OUT-OF-ORDER LOADS/STORES

Mahdi Nazm Bojnordi

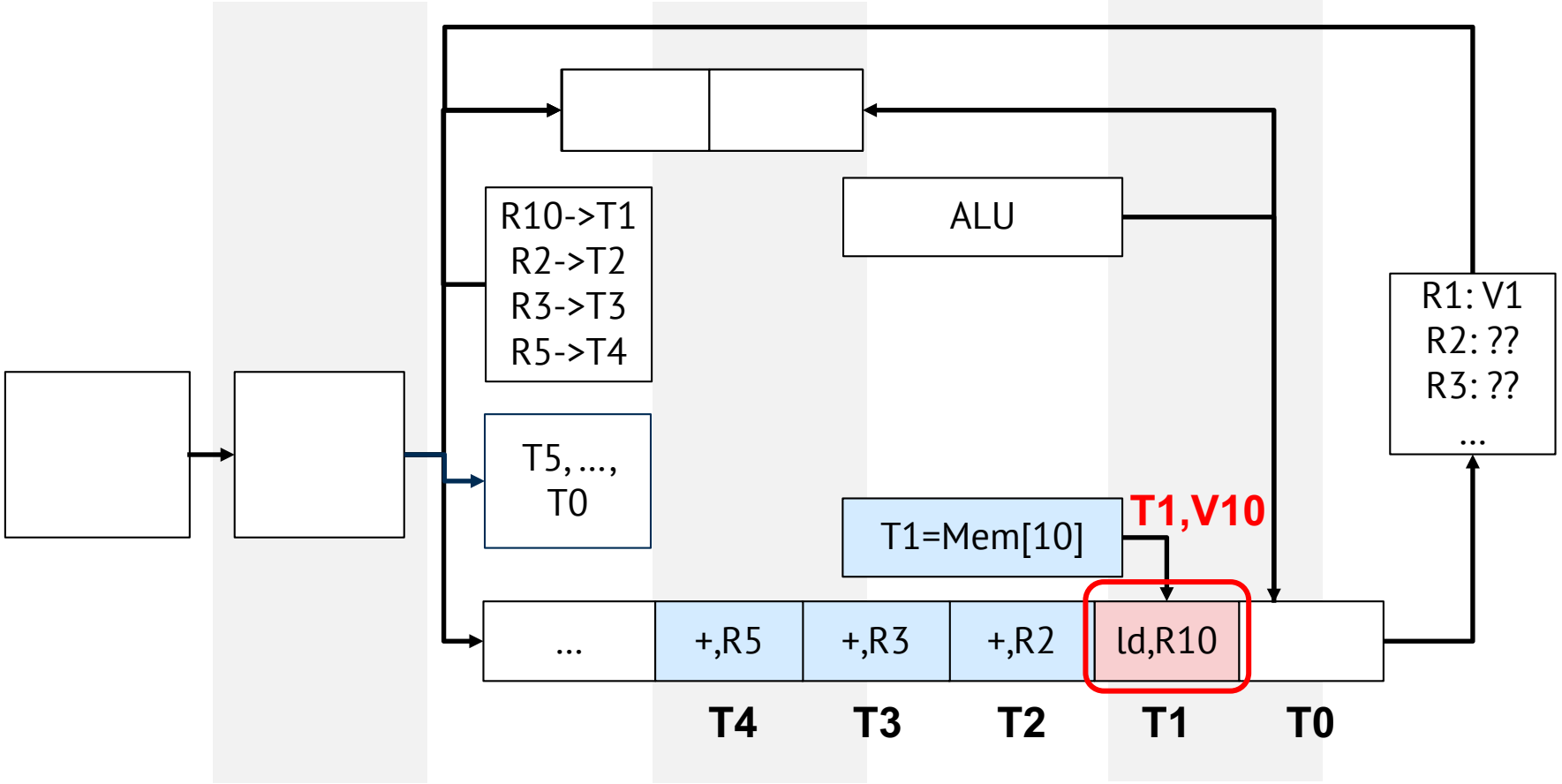
Assistant Professor

School of Computing

University of Utah

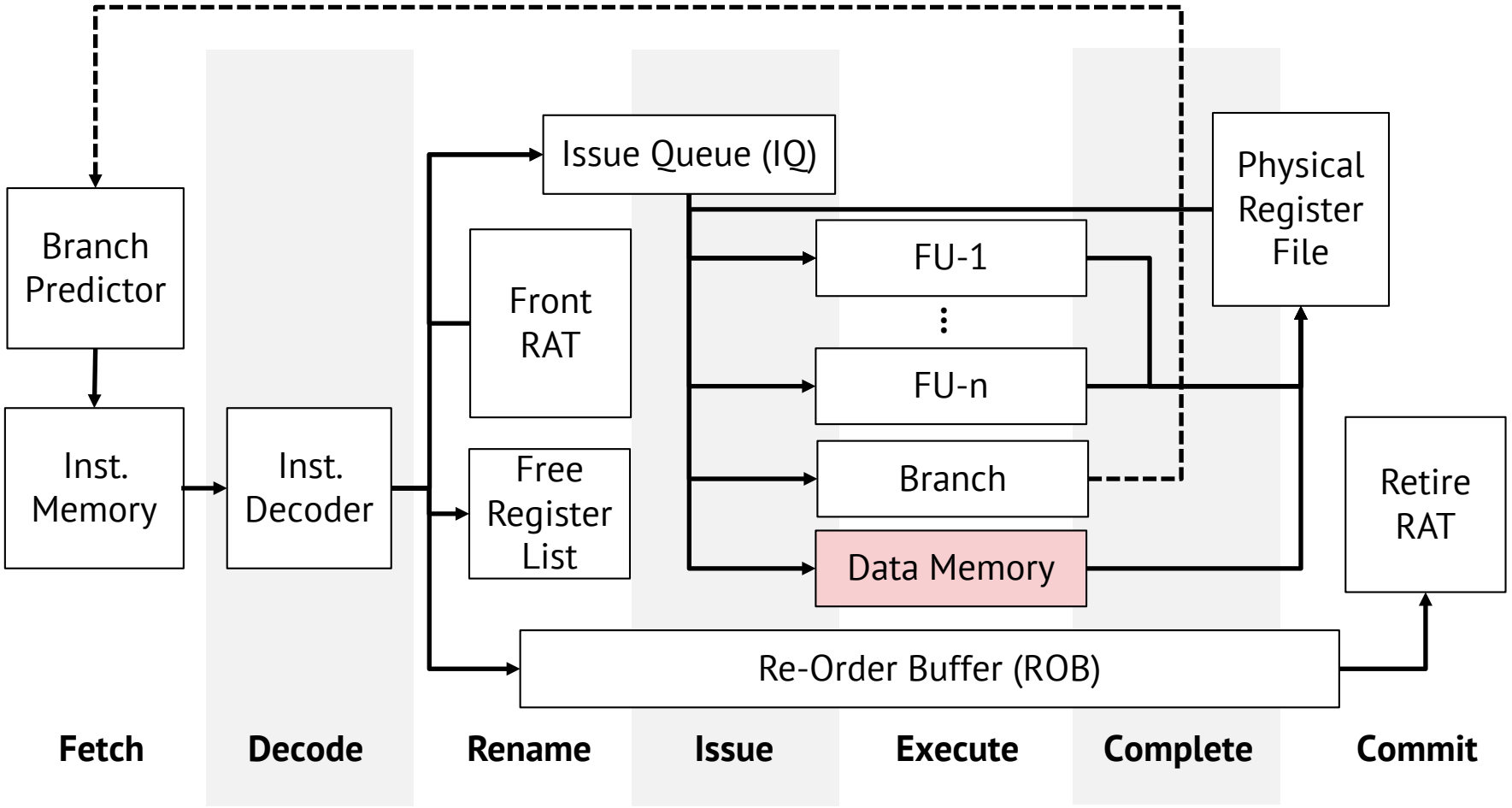
Recall: Out-of-Order Execution

- Memory accesses require long time to complete.



Loads and Stores

- Imagine multiple load/store in IQ.



Memory Data Dependence

- Can we continue executing loads/stores out-of-order?

Instructions in the issue queue

Load	R1←Mem[R2]
Load	R3←Mem[R4+8]
Store	R5→Mem[R6]
Load	R7←Mem[R8+16]
Store	R9→Mem[R10]

Memory

Memory Data Dependence

- Can we continue executing loads/stores out-of-order?
  - Effective address is required for dependence check

