CS 271
Computer Architecture & Assembly Language

Lecture 19
More Hardware
RISC Architecture
3/8/22, Tuesday

Oregon State University

Odds and Ends

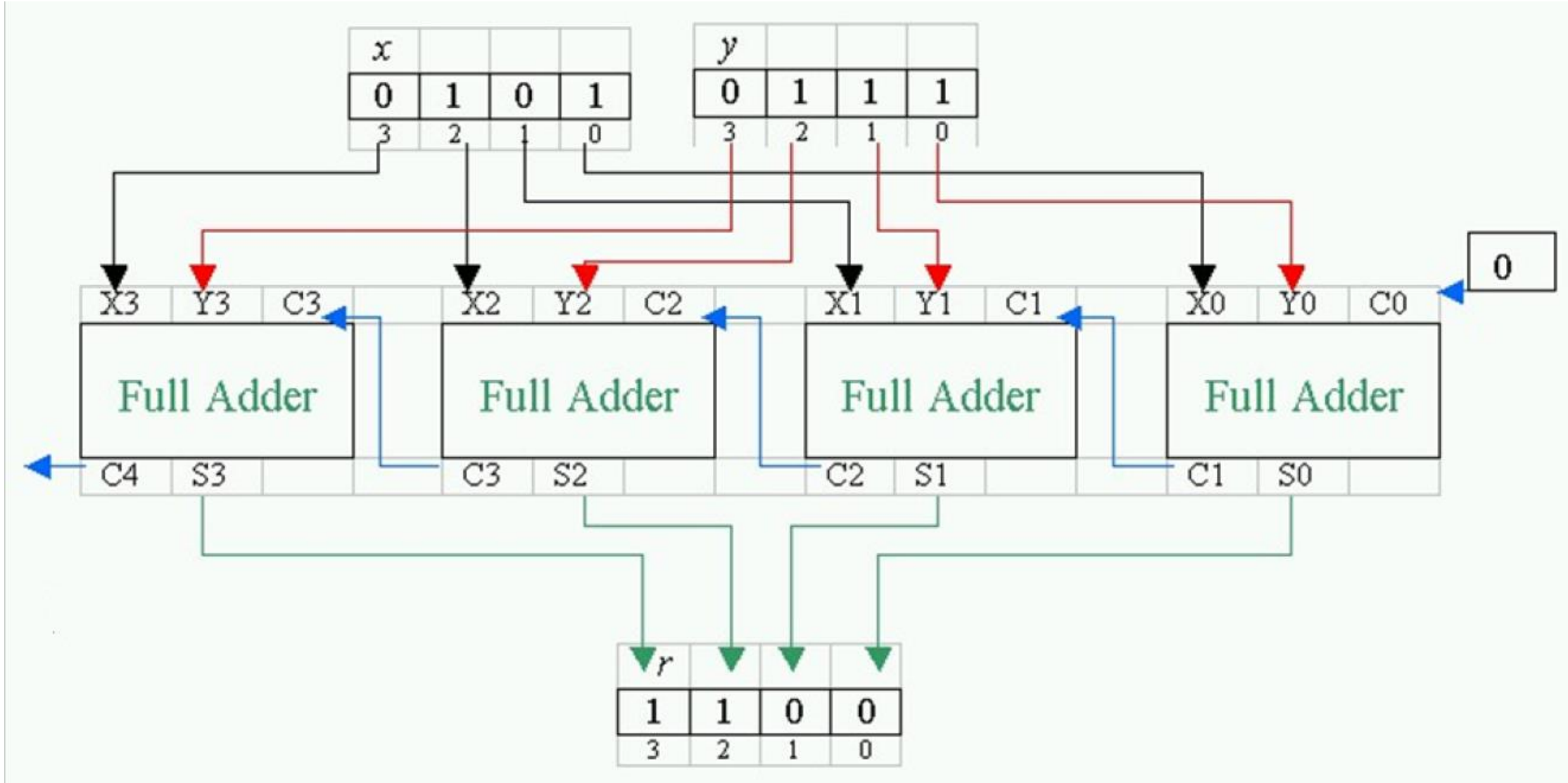
- Clarifications
  - Assignment 5 Grades
  - Final Project:
    - Sign extension for the decoy mode is NOT required if not doing EC
      - i.e., if inputs are -1 and -2, the output doesn't have to be -3
    - Comment out the main procedure and .data when submitting the file

Lecture Topics:

