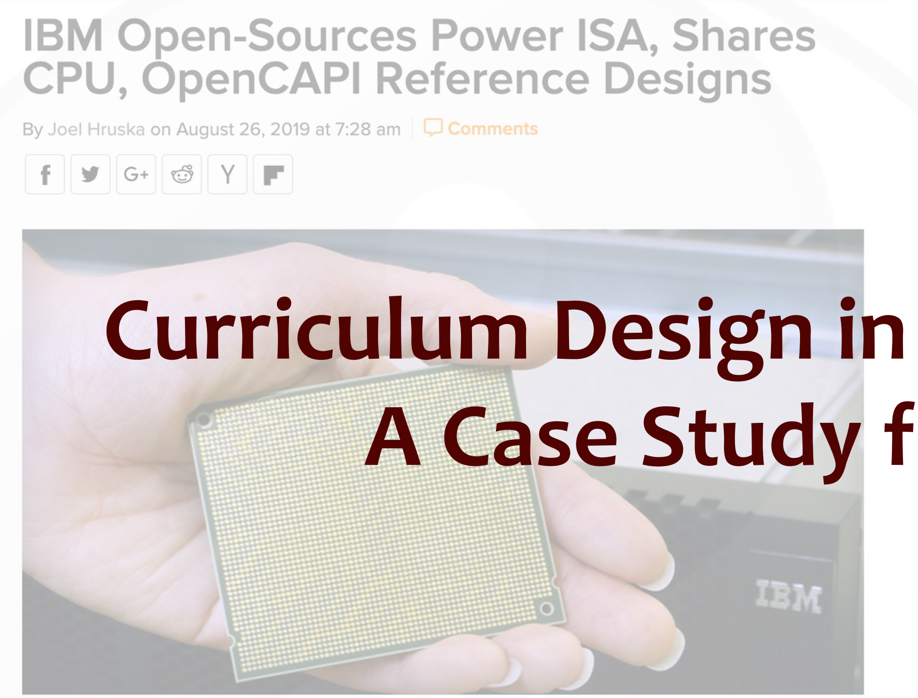
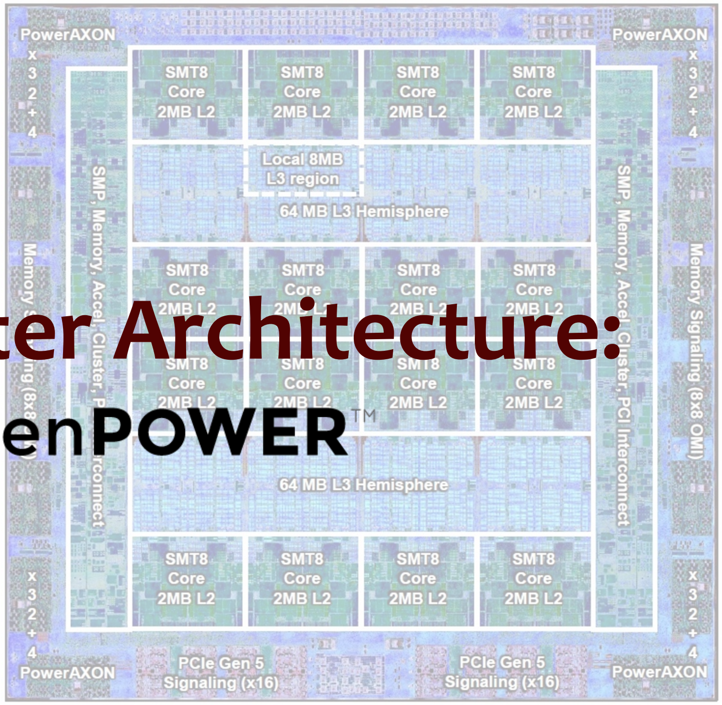


IBM Open-Sources Power ISA, Shares CPU, OpenCAPI Reference Designs

By Joel Hruska on August 26, 2019 at 7:28 am | Comments



Curriculum Design in Computer Architecture: A Case Study for OpenPOWER™

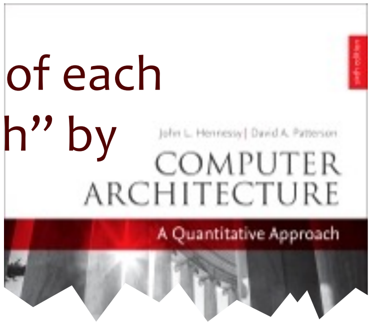
Dr. Wu Feng, Professor of CS and ECE at Virginia Tech (VT)
wfeng@vt.edu

Atharva Gondhalekar, PhD Student in ECE at VT

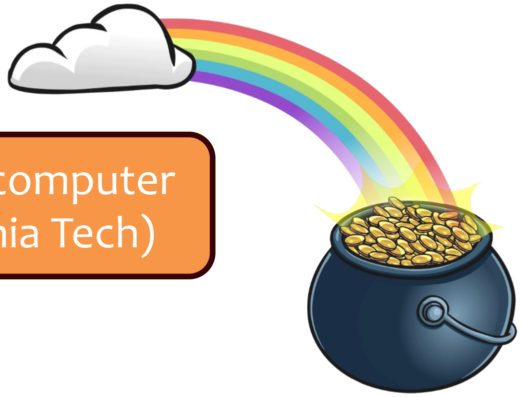
VIRGINIA TECH™

SyNeRGy
synergy.cs.vt.edu

Project Overview



- Goal
  - Deliver a curriculum on the POWER architecture for the end of each chapter of "Computer Architecture: A Quantitative Approach" by Hennessy and Patterson (a semester-long course)
- Proposed Tasks
  - Design supplemental course material about the POWER architecture for a senior-level undergrad course in computer organization (e.g., *CS/ECE 4504: Computer Organization* at Virginia Tech)
    - Fundamentals of Quantitative Design and Analysis (Chapter 1)
    - Memory Hierarchy (Chapter 2)
    - Instruction-Level Parallelism (Chapter 3)
    - Data-Level Parallelism (Chapter 4)
    - Thread-Level Parallelism (Chapter 5)
- Focus of this Overview? Exercises for CS/ECE 4504



