

ECE 4750 Computer Architecture, Fall 2016

T17 Advanced Processors: Multithreaded Processors

School of Electrical and Computer Engineering
Cornell University

revision: 2016-12-02-09-26

1	Multithreading Overview	2
2	Vertical Multithreading	3
3	Simultaneous Multithreading	6

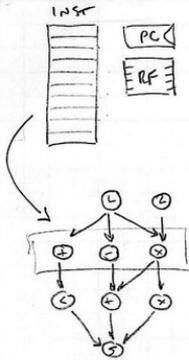
1. Multithreading Overview

ILP vs DLP vs TLP

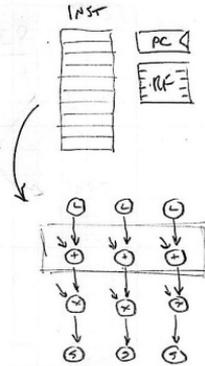
ILP = INSTRUCTION-LEVEL PARALLELISM
 DLP = DATA-LEVEL PARALLELISM
 TLP = THREAD-LEVEL PARALLELISM

SISD	MISD
SIMD	MIMD

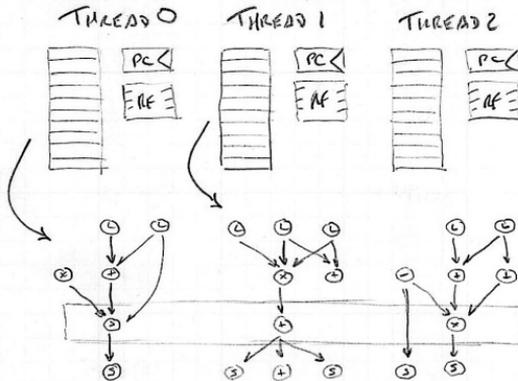
ILP



DLP



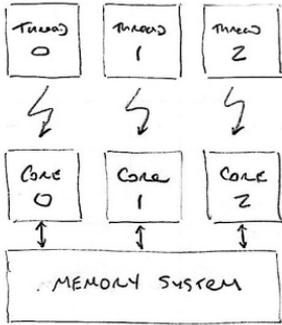
TLP



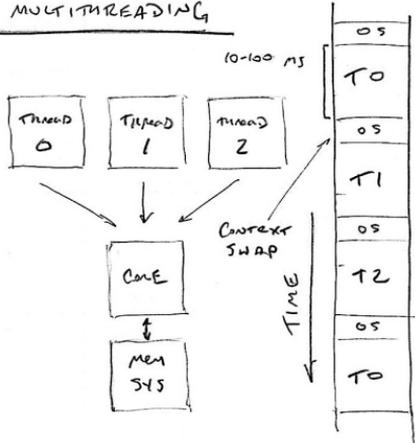
- TLP from multiprogramming (multiple applications)
- TLP from multithreaded applications
 - ↳ run one application faster with multiple threads
 - ↳ PTHREADS, CLIK, TBB, OPENMP

2. Vertical Multithreading

MULTICORE VS COARSE-GRAIN MULTITHREADING



MULTICORE



COARSE-GRAIN MULTITHREADING

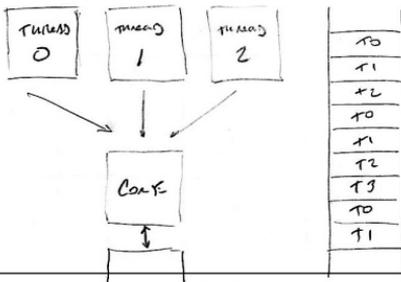
FINE-GRAIN MULTITHREADING

HARDWARE SUPPORT TO ENABLE INTERLEAVING MULTIPLE THREADS OF A SINGLE CORE AT A VERY FINE GRANULARITY.