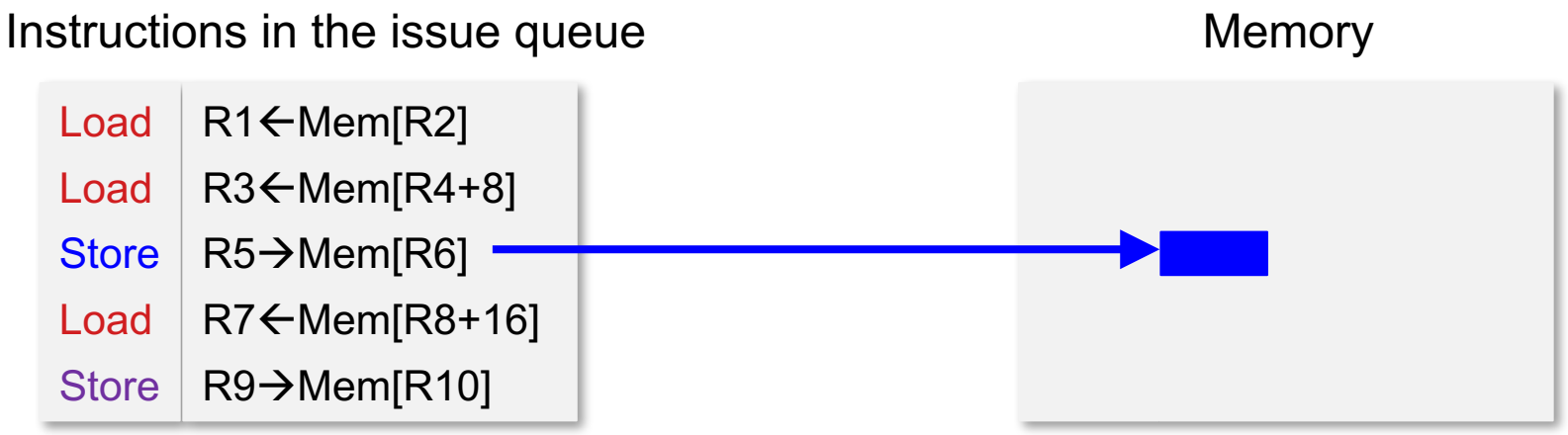
Instructions in the issue queue

Load	R1←Mem[R2]
Load	R3←Mem[R4+8]
Store	R5→Mem[R6]
Load	R7←Mem[R8+16]
Store	R9→Mem[R10]

Memory

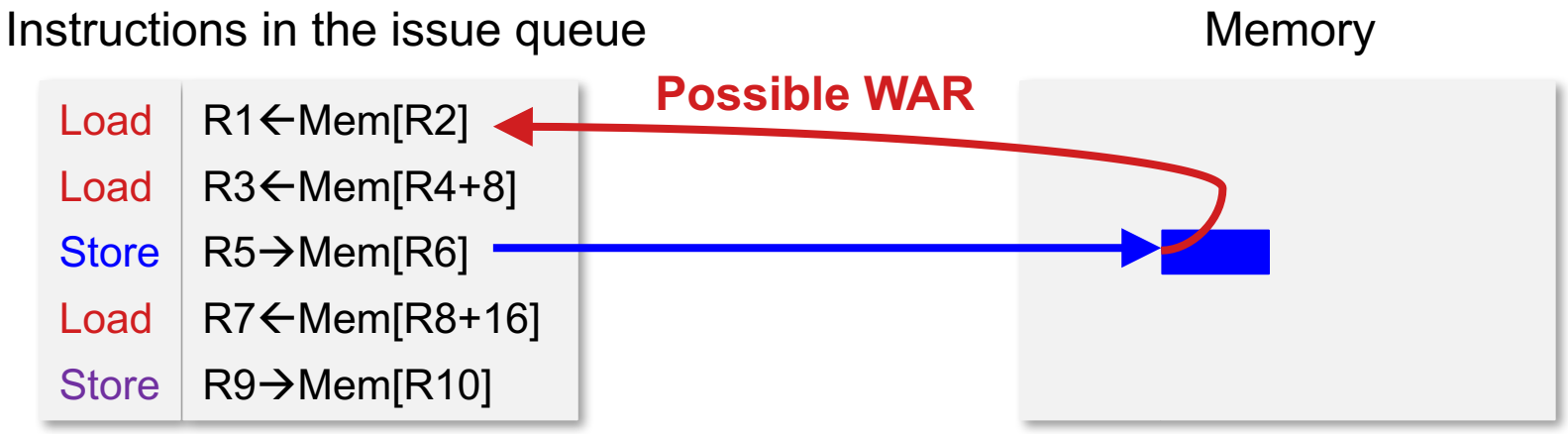
Memory Data Dependence

- Can we continue executing loads/stores out-of-order?
  - Effective address is required for dependence check