Bonus: POWER-based projects for a graduate-level course in computer architecture (e.g., *CS/ECE 5504: Computer Architecture* at Virginia Tech)

Chapter 2: Memory Hierarchy

- Memory Hierarchy in POWER CPUs
  - gem5-based exercises
    - Evaluation* *with* and *without* caches
    - Characterization of the impact of varying memory-access patterns
    - Evaluation of tradeoffs with respect to cache associativity, cache size, and memory technology
    - Exercises available via github (and eventually, the OpenPOWER Foundation)

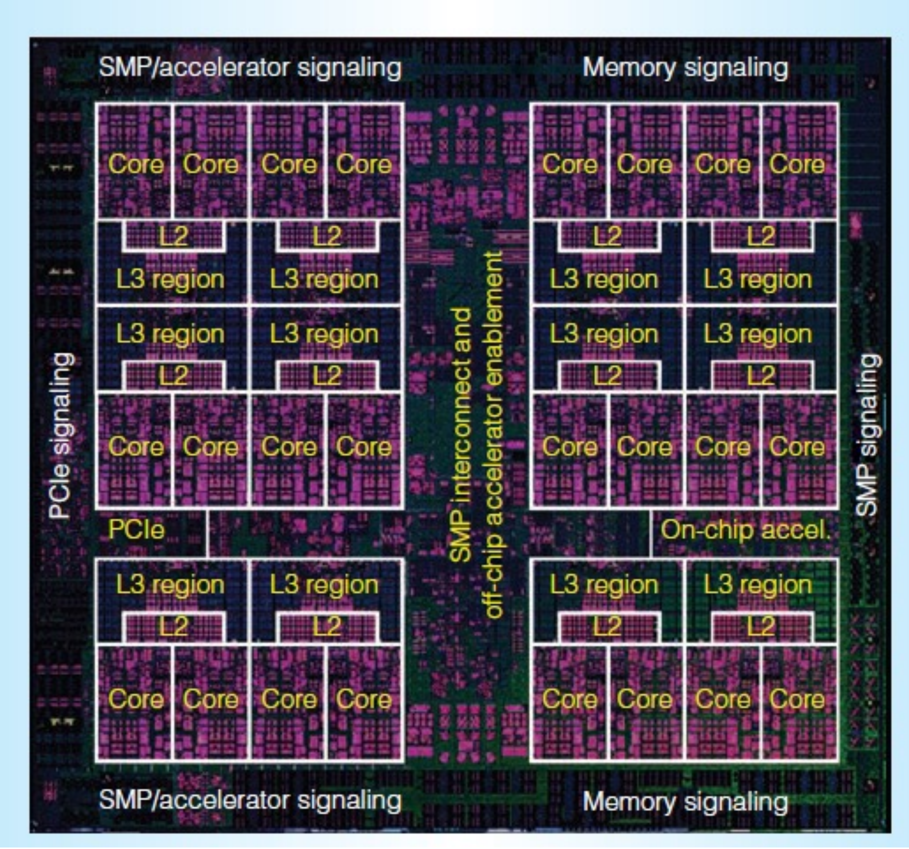
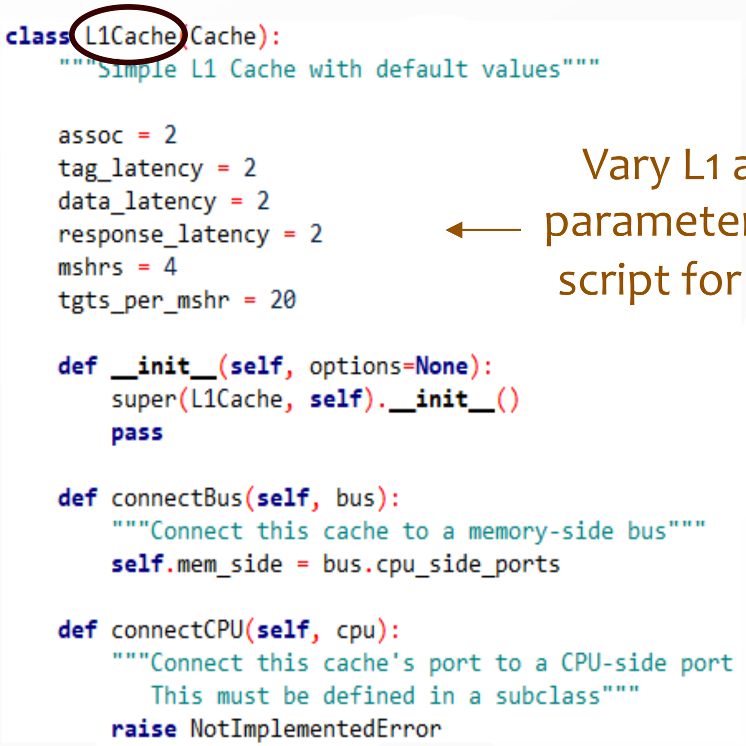

SMP/accelerator signaling
Memory signaling
Core
L2
L3 region
PCIe signaling
SMP interconnect and off-chip accelerator enablement
SMP signaling
PCIe
On-chip accel.
SMP/accelerator signaling
Memory signaling

Image source: [IBM Power9 Processor Architecture]()

* Evaluation can be with respect to performance, power, energy efficiency, resilience, productivity, and so on..