- More Architecture
- RISC Architecture

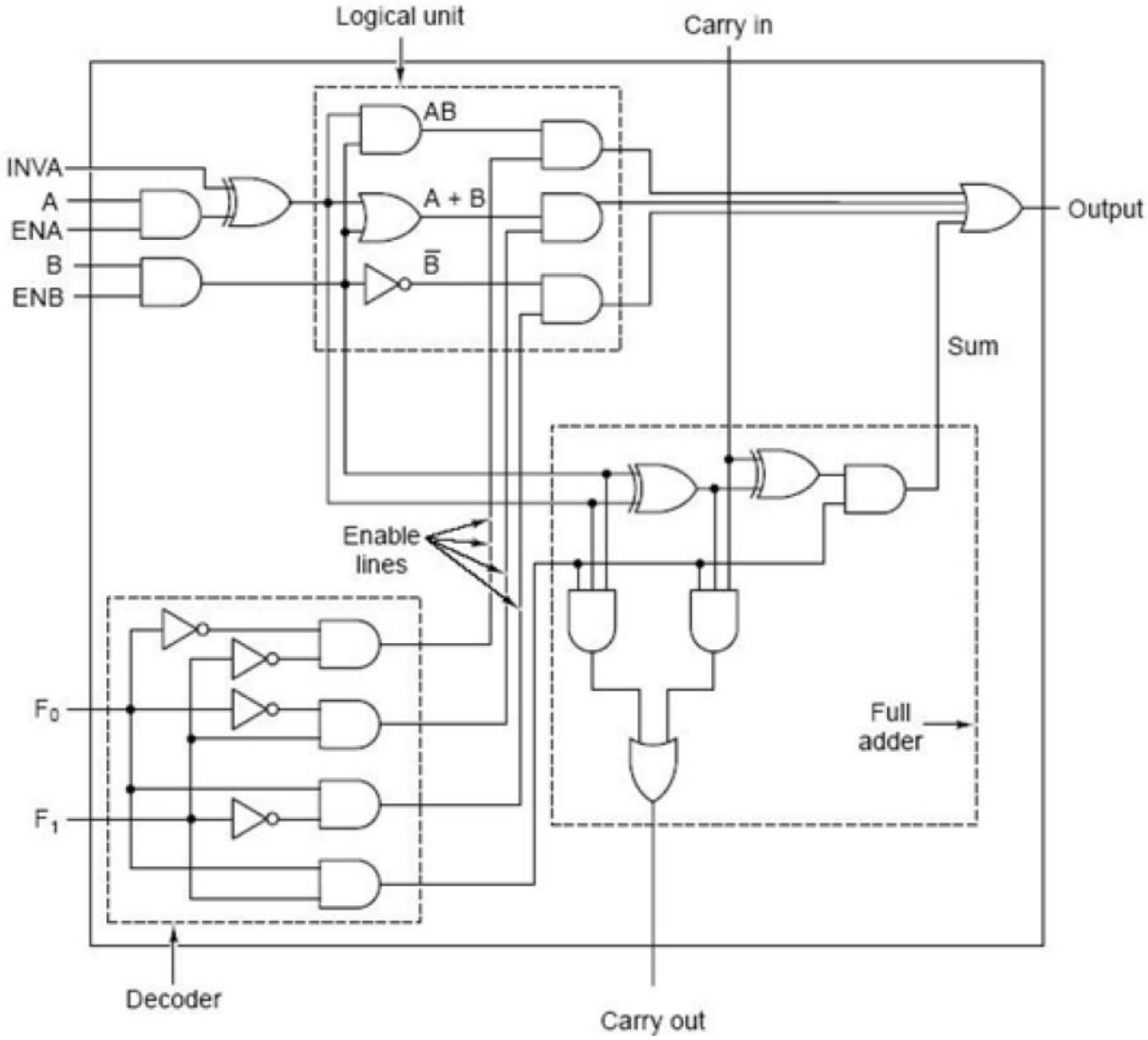
More Architecture

Ripple Carry Adder



“Growing” an ALU

- … from integrated circuits
  - Full-adder
  - 2s-complementer
  - Shifter
  - Comparator



“Growing” an ALU

- ALU
- Registers
- Control Unit
  - Clock
  - Instruction pointer
  - Instruction fetch, decode, execute
  - Operand fetch, store
- Addressing unit

Tying Things Togethers:

The internal bus

- Control Unit, ALU, Registers, Addressing Unit communicate via a bus.
- Speed depends on
  - Bus width: number of bits that can transfer simultaneously
  - Bus length
- Bus arbitration
- Multiple buses

