Bits of Architecture

Introduction to Pipelining

Our Single-Cycle Processor

Branch

Zero

Mux

ALU

4

ALU

PC

Instruction Memory

Registers

RegWrite

Mux

ALU Op

ALU

Mux

MemWrite

Data Memory

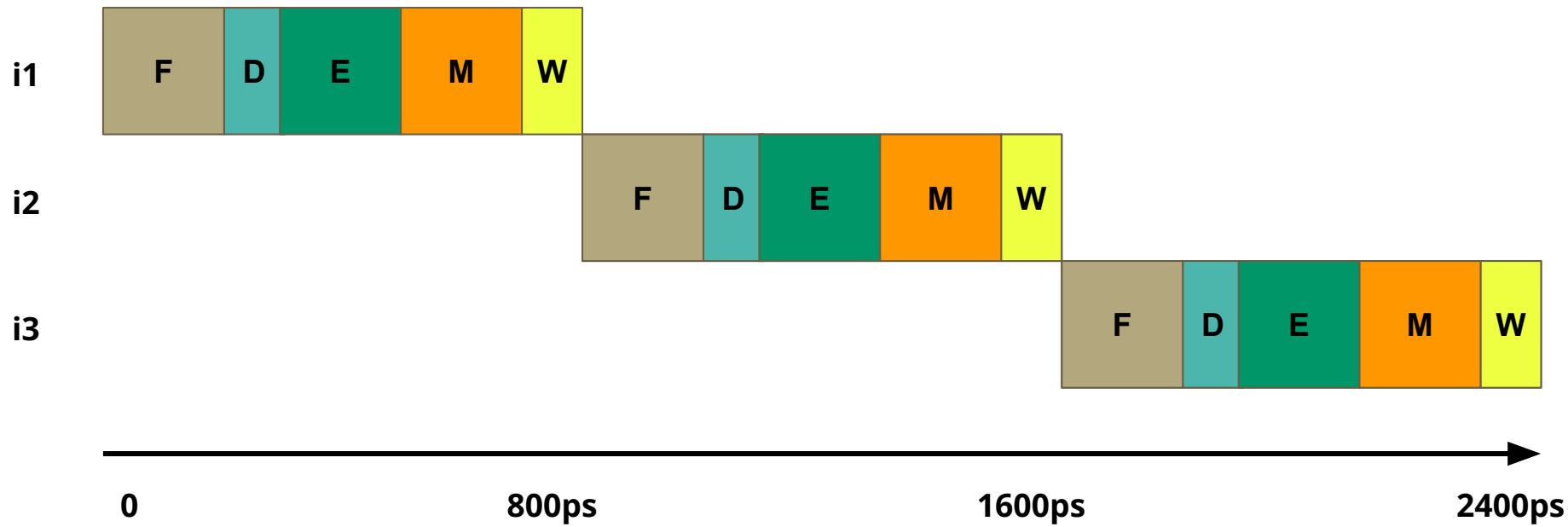**MemRead**

Instruction

Control
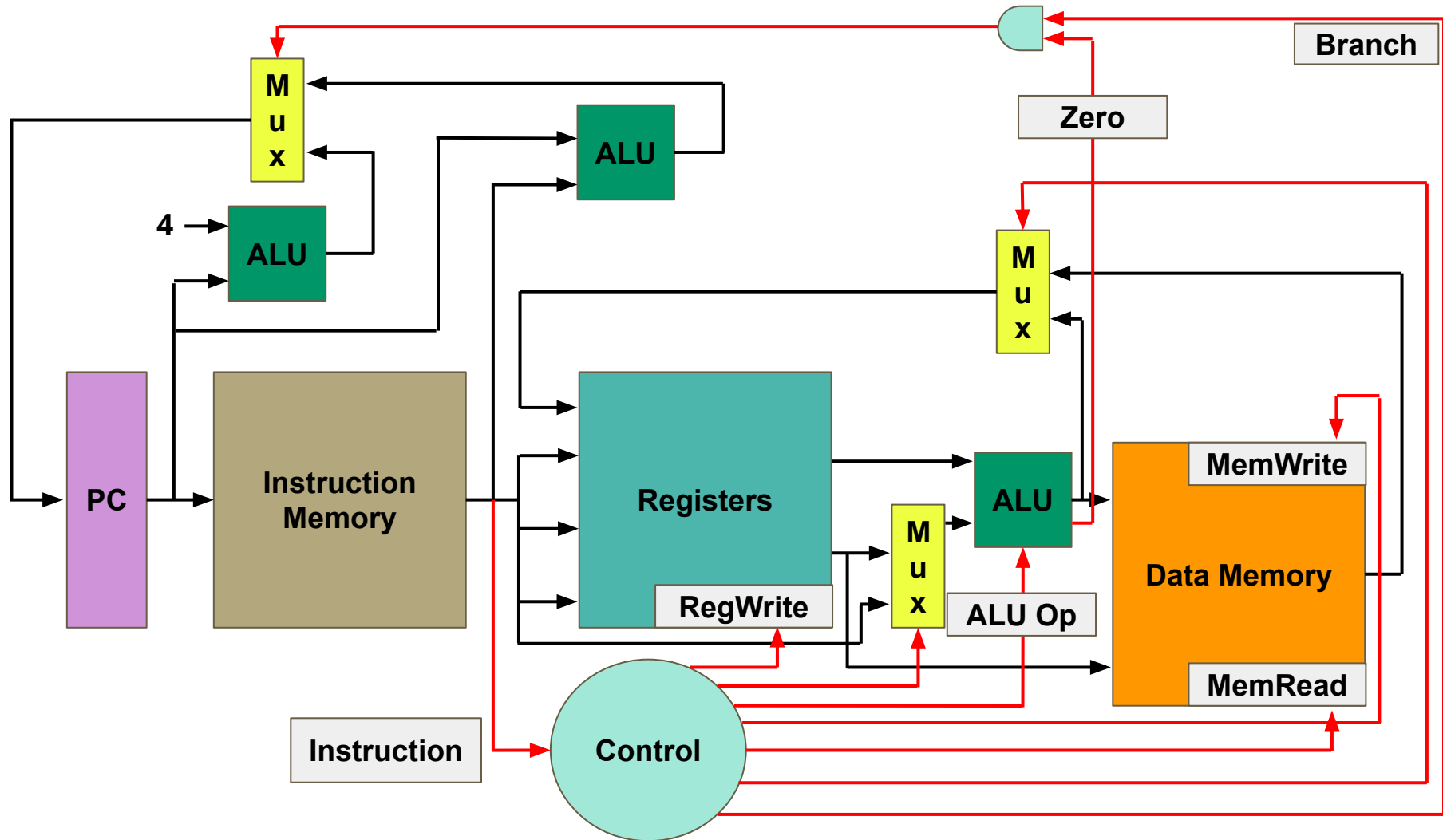
The Problem With Single-Cycle Processors