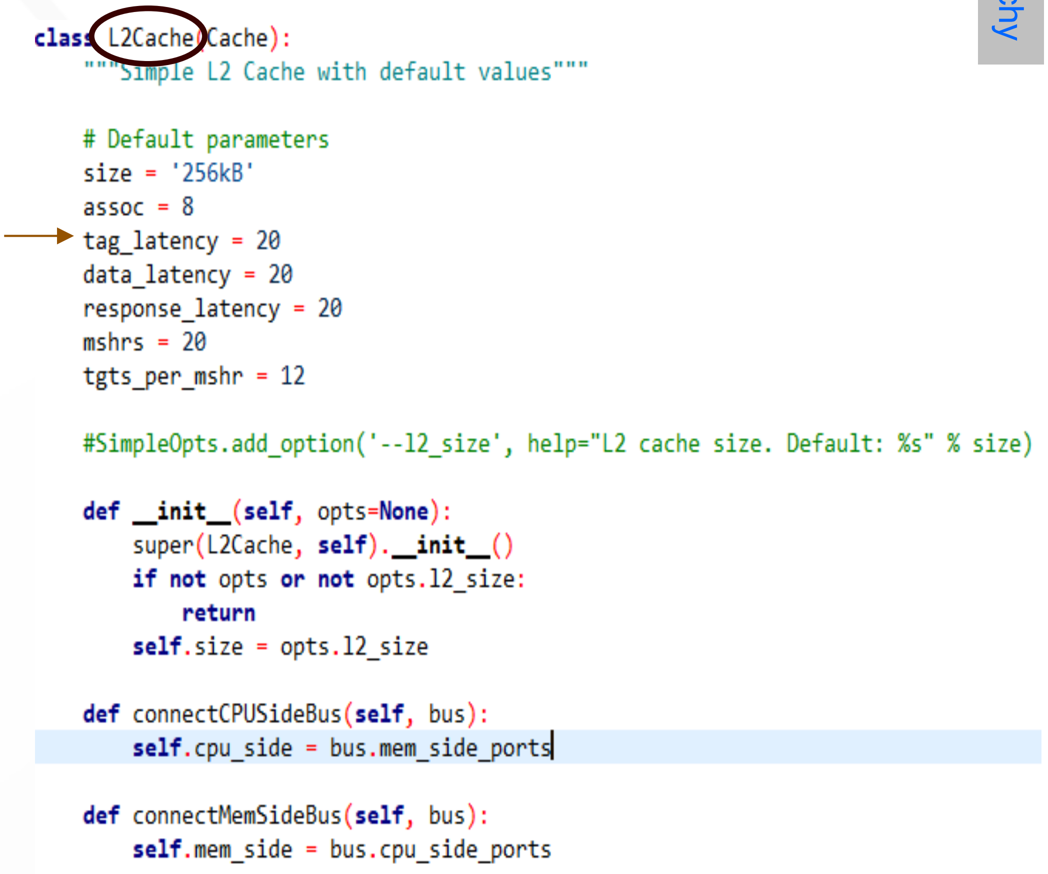
Example Exercise for Memory Hierarchy:

Varying Cache Parameters in gem5



```
class L1Cache(Cache):
    """Simple L1 Cache with default values"""

    assoc = 2
    tag_latency = 2
    data_latency = 2
    response_latency = 2
    mshrs = 4
    tgts_per_mshr = 20

    def __init__(self, options=None):
        super(L1Cache, self).__init__()
        pass

    def connectBus(self, bus):
        """Connect this cache to a memory-side bus"""
        self.mem_side = bus.cpu_side_ports

    def connectCPU(self, cpu):
        """Connect this cache's port to a CPU-side port
           This must be defined in a subclass"""
        raise NotImplementedError
```

← Vary L1 and L2 parameters in the script for caches →

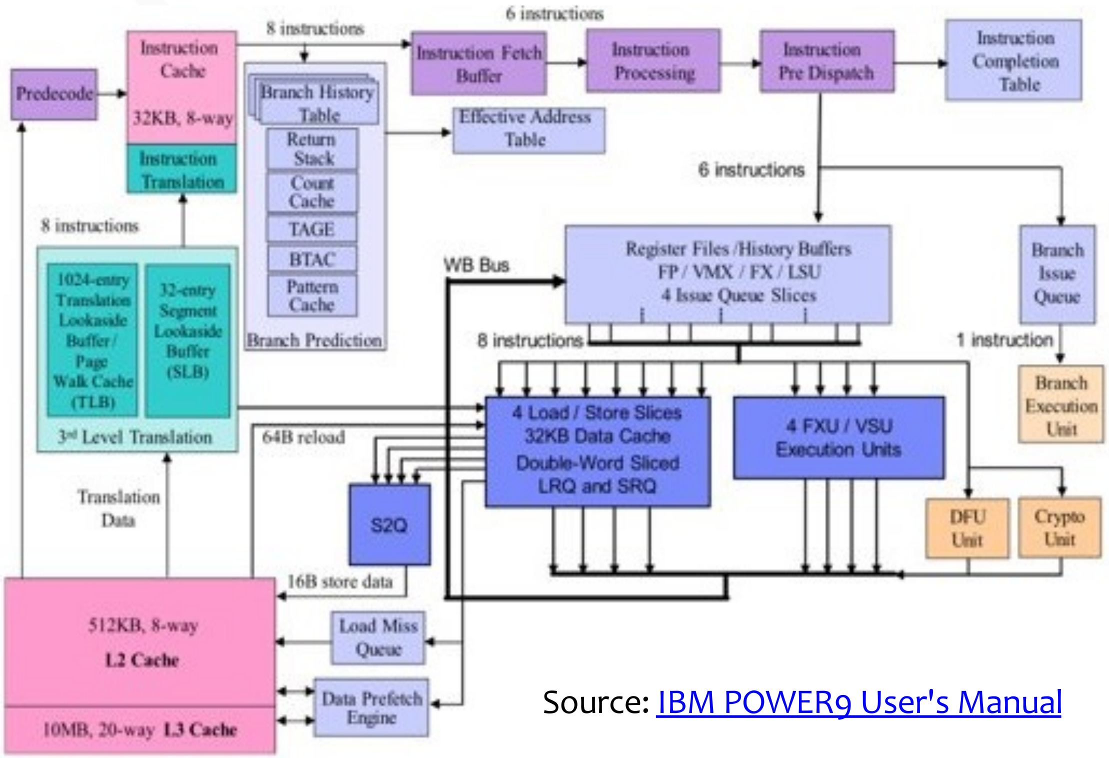


```
class L2Cache(Cache):
    """Simple L2 Cache with default values"""

    # Default parameters
    size = '256kB'
    assoc = 8
    tag_latency = 20
    data_latency = 20
    response_latency = 20
    mshrs = 20
    tgts_per_mshr = 12

    #SimpleOpts.add_option('--l2_size', help="L2 cache size. Default: %s" % size)

    def __init__(self, opts=None):
        super(L2Cache, self).__init__()
        if not opts or not opts.l2_size:
            return
        self.size = opts.l2_size

    def connectCPUSideBus(self, bus):
        self.cpu_side = bus.mem_side_ports

    def connectMemSideBus(self, bus):
        self.mem_side = bus.cpu_side_ports
```