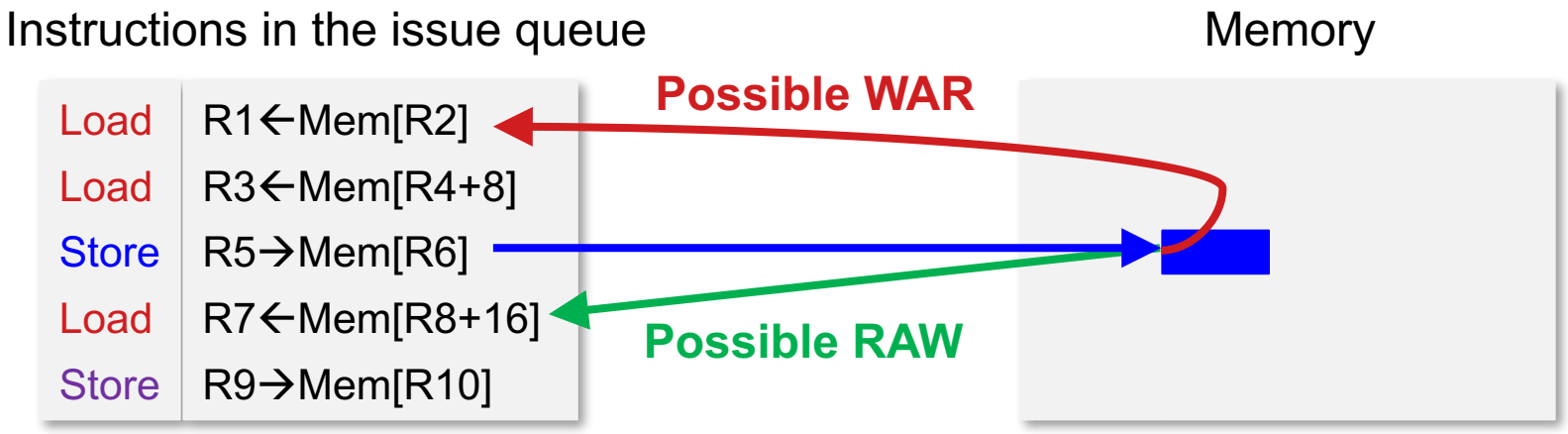
Memory Data Dependence

- Can we continue executing loads/stores out-of-order?
  - Effective address is required for dependence check



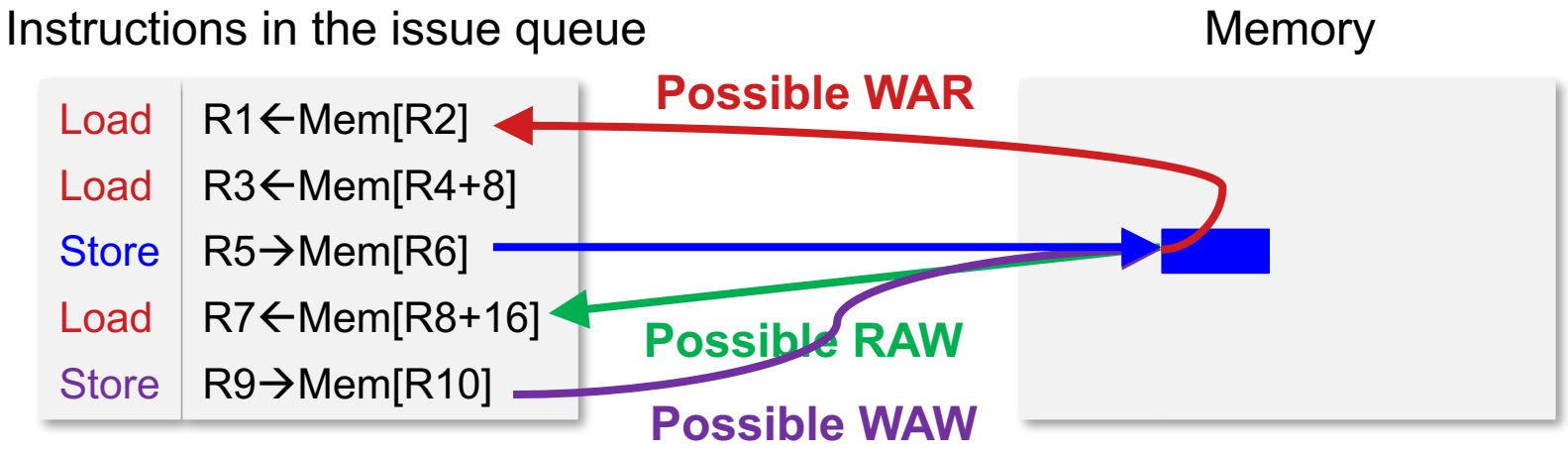
Memory Data Dependence

- Can we continue executing loads/stores out-of-order?
  - Effective address is required for dependence check



