

CprE 381: Computer Organization and Assembly Level Programming

Parallelism

Henry Duwe
Electrical and Computer Engineering
Iowa State University

Administrative

- OH 2-3pm TODAY
- HW11 due on Mon April 29
 - Real cache exploration
 - Final HW
- Part 4 due in lab next week
 - **WARNING:** No extensions!!!
- Final Exam (Really Exam 3)
 - When: Mon May 6 at 7:30am-9:30am
 - Where: **Marston 2155 (everyone)**
 - What: Control hazards and data forwarding through HW security

Parallelism is Everywhere

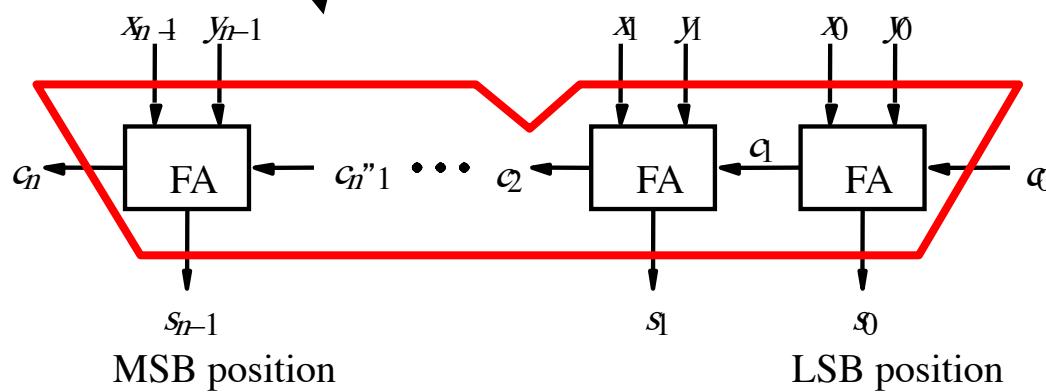
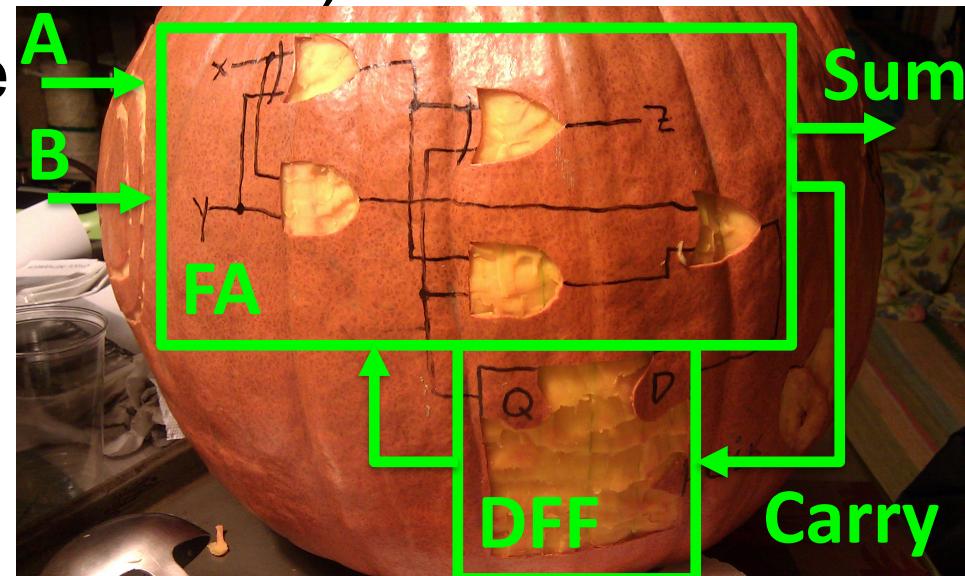
- Where have we already encountered parallelism?