L1 Associativity	Matrix Multiplication Runtime (ms)
2	324
4	311
8	305

Chapter 3: Instruction-Level Parallelism (ILP)

- ILP in POWER CPUs
  - Curriculum for POWER
    - Tomasulo’s Algorithm
    - Case Study: Out-of-order execution in POWER9
  - gem5-based exercises
    - Evaluation of branch prediction policies using a matrix multiplication (integer) workload
    - Exercises available via github (and eventually, the OpenPOWER Foundation)



Source: IBM POWER9 User's Manual

Example Exercise for ILP:
Branch Prediction in gem5

- Evaluate the performance of *branch prediction policies* in gem5
  - 2-bit local branch predictor
  - Tournament predictor
  - TAGE (default in POWER9)
  - None

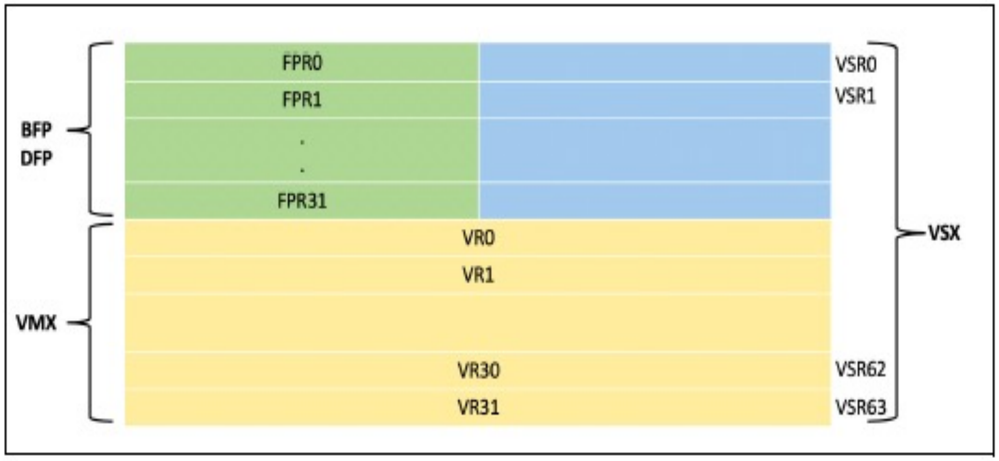
Branch prediction evaluation for matrix multiplication

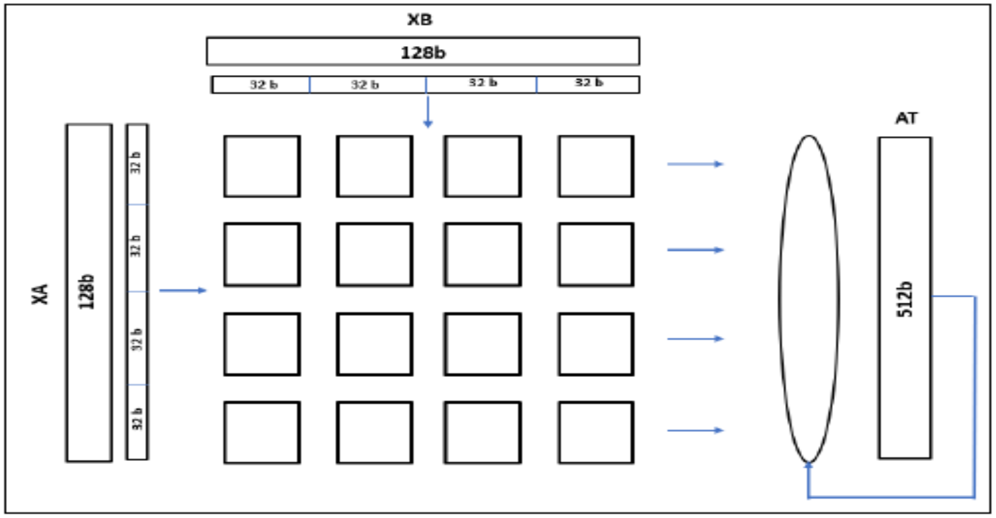
BRANCH PREDICTOR (BP)	# BP LOOKUPS	# INCORRECT PREDICTIONS	RUNTIME (MS)
2-bit local	296421	6054	931
Tournament	296421	5864	931
TAGE	296421	1903	930
None	N/A	N/A	936

Caveat: The above evaluation uses an *in-order* execution/completion backend (SimpleCPU) instead of *out-of-order* execution/completion backend (O3CPU). Why? O3CPU is *not* implemented for POWER in gem5.

