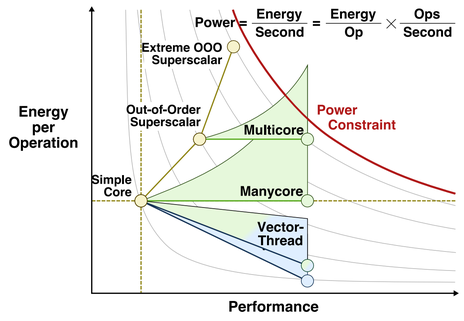
Maven: A Manycore Vector-Thread Processor

Yunsup Lee, Christopher Batten, Krste Asanović
Parallel Computing Laboratory
University of California, Berkeley

1 Motivation

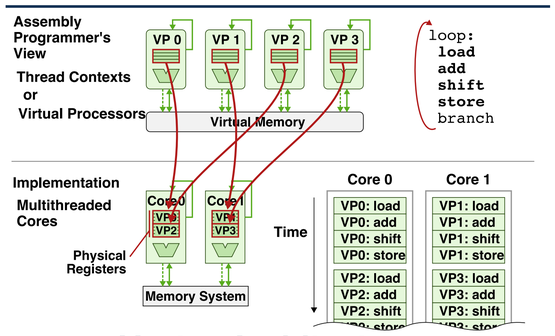
Future manycore processors will be energy-constrained, and thus the primary metric for evaluating these architectures will be their **energy-efficiency**. In this work, we investigate new architectural and microarchitectural mechanisms which enable a wider array of applications to be mapped to energy-efficient vector units.



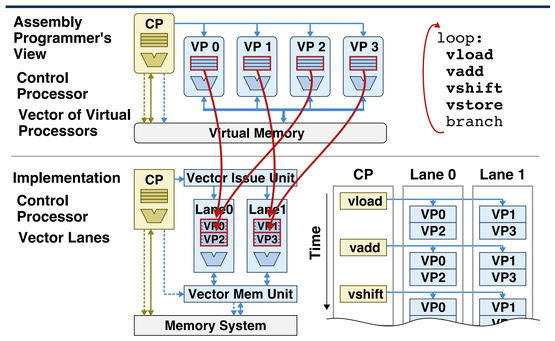
2 VT Architecture Paradigm

Vector-threading (VT) is a new abstraction for programmers in which a control processor (CP) manages a **vector of virtual processors** (VPs). The control processor can use vector-fetch commands to broadcast instructions to all VPs, or each VP can use VP control instructions to direct its own control flow. Vector memory commands move blocks of data in and out of VP registers, while VP loads and stores enable less structured data-access patterns.

