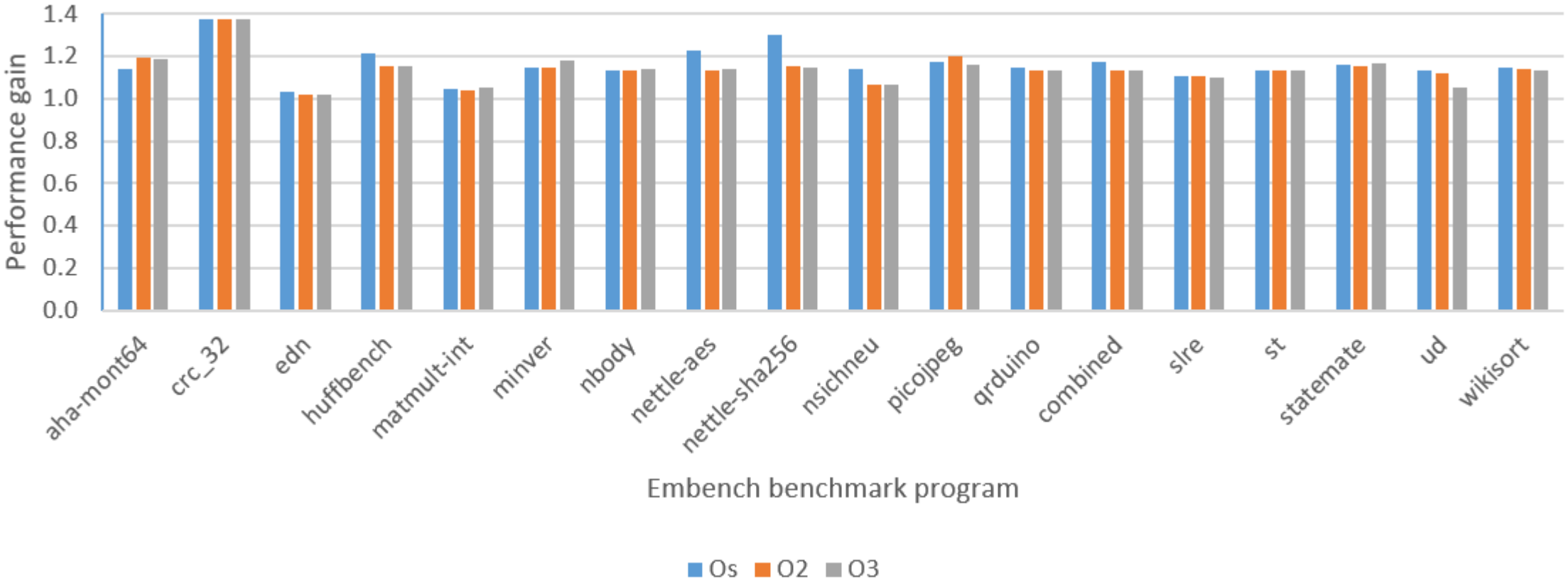
RVCorep-32IM and VexRiscv FPGA Implementation

- We evaluate the operating frequency and hardware resource utilization for Nexys A7 board containing Xilinx Artix-7 FPGA

Core	Frequency (MHz)	LUT	Register	BRAM
RVCoreP-32IM	164	1,556	860	1.0
VexRiscv	143	1,428	1,015	2.5