Chapter 4: Data-Level Parallelism (DLP)

- DLP in POWER CPUs
  - Curriculum for POWER
    - Intro to vector built-in functions & vector-scalar extensions (VSX)
  - Exercises based on POWER9 / POWER10 Functional Simulator
    - Matrix multiplication
      - Using vector-scalar extensions (VSX)
      - Matrix Multiply Assist (MMA) architecture
    - Exercises available via github (and eventually, the OpenPOWER Foundation)





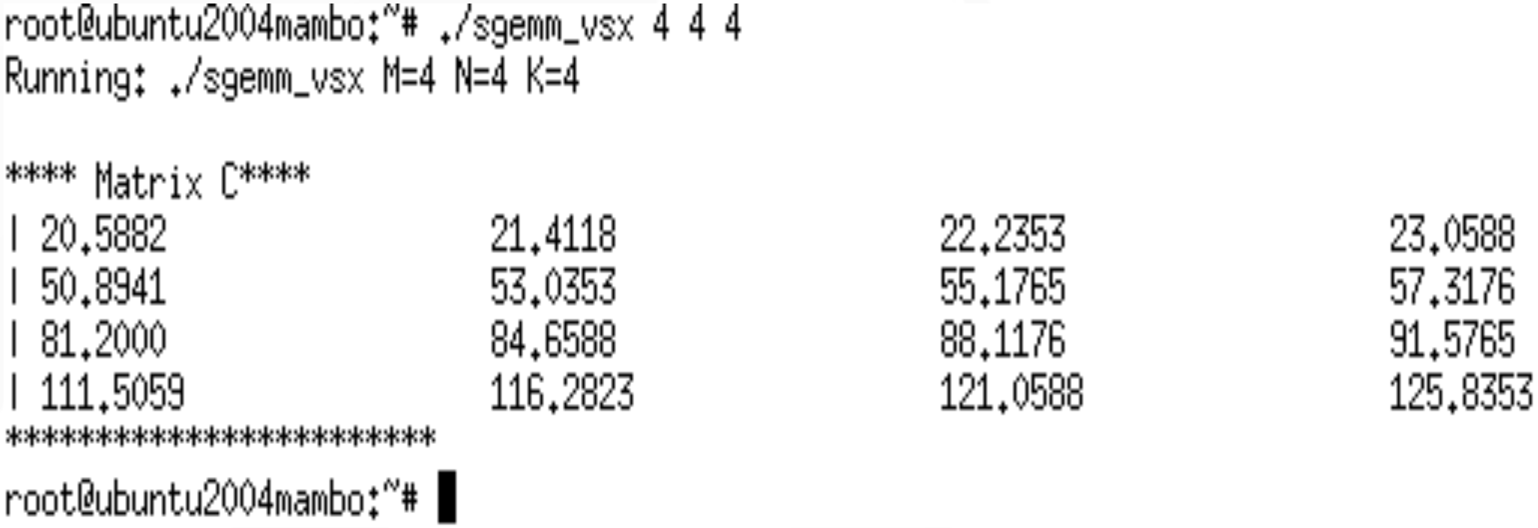
MMA xvf32gerpp instruction operation

Source: MMA Best Practice Guide

Example Exercise for DLP:

Getting Started with VSX

- Manual vectorization of matrix multiplication using vector scalar extensions (VSX)



```
root@ubuntu2004mambo:~# ./sgemm_vsx 4 4 4
Running: ./sgemm_vsx M=4 N=4 K=4

**** Matrix C****
| 20.5882             21.4118             22.2353             23.0588
| 50.8941             53.0353             55.1765             57.3176
| 81.2000             84.6588             88.1176             91.5765
| 111.5059            116.2823            121.0588            125.8353
************************
root@ubuntu2004mambo:~# █
```

- Tested the correctness on POWER10 functional simulator

Chapter 5: Thread-Level Parallelism (TLP)

- TLP in POWER CPUs
  - Curriculum on OpenMP on POWER
    - OpenMP can be compiled and run anywhere
  - Exercises on parallelization using OpenMP
    - Parallelize matrix multiplication using OpenMP
    - Evaluate the impact of work-sharing constructs
    - Combine DLP and TLP to further optimize matrix multiplication