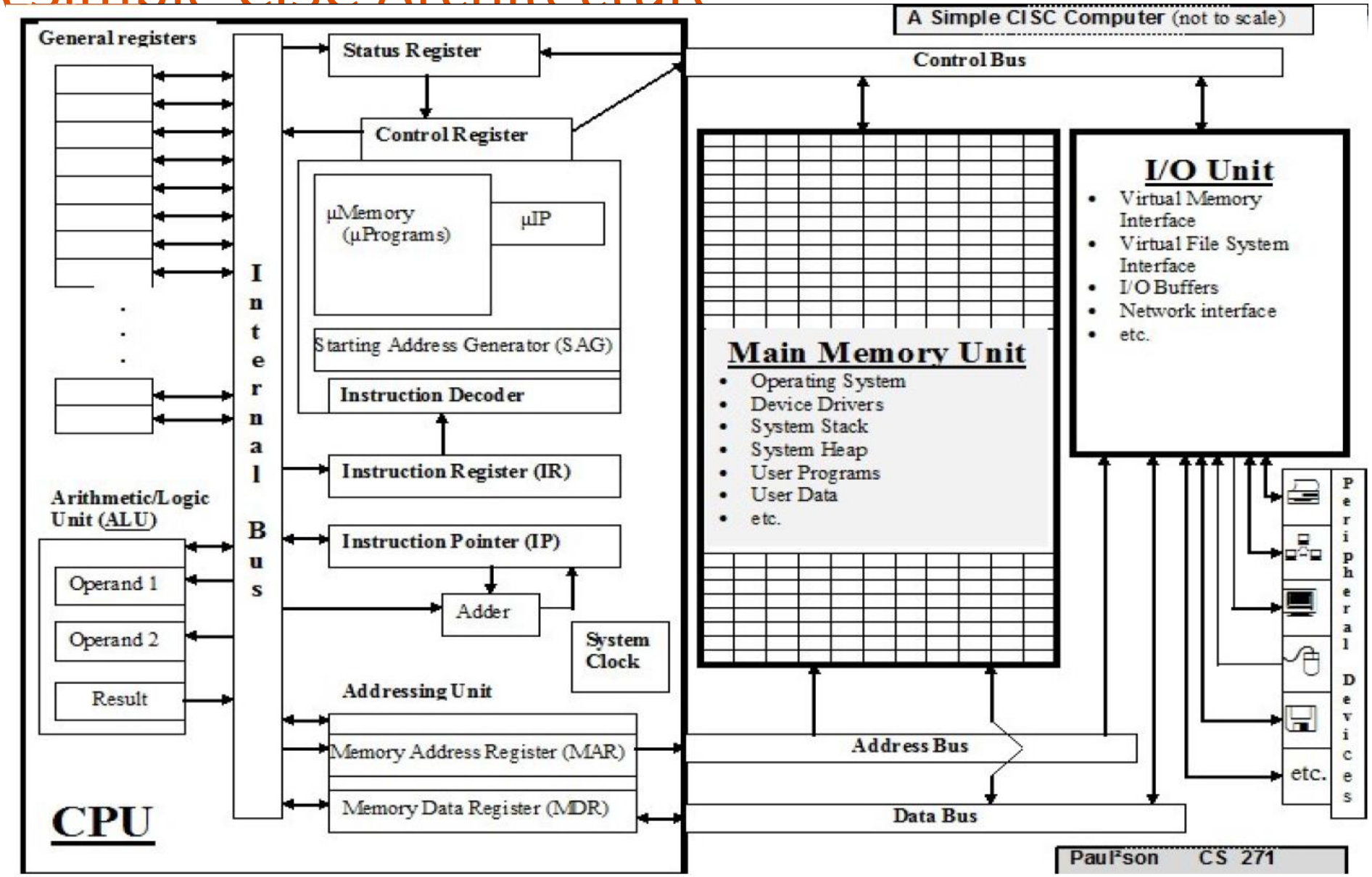
“Growing” Memory From Gates

- Latches → chips
- RAM (Random Access Memory)
  - SRAM (Static RAM)
  - DRAM (Dynamic RAM)
  - SDRAM (Synchronous DRAM)
- ROM (Read-Only Memory)
  - PROM (Programmable ROM)
  - EPROM (Erasable ROM)
  - EEPROM (Electrically EPROM)
- Memory organization
- Memory addressing

Clock Cycles

- So how fast is this?
  - Execution near light-speed
  - Computer can transition to a new “state” at every tick of the system clock
- Clock cycle length determines CPU speed (mostly)
- … but clock cycle length depends on distance between components.

A Simple CISC Architecture



RISC Architecture

Improving CISC

- CISC speed (and convenience) is increased by
  - More efficient microprograms
  - More powerful ISA level instructions
  - Cache memory
  - More registers
  - Wider buses
  - Make it smaller
  - More processors
  - Floating point instructions
  - Etc.

Limitations of CISC

Improving a specific architecture requires instructions to be backward compatible.

So … how about a different architecture?

RISC Machines
Reduced Instruction Set Computer

- Much smaller set of instructions at ISA level
- Instructions are like CISC micro-instructions

- RISC assembly level programs look much longer (more instructions) than CISC assembly level programs, but thy execute faster.
  - Why?

RISC Design Principles

- Instructions executed directly by hardware (no microprograms).
- Maximize rate of fetching instructions.
  - Instruction cache
- Instructions easy to decode
  - Fetching operands, etc.
- Only LOAD and STORE instructions reference memory
- Plenty of registers

More Speed Improvement

- Minimize memory and I/O accesses
  - Cache
  - Separate I/O unit (buffers/processing)
  - Separate network communication unit (NIC)
  - Etc.

- Parallel processing