WE WILL DISCUSS TWO VARIANTS OF FINE-GRAIN MULTITHREADING:

- VERTICAL MULTITHREADING
- SIMULTANEOUS MULTITHREADING (SMT)

VERTICAL MULTITHREADING



SWITCH BETWEEN THREADS AT A CYCLE-BY-CYCLE GRANULARITY

STATE FOR ALL THESE THREADS KEPT IN DEDICATED HARDWARE

THREAD SCHEDULING HANDLED BY HARDWARE

SCHEDULING POLICIES

1. STATIC FIXED INTERLEAVING

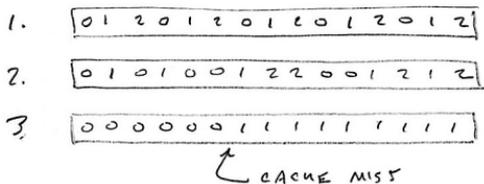
- EACH OF N THREADS EXECUTE ONE INSTRUCTION EVERY N CYCLES
- IF THREAD IS NOT READY TO GO CAN EITHER:
 - STALL ENTIRE FRONT-END
 - INSERT DUBBLE, BUT DO NOT STALL FRONT-END
- CAN POTENTIALLY ELIMINATE INTERLOCKING + BYPASS NETWORK

2. DYNAMIC INTERLEAVING

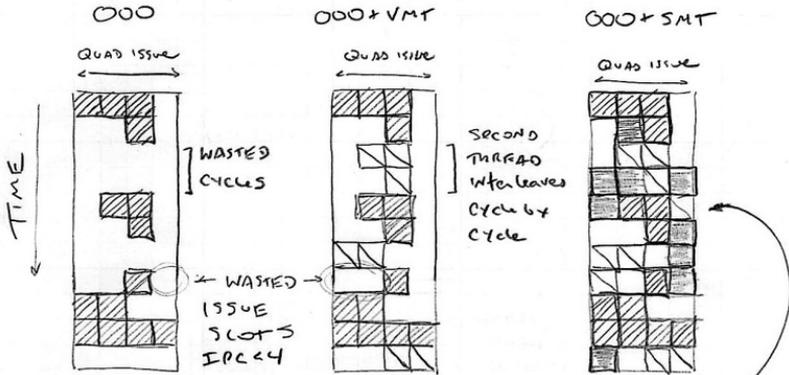
- HARDWARE KEEPS TRACK OF WHICH THREADS ARE READY
- PICKS NEXT THREAD TO EXECUTE BASED ON PRIORITY SCHEME

3. COARSE-GRAIN HARDWARE INTERLEAVING

- USE THREADS TO HIDE OCCASIONAL CACHE MISS LATENCY



SIMULTANEOUS MULTITHREADING (SMT)



ON THIS cycle WE ARE ISSUING FOUR INSTRUCTIONS FROM THREE THREADS AT THE SAME TIME

SMT USES THE FINE GRAIN CONTROL ALREADY PRESENT IN AN OOO SUPERSCALAR PROCESSOR TO ALLOW INSTRUCTIONS FROM DIFFERENT THREADS TO ISSUE AT THE SAME TIME

ADD MULTIPLE FETCH ENGINE TO ENABLE FETCHING + DECODING INSTRUCTIONS FROM DIFFERENT THREADS

IQ DOES NOT KNOW ABOUT THREADS. SIMPLY FINDS INSTRUCTIONS THAT ARE READY TO ISSUE - THESE INSTRUCTIONS MAY OR MAY NOT BE FROM DIFFERENT THREADS

* SMT ADAPTS TO PARALLELISM TYPE

- FOR APPLICATIONS WITH HIGH ICP BUT NO TLP, APP CAN USE ENTIRE WIDTH OF THE MACHINE
- FOR APPLICATIONS WITH HIGH TLP BUT LESS ICP, THE WIDTH OF THE MACHINE IS SHARED ACROSS THREADS

3. Simultaneous Multithreading

Draw a pipeline diagram for the assembly loop to the right executing on a dual-issue IO2L microarchitecture with register renaming, memory disambiguation, perfect branch prediction, and *two* SMT threads. Draw the diagram to illustrate how both threads simultaneously execute the first iteration of the loop.

```
lw    x1, 0(x2)
mul   x3, x1, x4
sw    x3, 0(x5)
addi  x2, x2, 4
addi  x5, x5, 4
addi  x7, x7, -1
bgtz  x7, loop
```