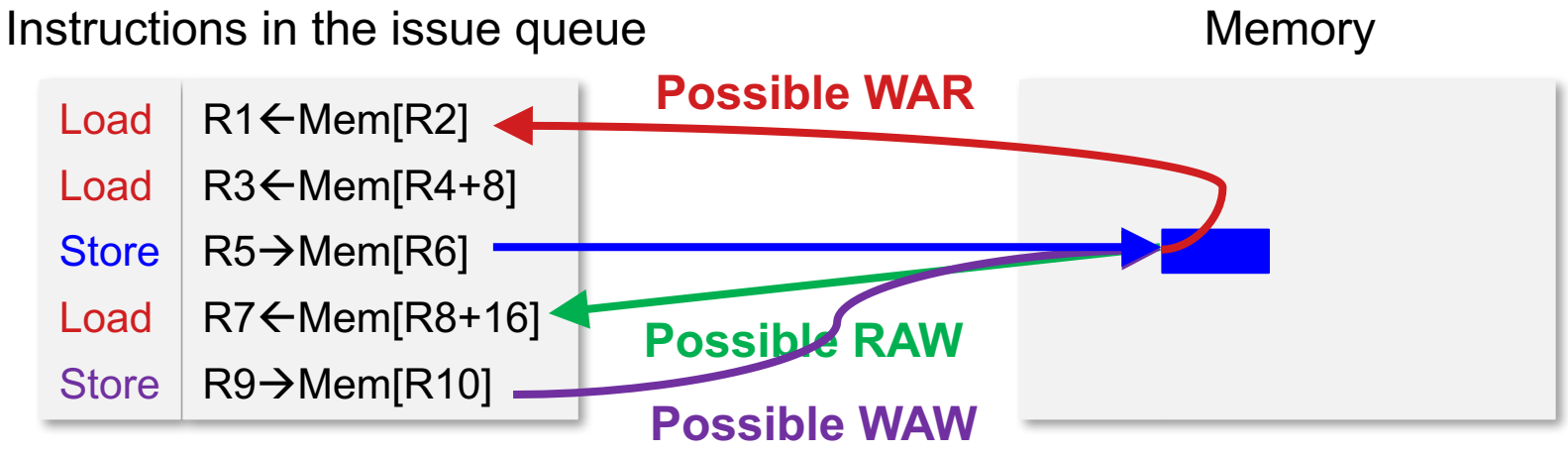
Memory Data Dependence

- Can we continue executing loads/stores out-of-order?
  - Effective address is required for dependence check



Memory Data Dependence

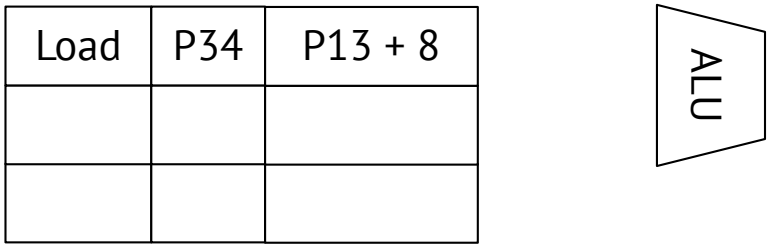
- Can we continue executing loads/stores out-of-order?
  - Effective address is required for dependence check



Does renaming help?

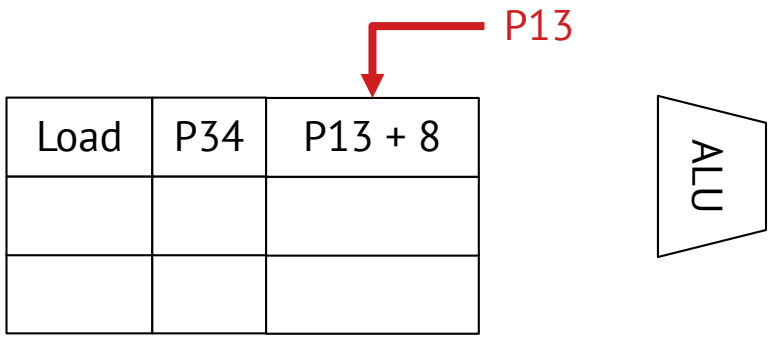
Load-Store Queue

- Dedicated queue only for load/store instructions
  - Check availability of operands every cycle
- Two steps for load/store instructions
  - Compute the effective address when register is available
  - Send the request to memory if there is no memory hazards



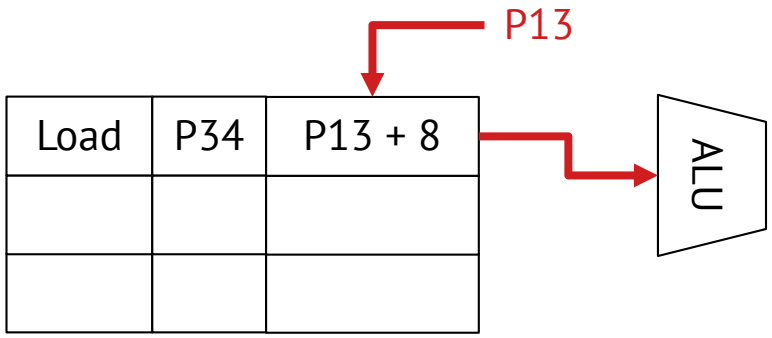
Load-Store Queue

- Dedicated queue only for load/store instructions
  - Check availability of operands every cycle
- Two steps for load/store instructions
  - Compute the effective address when register is available
  - Send the request to memory if there is no memory hazards