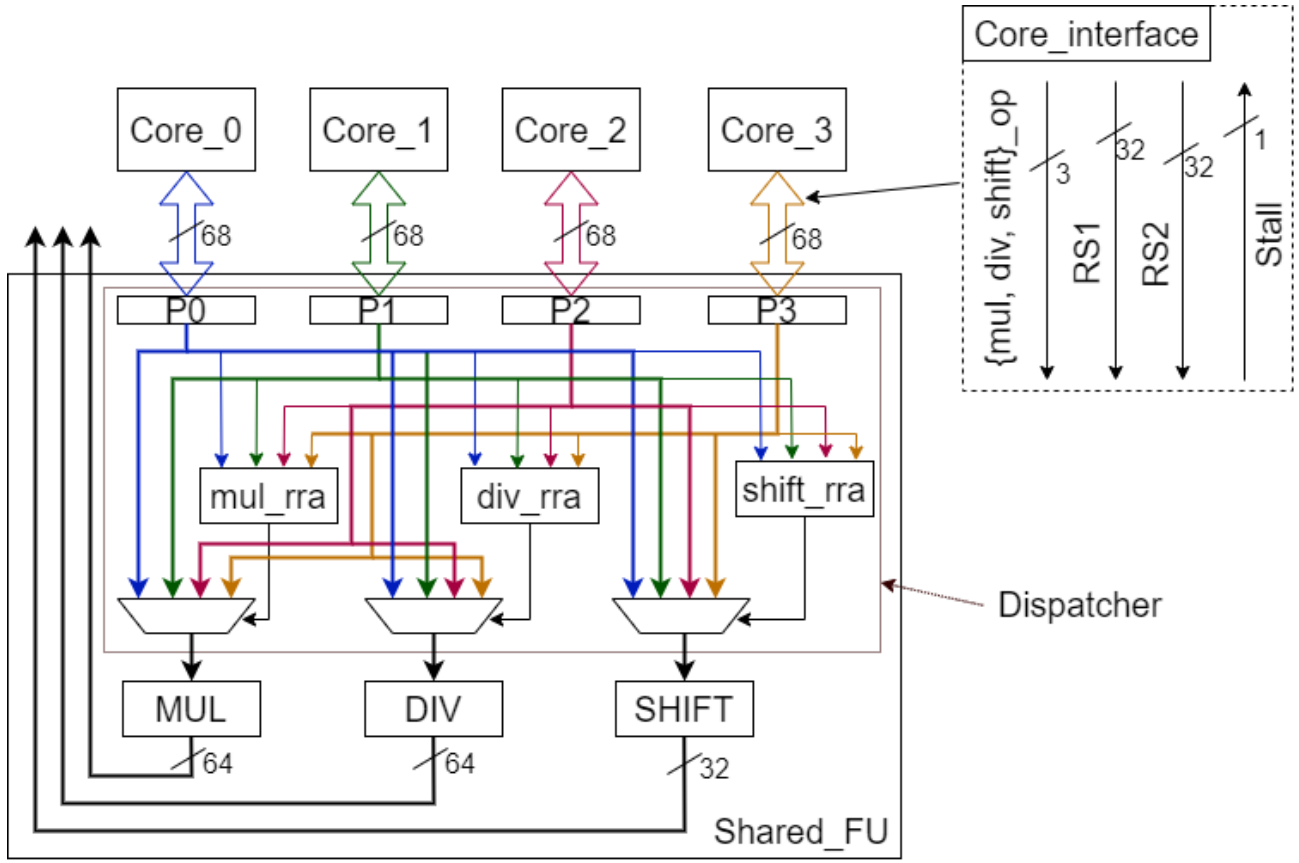
RVCorep-32IM and VexRiscv Benchmark

- We perform the simulation using Embench benchmark programs to compare execution time of the same program in RVCoreP-32IM and VexRiscv.
- Figure shows the normalized performance of RVCoreP-32IM compared to VexRiscv



Sharing functional units in multicore RVCoreP-32IM

- The multiplication, division, and shift operation require a considerable amount of logic resources compared to the ALU. Those units are less frequently used compared to ALU
- We shared the multiplication, division, and shift units among the cores



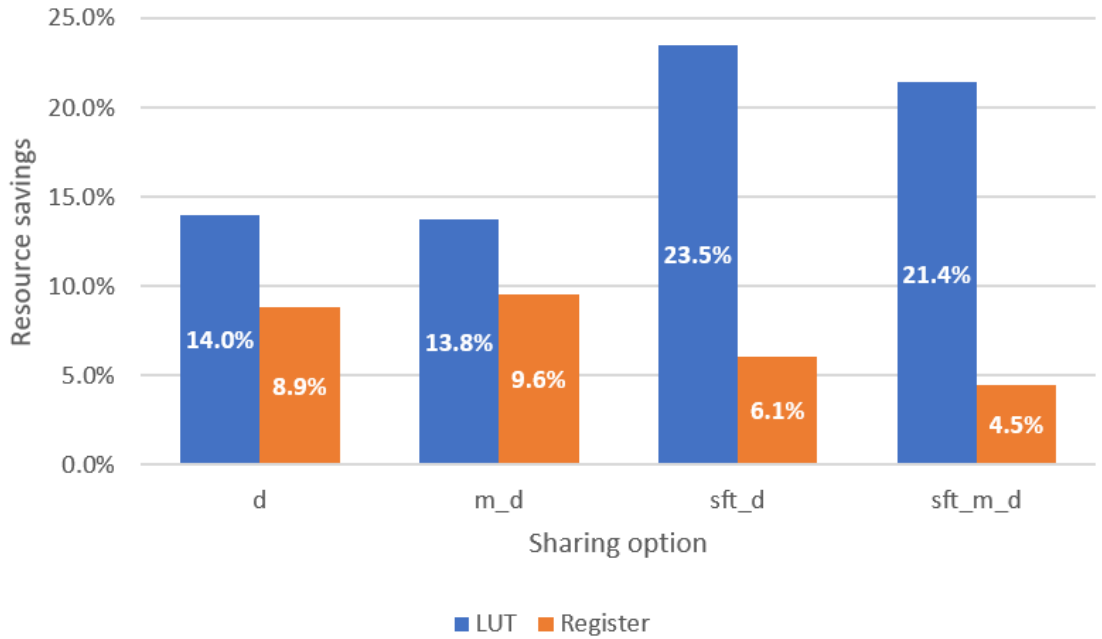
Sharing configurations in RVCoreP-32IM

- RVCopreP-32IM have the following sharing options-
  - No sharing
  - Sharing shifter, multiplier and divider (sft_m_d)
  - Sharing shifter and divider (sft_d)
  - Sharing multiplier and divider (m_d)
  - Sharing divider (d)
- We experimented with 4-core to investigate the FPGA logic utilization and performance impact