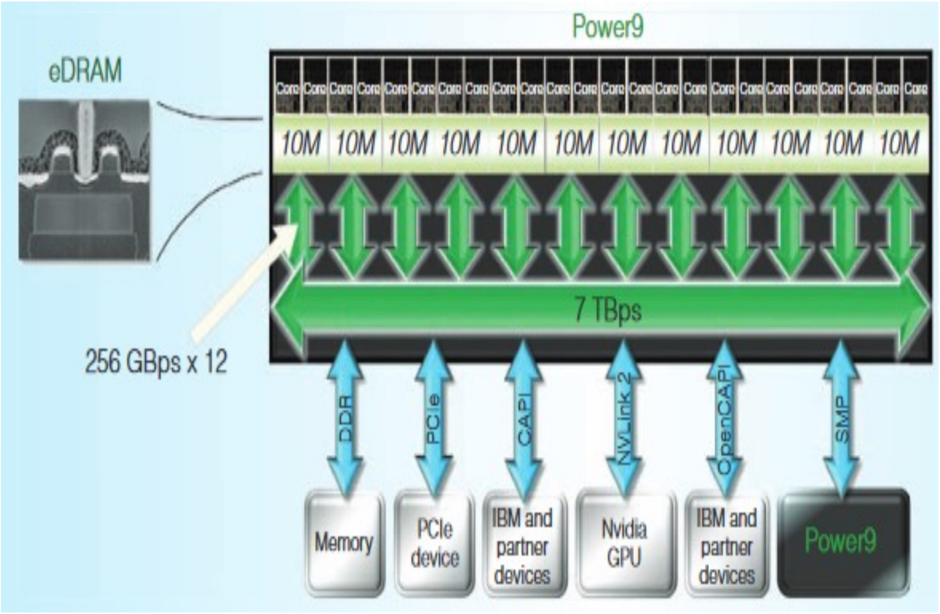
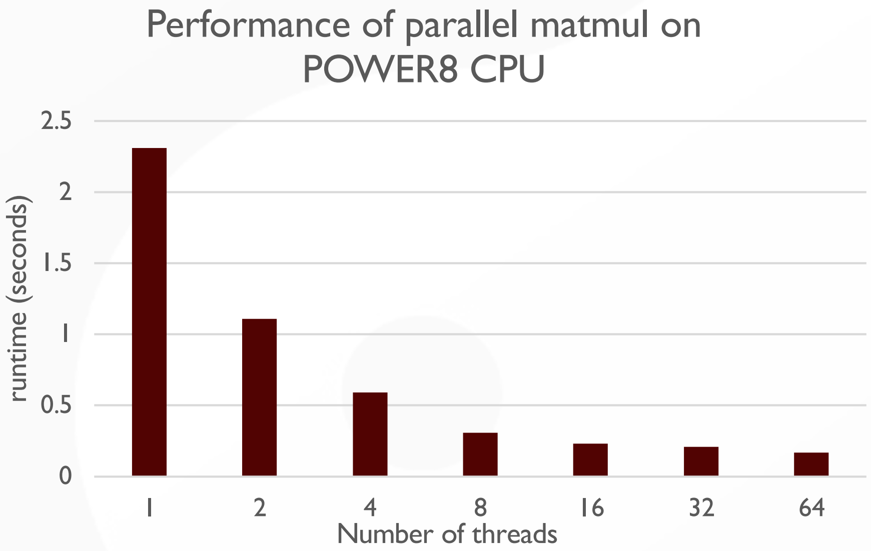
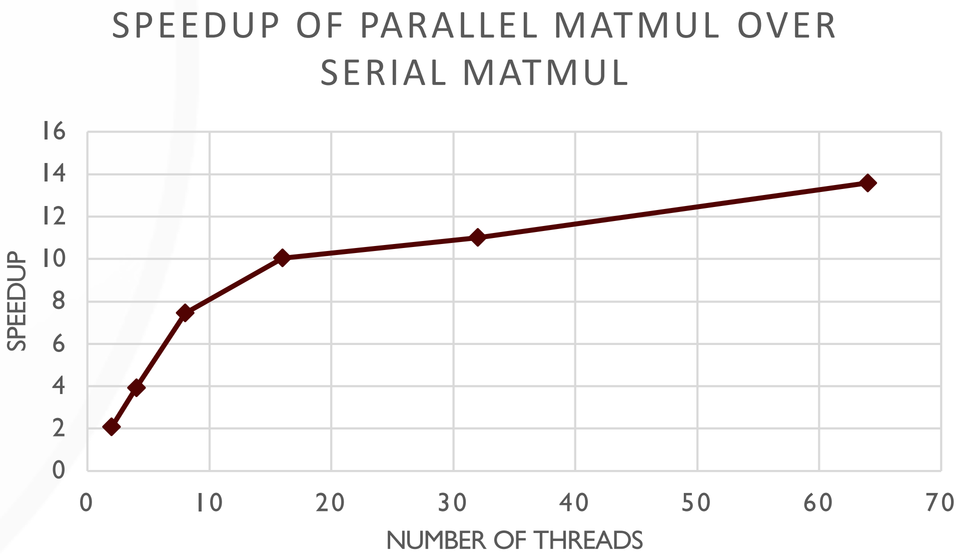

Power9
eDRAM
Core Core Core Core Core Core Core Core Core Core Core Core Core Core Core Core Core Core Core Core Core Core Core Core
10M 10M 10M 10M 10M 10M 10M 10M 10M 10M 10M 10M
7 TBps
256 GBps x 12
DDR
PCIe
CAPI
NVLink 2
OpenCAPI
SMP
Memory
PCIe device
IBM and partner devices
Nvidia GPU
IBM and partner devices
Power9

Image source: [IBM Power9 Processor Architecture]()

Example Exercise for TLP:
Parallelization via OpenMP

- OpenMP-based exercises on parallelizing the given workloads
  - Example: Evaluate the performance of parallelized matrix multiplication of two matrices of size 512 *512 on POWER8 CPU





Tools and Resources

- Tools/Emulators/Simulators
  - gem5: Used to evaluate workloads when architecture is varied
  - IBM POWER10 functional simulator, Libre-SOC, and Microwatt

.

.

.

.