Single-Cycle Costs

- Our clock rate is based on our **critical path**
- Every instruction takes as long as the critical path
- Critical path = 800ps

| Class	Fetch	Decode	ALU	Data Memory	Register Write	Total
Load	200	100	200	200	100	**800**
Store	200	100	200	200		700
R-Type	200	100	200		100	600
Branch	200	100	200			500

Introduction to Pipelining

Pipelining

- Instead of doing all the work in a single cycle, just do 1 stage of work
- Pros
  - Clock cycle depends on longest stage
  - Overlapping instructions
- Cons
  - More complex core
  - Hazards

All units in ps						
Class	**Fetch**	**Decode**	**ALU**	**Data Memory**	**Register Write**	**Total**
Load	**200**	100	**200**	**200**	100	800
Store	**200**	100	**200**	**200**		700
R-Type	**200**	100	**200**		100	600
Branch	**200**	100	**200**			500

Execution Over Time
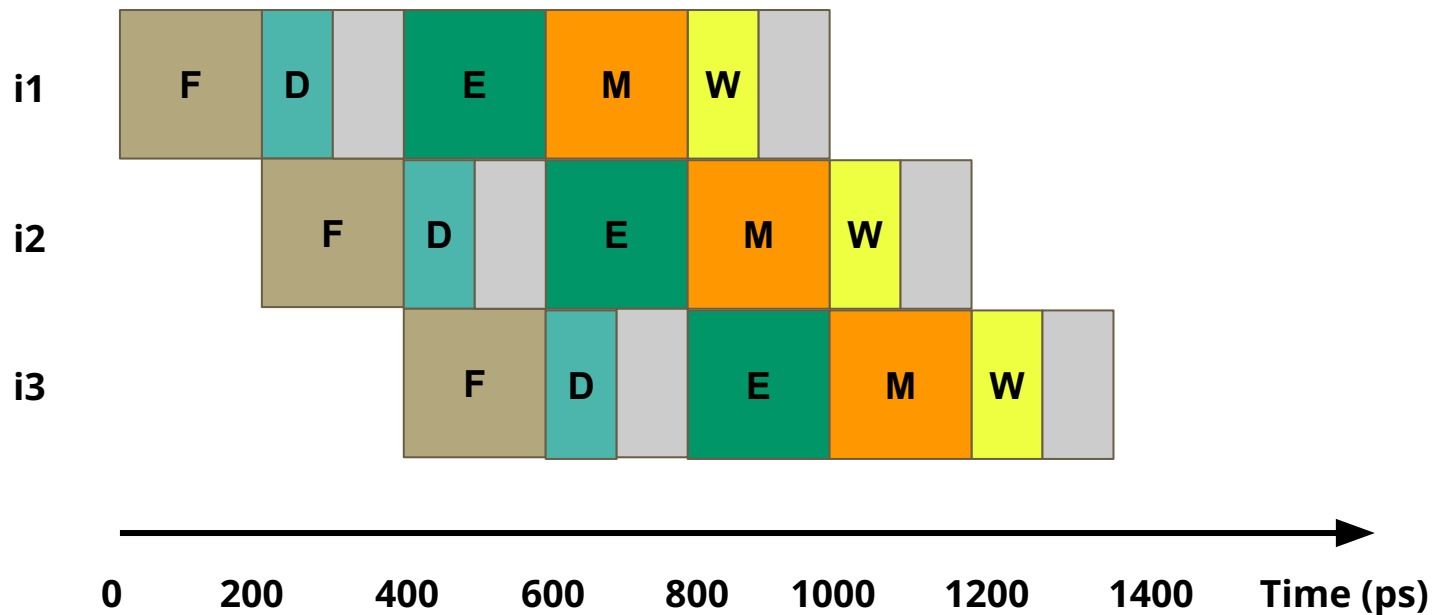
Performance Improvement

- Speedup (ideally) equal to the number of pipeline stages
    - 5 stages = 5x performance
- At steady state...
    - A single-cycle processor is completing 1 instruction every cycle
    - A pipelined processor is completing 1 instruction every cycle
- In our case
    - Single-cycle = 1/800ps
    - Pipelined = 1/200ps
    - Ratio = 800ps/200ps = ~4x speedup