A (Brief) History Lesson in Parallelism

- Early computers (circa 1951) bit-serial
 - Smaller size/volume

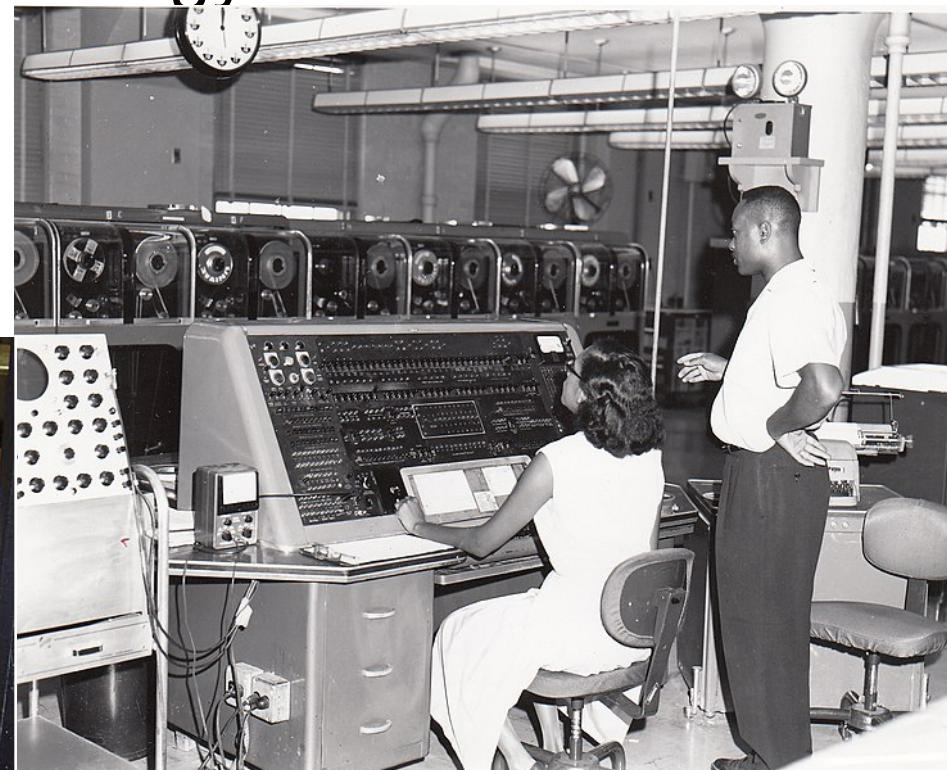
Bit-serial

Bit-parallel



A (Brief) History Lesson in Parallelism

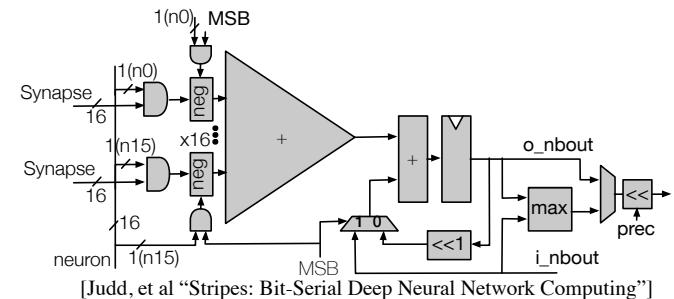
- Early computers (circa 1951) bit-serial
 - Smaller size/volume
 - Memory/storage technology interface
 - Example:
 - UNIVAC I
 - Mercury memory



[https://commons.wikimedia.org/wiki/File:Mercury_memory.jpg]