Existing Major Issues in gem5 for POWER ISA

- No support for out-of-order (O3) execution for POWER ISA
  - Out-of-order (O3) execution is a *key* feature of modern POWER CPUs
  - Lack of support for O3 prevents cycle-accurate simulation of modern POWER CPU configurations
- Incorrect byte-swap function for simulating a *big-endian* ISA on a *little-endian* ISA
  - https://gem5.atlassian.net/browse/GEM5-1226
  - Operands casted into uint64_t
    - Problem: Fractional parts will get ignored in such a casting
- Unimplemented vector instructions
  - Example → `xxlxor` → performs a bitwise "xor" operation on two vectors
    - Problem: Lack of support for vector instructions prevents simulation of HPC benchmarks, e.g., LINPACK

Caveats for POWER ISA in gem5 & power-gem5

- GitHub: gem5
  - Read-only mirror of the gem5 simulator
  - Upstream repository at https://gem5.googlesource.com
  - Code reviews to https://gem5-review.googlesource.com/
  - Mirrors synchronized every 15 minutes.
- GitHub: power-gem5 (Forked from gem5/gem5)
  - Repository: https://github.com/power-gem5
  - Out-of-date mirror of gem5 with POWER code prototypes for gem5
    - Tooling: Relies on Python 2 support
    - Any code patches (updates or additions) that work in power-gem5 *must* be re-based for gem5, which now uses Python 3
- Recommendation
  - Develop POWER code patches from the gem5 repository at https://gem5.googlesource.com to avoid re-basing

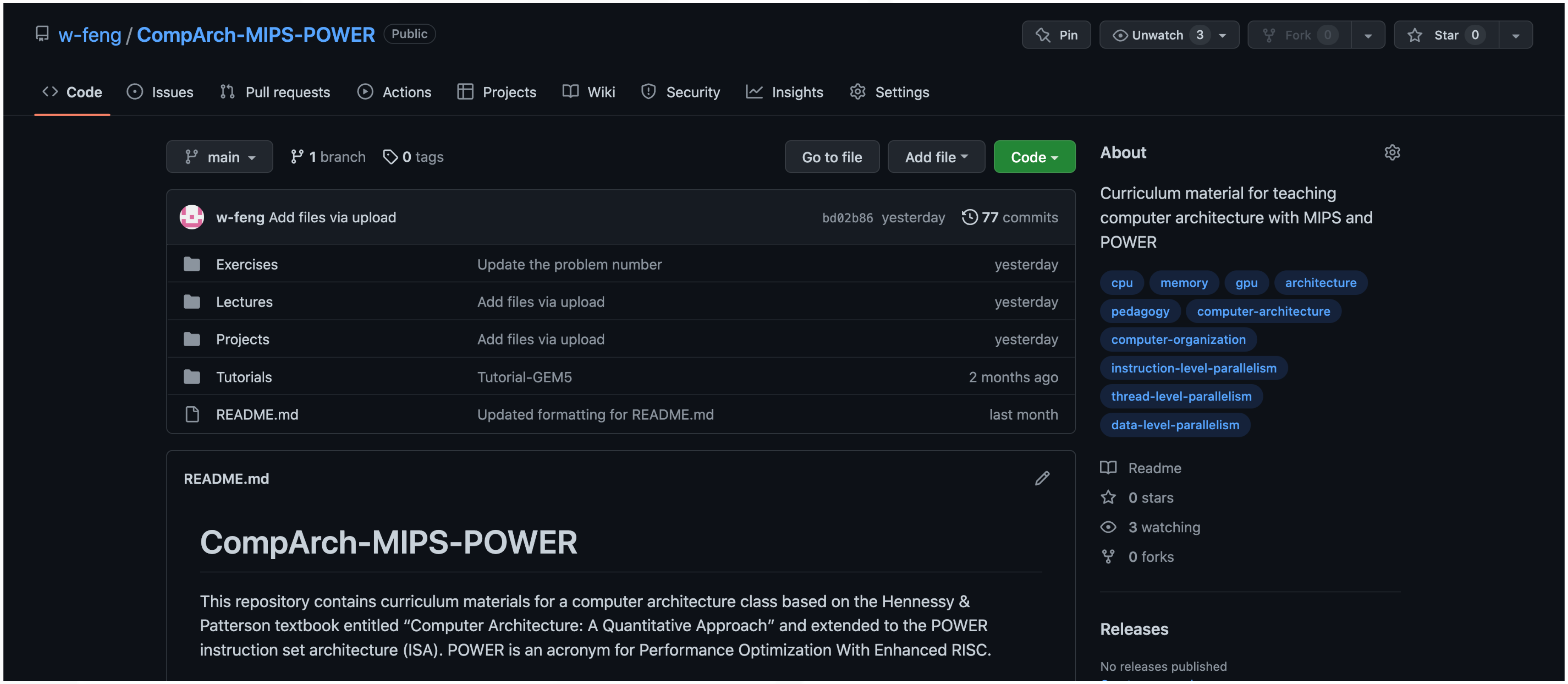
Tools and Resources

- Tools/Emulators/Simulators
  - gem5: Used to evaluate workloads when architecture is varied
  - IBM POWER10 functional simulator, Libre-SOC, and Microwatt
- Advanced Research Computing (ARC) Center @ VT
  - Huckleberry Cluster @ ARC: 14 IBM "Minksy" S822LC compute nodes
    - Each node with two IBM Power8 CPUs (3.26 GHz)

- Additional Resources
  - OpenPOWER Foundation
  - POWER9 Processor Architecture
  - POWER9 Processor User's Manual
  - POWER ISA 3.0
  - MMA Best Practices Guide
  - IBM – OpenMP support

GitHub https://github.com/w-feng/CompArch-MIPS-POWER

GitHub Repository for POWER-Oriented Curriculum (Lecture Slides, Tutorials, Exercises, and Projects)



w-feng Add files via upload bd02b86 yesterday 77 commits

Exercises	Update the problem number	yesterday
Lectures	Add files via upload	yesterday
Projects	Add files via upload	yesterday
Tutorials	Tutorial-GEM5	2 months ago
README.md	Updated formatting for README.md	last month