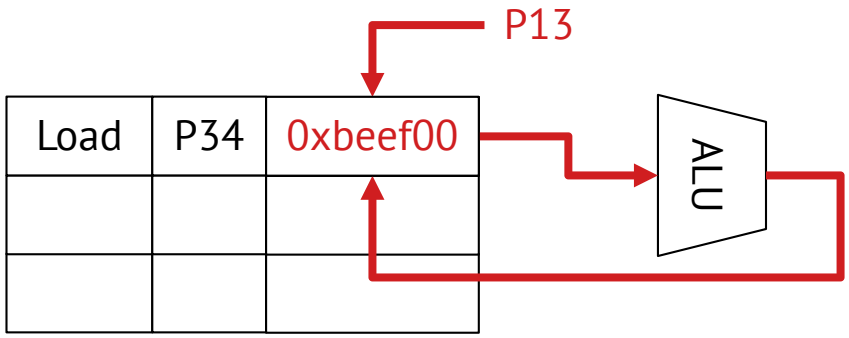
Load-Store Queue

- Dedicated queue only for load/store instructions
  - Check availability of operands every cycle
- Two steps for load/store instructions
  - Compute the effective address when register is available
  - Send the request to memory if there is no memory hazards



Load-Store Queue

- Dedicated queue only for load/store instructions
  - Check availability of operands every cycle
- Two steps for load/store instructions
  - Compute the effective address when register is available
  - Send the request to memory if there is no memory hazards

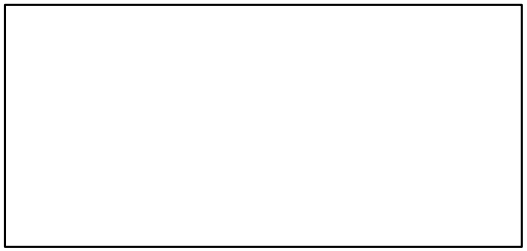


Memory Dependence Check

- Checking for RAW, WAR, and WAW hazards

Load	P34	0x12345
Load	P61	
Store	P26	
Load	P11	
Load	P29	0x12345
Store	P30	0x11111
Load	P15	0x22222
Load	P10	0x11111

1. Which load instructions can be issued?



Memory

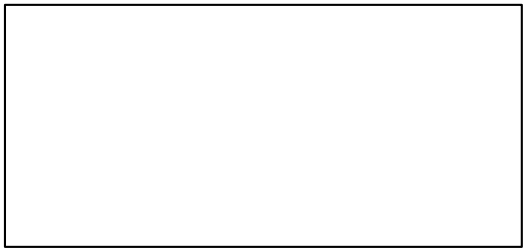
Memory Dependence Check

- Checking for RAW, WAR, and WAW hazards

Load	P34	0x12345
Load	P61	
Store	P26	
Load	P11	
Load	P29	0x12345
Store	P30	0x11111
Load	P15	0x22222
Load	P10	0x11111

1. Which load instructions can be issued?

Due to RAW hazards, only those loads that are not following any unknown stores can be issued.



Memory

Memory Dependence Check

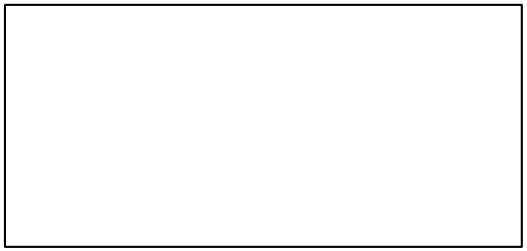
- Checking for RAW, WAR, and WAW hazards

Load	P34	0x12345
Load	P61	
Store	P26	
Load	P11	
Load	P29	0x12345
Store	P30	0x11111
Load	P15	0x22222
Load	P10	0x11111

1. Which load instructions can be issued?

Due to RAW hazards, only those loads that are not following any unknown stores can be issued.

Can we bypass memory?



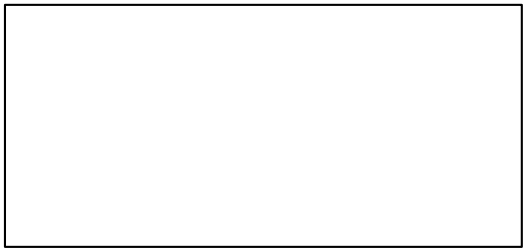
Memory

Memory Dependence Check

- Checking for RAW, WAR, and WAW hazards

Load	P34	0x12345
Load	P61	
Store	P26	
Load	P11	
Load	P29	0x12345
Store	P30	0x11111
Load	P15	0x22222
Load	P10	0x11111

2. Which store instructions can be issued?



Memory

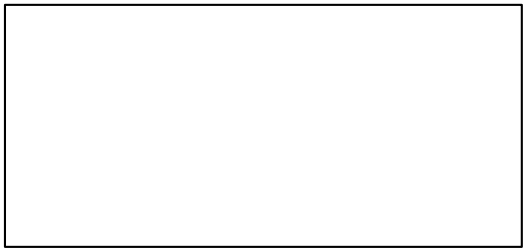
Memory Dependence Check

- Checking for RAW, WAR, and WAW hazards

Load	P34	0x12345
Load	P61	
Store	P26	
Load	P11	
Load	P29	0x12345
Store	P30	0x11111
Load	P15	0x22222
Load	P10	0x11111

2. Which store instructions can be issued?

Due to WAW and WAR hazards, only when there is no older instructions. (why?)