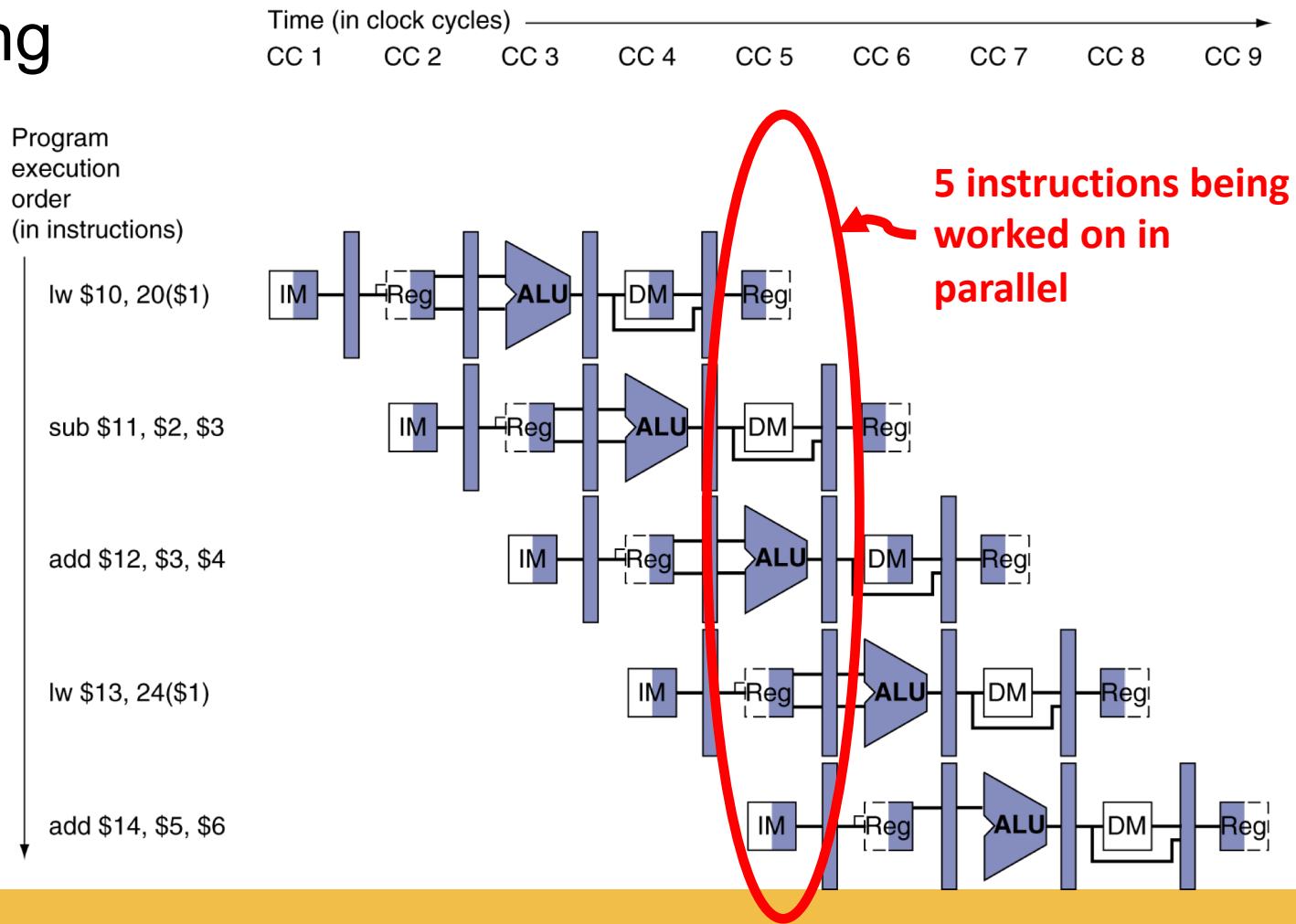
A (Brief) History Lesson in Parallelism

- Early (circa 1980s) massively parallel computers bit-serial ALUs
- Almost all computers bit-parallel today
 - Microcontrollers: 16-, 32-bit
 - Microprocessors: 32-, 64-bit
- Some current proposals for useful bit-serial computation
 - Dynamically varying precision