Curriculum material for teaching computer architecture with MIPS and POWER

README.md

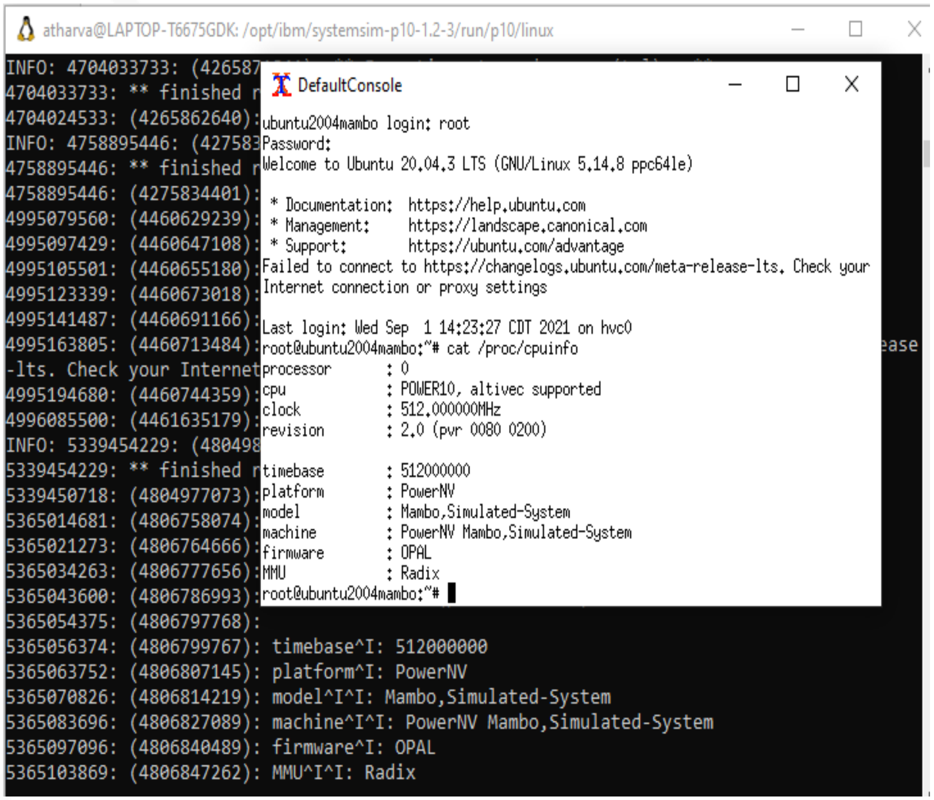
CompArch-MIPS-POWER

This repository contains curriculum materials for a computer architecture class based on the Hennessy & Patterson textbook entitled “Computer Architecture: A Quantitative Approach” and extended to the POWER instruction set architecture (ISA). POWER is an acronym for Performance Optimization With Enhanced RISC.

Bonus: POWER Projects from CS/ECE 5504: *Computer Architecture (Graduate-Level Course)*

GitHub https://github.com/w-feng/CompArch-MIPS-POWER

- *Projects* directory on GitHub
  - A list of prospective projects, oriented towards POWER
- *Tutorials* directory on GitHub
  - gem5 simulator (cycle accurate)
  - POWER10 functional simulator
  - Microwatt


atharva@LAPTOP-T6675GDK: /opt/ibm/systemsim-p10-1.2-3/run/p10/linux
DefaultConsole

```
INFO: 4704033733: (426587
4704033733: ** finished r
4704024533: (4265862640):
INFO: 4758895446: (427583
4758895446: ** finished r
4758895446: (4275834401):
4995079560: (4460629239):
4995097429: (4460647108):
4995105501: (4460655180):
4995123339: (4460673018):
4995141487: (4460691166):
4995163805: (4460713484):
-lts. Check your Internet
4995194680: (4460744359):
4996085500: (4461635179):
INFO: 5339454229: (480498
5339454229: ** finished r
5339450718: (4804977073):
5365014681: (4806758074):
5365021273: (4806764666):
5365034263: (4806777656):
5365043600: (4806786993):
5365054375: (4806797768):
5365056374: (4806799767): timebase^I: 512000000
5365063752: (4806807145): platform^I: PowerNV
5365070826: (4806814219): model^I^I: Mambo,Simulated-System
5365083696: (4806827089): machine^I^I: PowerNV Mambo,Simulated-System
5365097096: (4806840489): firmware^I: OPAL
5365103869: (4806847262): MMU^I^I: Radix
```

```
ubuntu2004mambo login: root
Password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.14.8 ppc64le)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

Last login: Wed Sep  1 14:23:27 CDT 2021 on hvc0
root@ubuntu2004mambo:~# cat /proc/cpuinfo
processor       : 0
cpu             : POWER10, altivec supported
clock           : 512.000000MHz
revision        : 2.0 (pvr 0080 0200)

timebase        : 512000000
platform        : PowerNV
model           : Mambo,Simulated-System
machine         : PowerNV Mambo,Simulated-System
firmware        : OPAL
MMU             : Radix
root@ubuntu2004mambo:~#
```

VIRGINIA TECH

SyNeRG synergy.cs.vt.edu

Summary

- POWER-Oriented Curriculum
  - Content
    - Lecture slides, tutorials, exercises, and projects
  - Target Audience (e.g., *CS/ECE 4504: Computer Organization* at VT)
    - A senior-level undergraduate course to beginning graduate-level course
  - "Standing on the Shoulders of Giants"
    - Based on the Hennessy & Patterson textbook, which uses the MIPS ISA.

Acknowledgements

- This work was supported in part by the following:
  - Department of Computer Science at Virginia Tech (CS@VT)
  - IBM Global University Programs
  - Textbook: Hennessy & Patterson, “Computer Architecture: A Quantitative Approach,” 6th edition, Elsevier, 2017.

COLLEGE OF ENGINEERING
COMPUTER SCIENCE
VIRGINIA TECH

IBM

John L. Hennessy | David A. Patterson
COMPUTER ARCHITECTURE
A Quantitative Approach