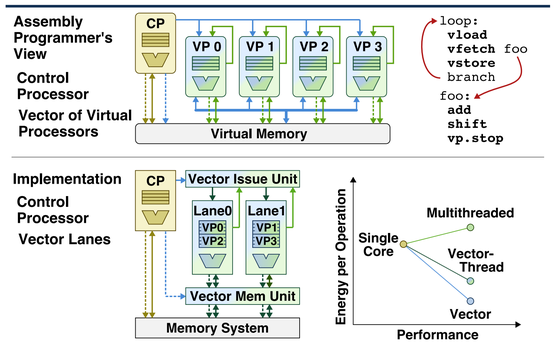
Multithreaded Architectures



Vector Architectures



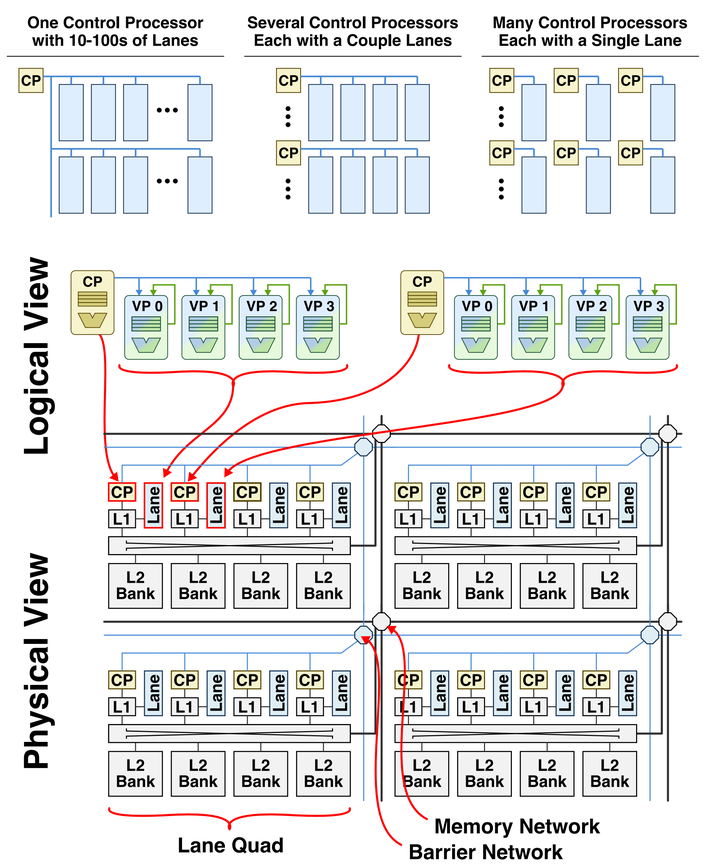
Vector-Thread Architectures



Yunsup Lee is supported by Microsoft (Award #024263) and Intel (Award #024894) funding and by matching funding by U.C. Discovery (Award #DIG07-10227).

3 The Maven VT Processor

The Maven VT processor is a manycore vector-thread architecture which uses a "**sea-of-lanes**" approach with tens to hundreds of single-lane vector-thread units tiled across the chip. Each lane is highly tuned to exploit data-level parallelism temporally, and lanes can be "ganged" together to exploit some degree of data-level parallelism spatially.



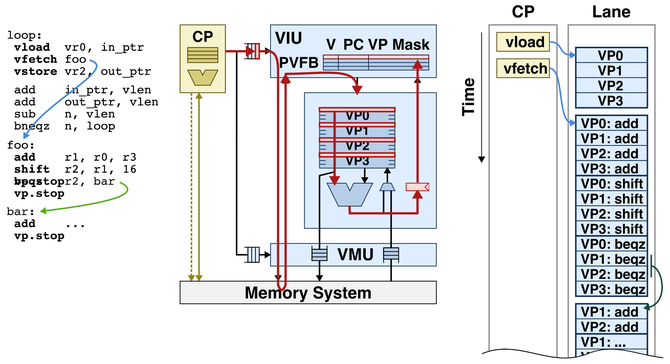
Architectural Features

- VP ISA = CP ISA
- Constant vector registers
- Trade-off vector length versus number of vector registers
- Fast hardware barrier implemented
- Vector memory accesses supported

Microarchitectural Features

- Single issue; issue logic amortized over full vector
- Multiple instructions can execute per cycle
- VP branches create vector fragments
  - Partial amortization
  - Dynamic reconvergence
  - Fragment collapsing
  - Fragment multi-issue

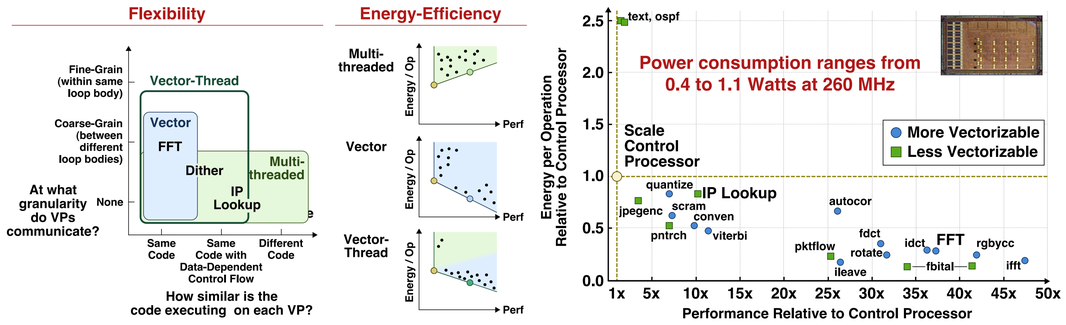
Maven Lane Microarchitecture



- Single issue; dependency checks & interlocks happen in issue unit
- VP branch resolutions are OR'd into the branch resolution mask and sent to issue unit
  - If mask is all zeros or all ones then issue unit continues, else ...
  - Issue unit inserts target PC and mask into pending vector fragment buffer (PVFB)
  - Issue unit continues down non-taken path
  - Issue unit can check PVFB for reconvergence
- Additional mechanisms for fragment multi-issue, fragment collapsing

4 Energy-Efficiency for Various Kinds of Kernels

The left graph shows the flexibility of vector-thread, vector, and multithreaded architectures. It also maps kernels on the energy-performance graph as well. As we don't have a energy model for Maven yet, we present the actual measured energy-efficiency on the Scale VT Processor. The Scale VT Processor is an instantiation of the vector-threading architectural paradigm designed for low-power and high-performance embedded systems. The logical virtual processor vector is striped across an array of physical **vector lanes**, so that each lane is responsible for managing the state and execution of multiple virtual processors.