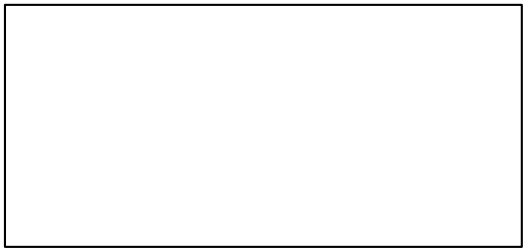
Memory

Memory Dependence Check

- Checking for RAW, WAR, and WAW hazards

Load	P34	0x12345
Load	P61	
Store	P26	0x22222
Load	P11	
Load	P29	0x12345
Store	P30	0x11111
Load	P15	0x22222
Load	P10	0x11111

Which instructions can be issued?



Memory

Memory Dependence Prediction

- Can we predict memory dependence?

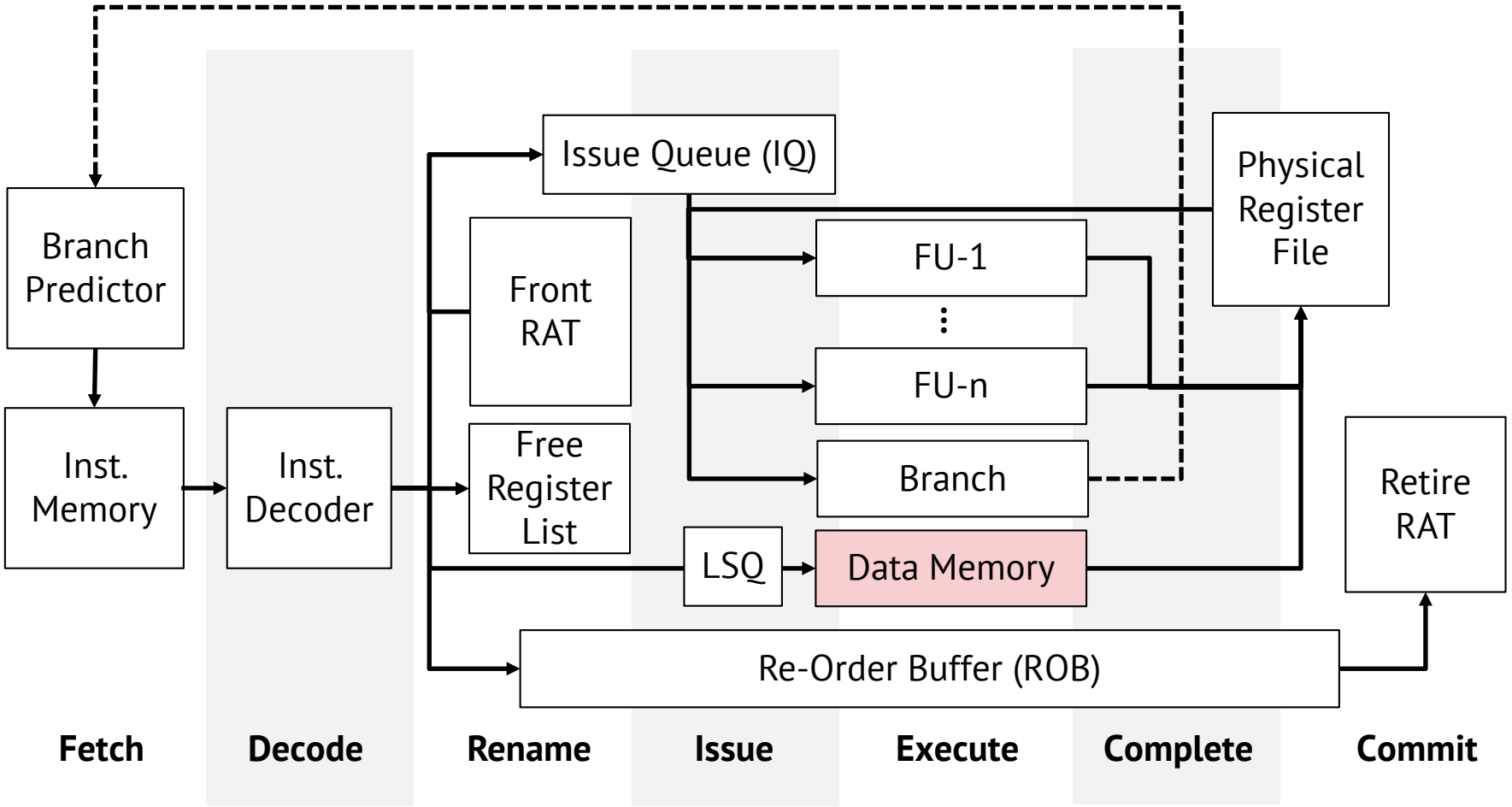
Load	P34	0x12345
Load	P61	
Store	P26	
Load	P11	
Load	P29	0x12345
Store	P30	0x11111
Load	P15	0x22222
Load	P10	0x11111

Issue/execute load instructions even if they are following unresolved stores

What if the prediction was not correct?