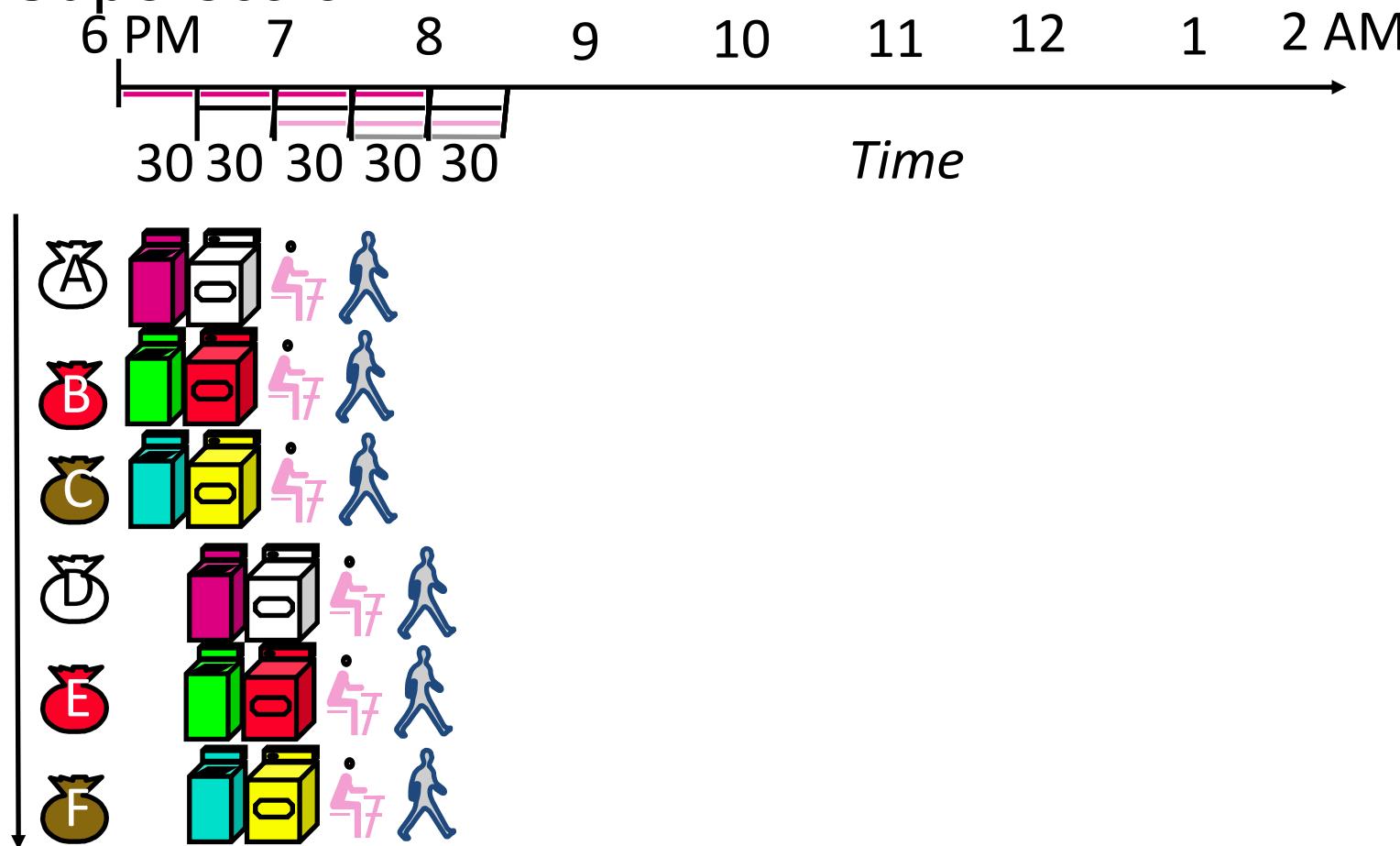
A (Brief) History Lesson in Parallelism

- In search of instruction-level parallelism (ILP)
 - Pipelining



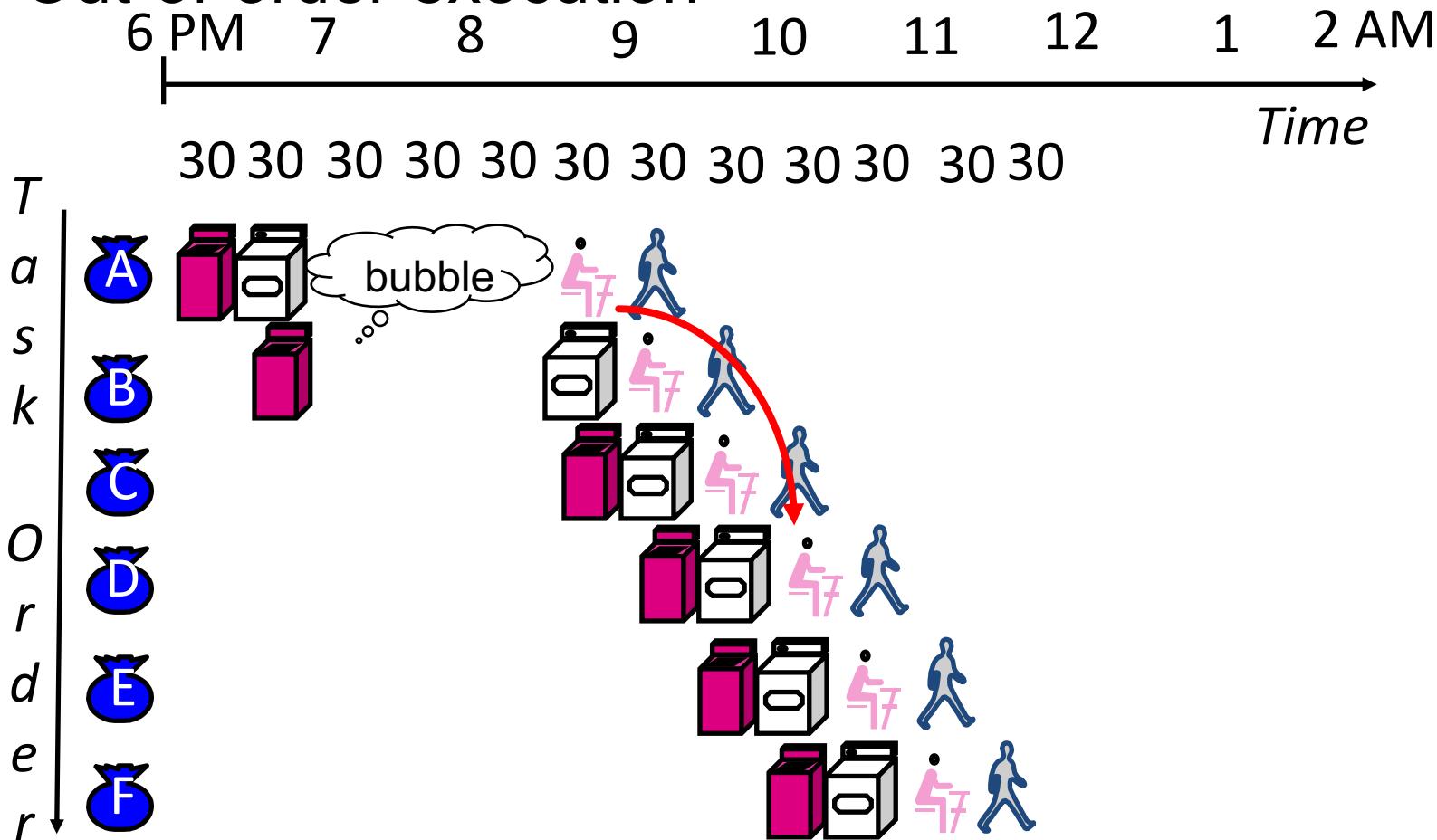