5 Executing Kernels on Maven

We picked three kernels (FFT, Dither, and IP Lookup) and show how they execute on Maven. The FFT kernel is straightfoward to vectorize. The IP lookup kernel seems easier to map it to a multithreaded architecture. The image dithering kernel is somewhat in the middle. Maven provides an easy programming model for these three kernels, and can run them energy-efficiently.

Fast Fourier Transform

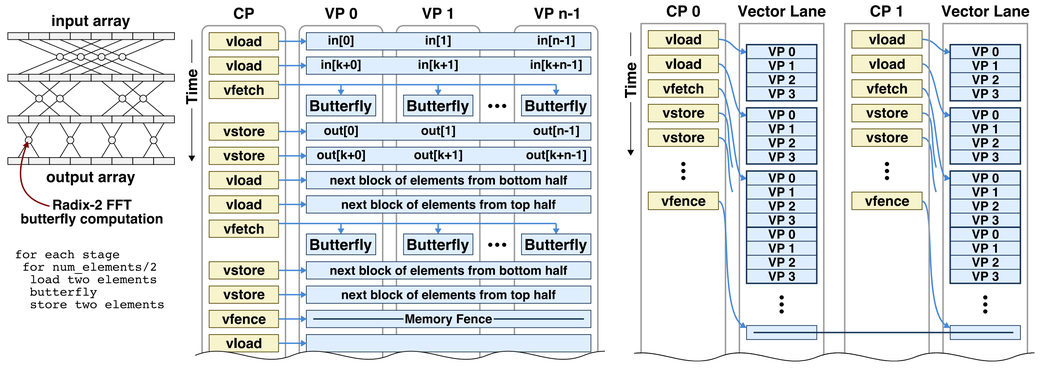
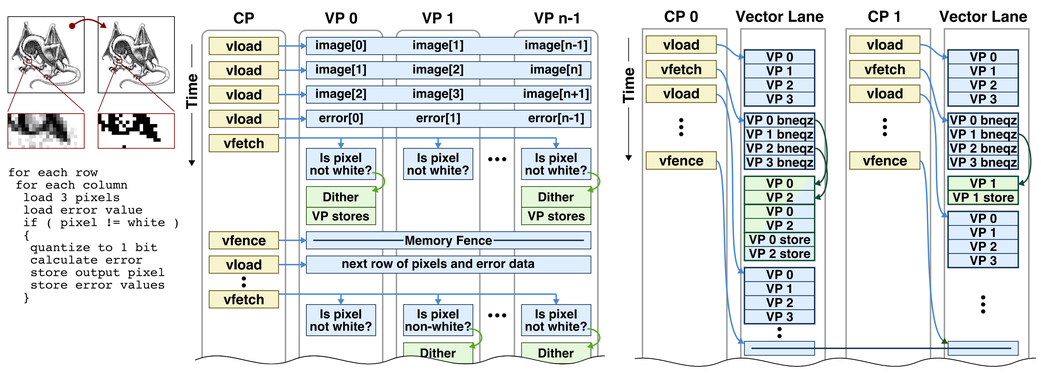
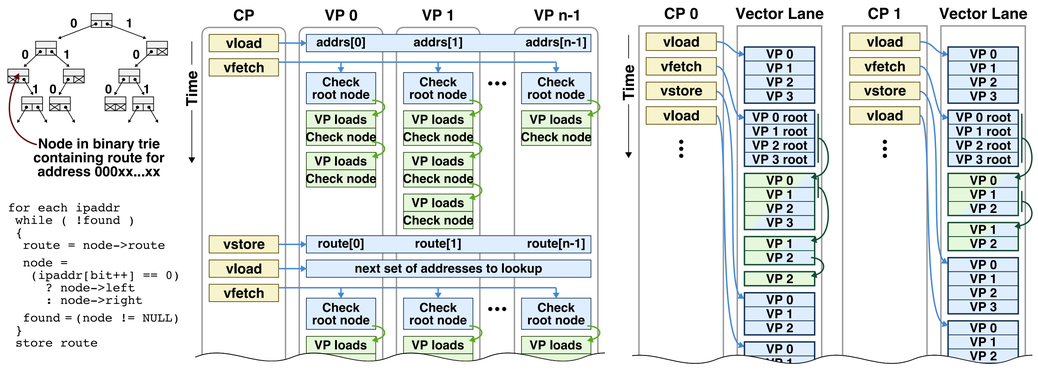


Image Dithering



IP Lookup



Yunsup Lee, Christopher Batten — 577C, 582 Soda Hall, Berkeley, CA 94720 — yunsup@cs.berkeley.edu, cbatten@mit.edu — http://www.cs.berkeley.edu/~yunsup, http://www.mit.edu/~cbatten