

---

---

# Lecture 17: Parallel Architectures and Future Computer Architectures

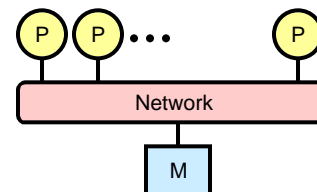
Prof. Kunle Olukotun  
EE 282h  
Fall 98/99

1

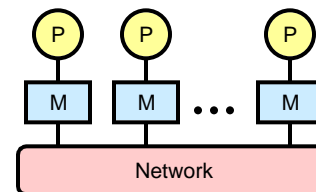
---

## Shared-Memory Multiprocessors

- Several processors share one address space
  - » conceptually a shared memory
  - » often implemented just like a multicomputer
    - address space distributed over private memories
- Communication is *implicit*
  - » read and write accesses to shared memory locations
- Synchronization
  - » via shared memory locations
    - spin waiting for non-zero
  - » barriers



Conceptual Model



Actual Implementation

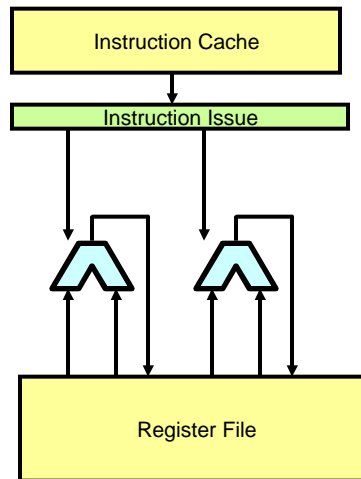
2





## IA-64 aka VLIW

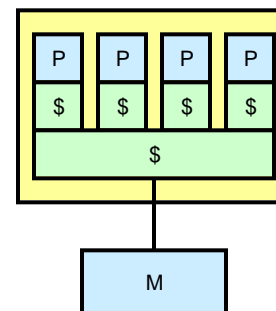
- Compiler schedules instructions
- Encodes dependencies explicitly
  - » saves having the hardware repeatedly rediscover them
- Support speculation
  - » speculative load
  - » branch prediction
- Really need to make communication explicit too
  - » still has global registers and global instruction issue



7

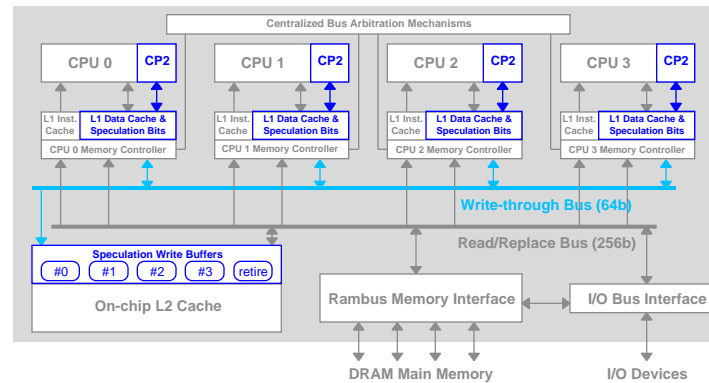
## Single-Chip Multiprocessors

- Build a multiprocessor on a single chip
  - » linear increase in *peak* performance
  - » advantage of fast interaction between processors
- Fine grain threads
  - » make communication and synchronization very fast (1 cycle)
  - » break the problem into smaller pieces
- Memory bandwidth
  - » Makes more effective use of limited memory bandwidth
- Programming model
  - » Need parallel programs



8

# Base Hydra Design

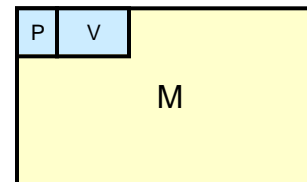


- > Single-chip multiprocessor
- > Four processors
- > Separate primary caches
- > Write-through data caches to maintain coherence
- > Shared 2nd-level cache
- > Separate read and write busses
- > Data Speculation Support

9

## Processor with DRAM (PIM) IRAM, VIRAM

- Put the processor and the main memory on a single chip
  - » much lower memory latency
  - » much higher memory bandwidth
- But
  - » need to build systems with more than one chip



64Mb SDRAM Chip  
 Internal - 128 512K subarrays  
 4 bits per subarray each 10ns  
 51.2 Gb/s

External - 8 bits at 10ns, 800Mb/s

1 Integer processor ~ 100KBytes DRAM  
 1 FP processor ~ 500KBytes DRAM

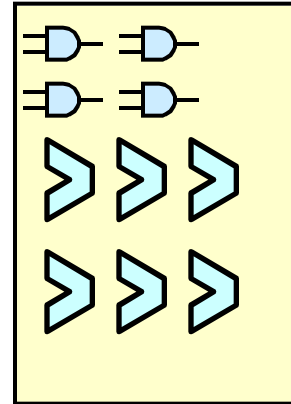
1 Vector Unit ~ 1 MByte DRAM

10

## Reconfigurable processors

---

- Adapt the processor to the application
  - » special function units
  - » special wiring between function units
- Builds on FPGA technology
  - » FPGAs are inefficient
    - a multiplier built from an FPGA is about 100x larger and 10x slower than a custom multiplier.
  - » Need to raise the granularity
    - configure ALUs, or whole processors
  - » Memory and communication are usually the bottleneck
    - not addressed by configuring a lot of ALUs
- Programming model
  - » Difficult to program
  - » Verilog



11

## Have a Good Christmas Break

---

- and good luck on the final.

12