A (Brief) History Lesson in Parallelism

- In search of instruction-level parallelism (ILP)
 - Superscalar



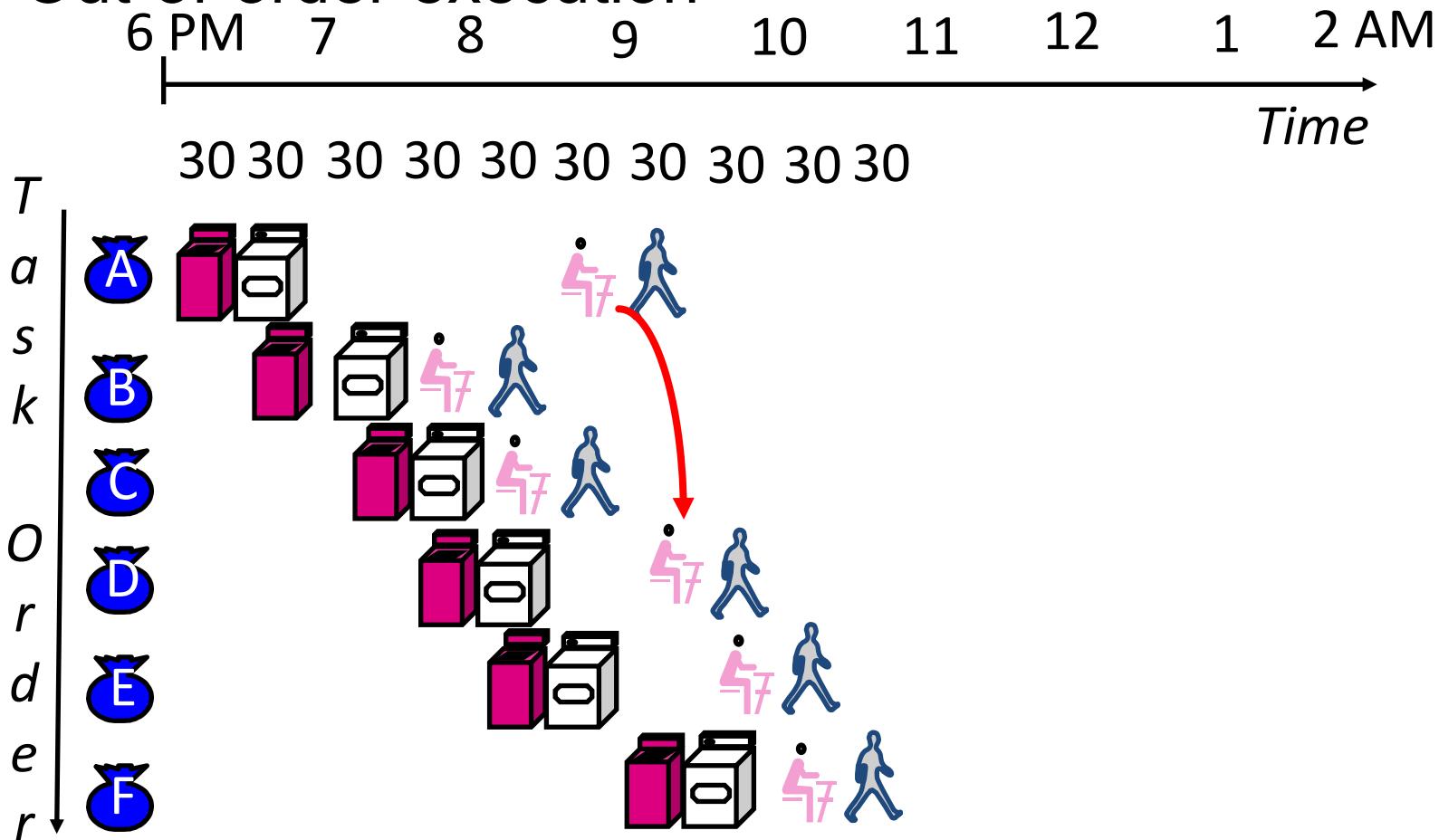
A (Brief) History Lesson in Parallelism