FPGA Resource Utilization of Multicore

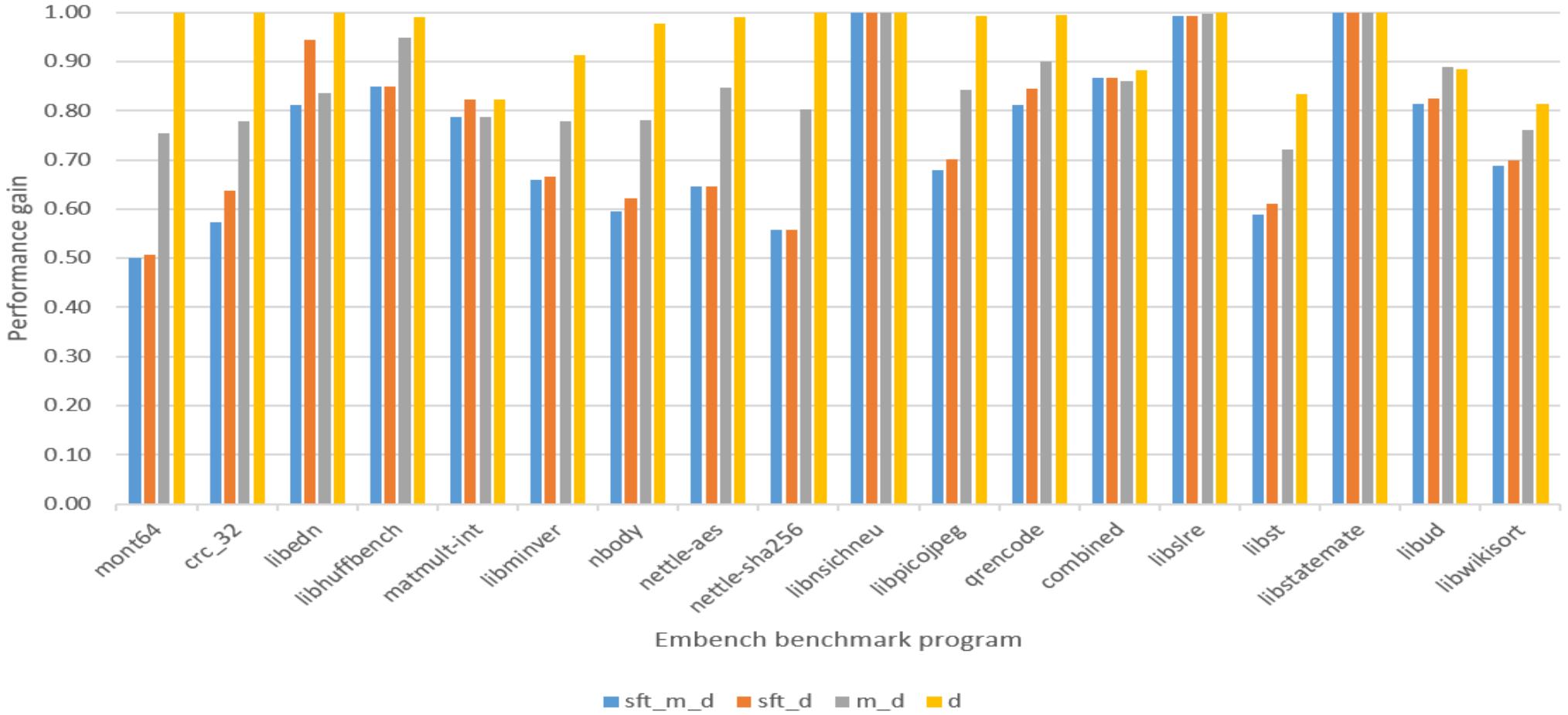
- FPGA implementation result of four RVCoreP-32IM with different sharing configurations are shown below-

Resource	No sharing	sft_m_d	sft_d	m_d	d
LUT	6,084	4,781	4,653	5,245	5,231
Register	3,994	3,814	3,751	3,612	3,640
DSP	16	4	16	4	16



Performance Impact of Sharing Logic

- The same Embench program was used to investigate the performance impact of sharing the functional units among the four cores.



Conclusion & Future Work

- FPGA logic utilization is an important design factor because of fixed and constraint resources
- We studied the resource utilization implication for sharing functional units like the shifter, multiplier, and divider in multicore (4-core)
- Sharing allows to reduce the DSP usage to 75%, LUT up to 23.5%, and register up to 9.6%.
- All kinds of workload are not affected by the resource sharing. Since FPGA is a reprogrammable device, user can choose the sharing options that he provides the best match for resource versus performance
- We are studying the 4-stage and 6-stage pipeline processor to find the resource and performance effect as well as adding the custom PE

Tokyo Tech

Thank You