- In search of instruction-level parallelism (ILP)
 - Out-of-order execution



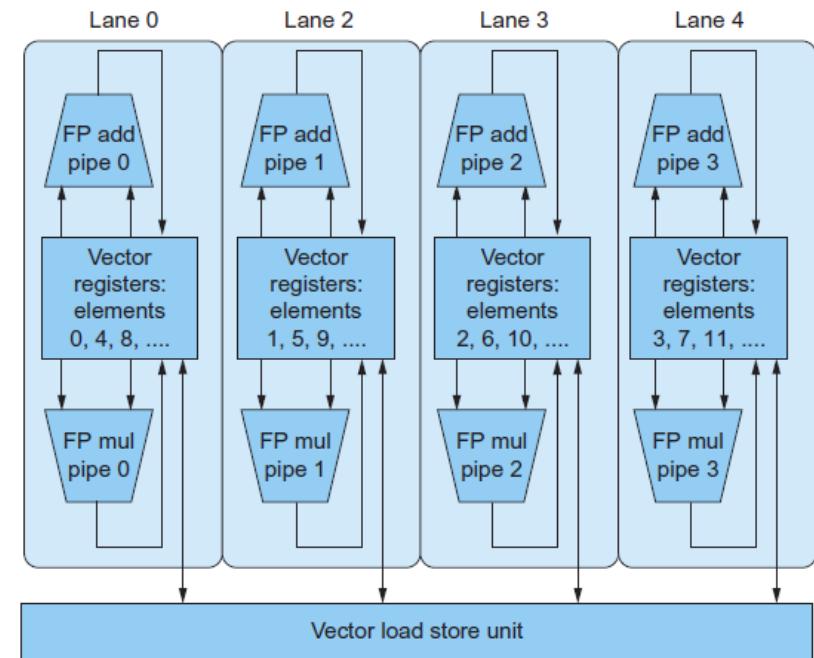
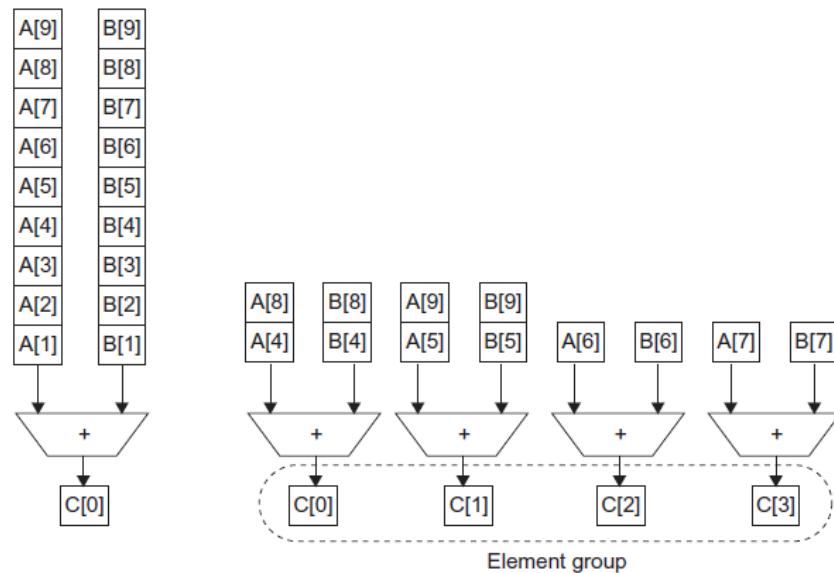
A (Brief) History Lesson in Parallelism

- In search of instruction-level parallelism (ILP)
 - Out-of-order execution



Review: SIMD Vector and Multimedia Extensions

- Simplify data-parallel programming
- Significantly reduces instruction-fetch bandwidth
- Vector instructions have a variable vector width, multimedia extensions have a fixed width



Review: SIMD Vector and Multimedia Extensions

- Simplify data-parallel programming
- Significantly reduces instruction-fetch bandwidth
- Vector instructions have a variable vector width, multimedia extensions have a fixed width



**Cray-1
“Super
Computer”
Vector
Processor
(circa 1976)**



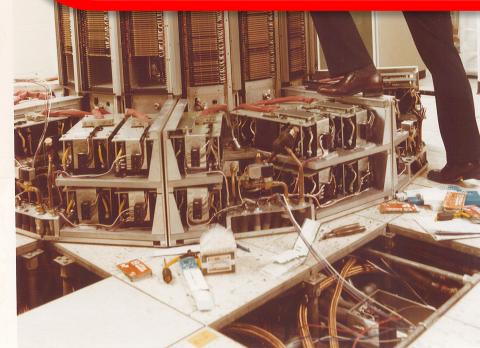
Review: SIMD Vector and Multimedia Extensions

- Simplify data-parallel programming
- Significantly reduces instruction-fetch bandwidth
- ~~Vector instructions have a variable vector width~~

In-class Assessment!

Access Code: CoreCore

Note: sharing access code to those outside of classroom or using access code while outside of classroom is considered cheating



Computer
Vector
Processor
(circa 1976)



Multiprocessors

- Multicore microprocessors
 - More than one processor per chip
- Requires explicitly parallel programming
 - Compare with instruction level parallelism
 - Hardware executes multiple instructions at once
 - Hidden from the programmer
 - Hard to do
 - Programming for performance
 - Load balancing
 - Optimizing communication and synchronization

Amdahl's Law (Again)

- Sequential part can limit speedup
- Example: 100 processors, 90× speedup?
 - $T_{\text{new}} = T_{\text{parallelizable}}/100 + T_{\text{sequential}}$
 - Speedup = $\frac{1}{(1-F_{\text{parallelizable}}) + F_{\text{parallelizable}}/100} = 90$
 - Solving: $F_{\text{parallelizable}} = 0.999$
- Need sequential part to be 0.1% of original time

Acknowledgments

- These slides contain material developed and copyright by:
 - Joe Zambreno (Iowa State)
 - Akhilesh Tyagi (Iowa State)
 - David Patterson (UC Berkeley)
 - Mary Jane Irwin (Penn State)
 - Christos Kozyrakis (Stanford)
 - Onur Mutlu (Carnegie Mellon)
 - Krste Asanović (UC Berkeley)
 - Karu Sankaralingam (UW Madison)
 - Morgan Kaufmann