Out-of-order Pipeline with LSQ

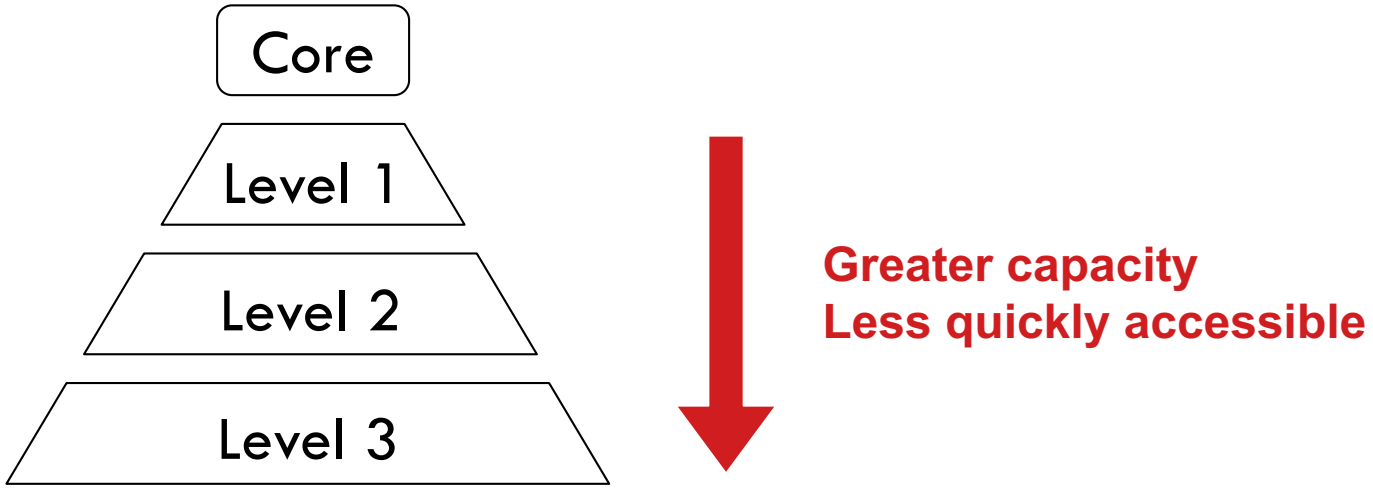
- LSQ is an extension to IQ



Memory Hierarchy

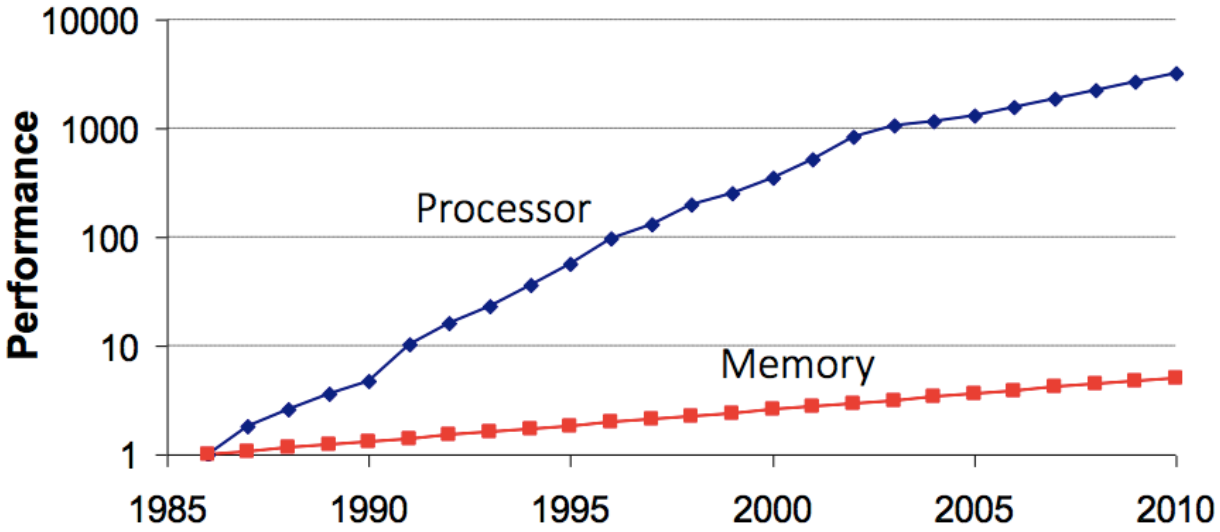
"Ideally one would desire an indefinitely large memory capacity such that any particular [...] word would be immediately available [...] We are [...] forced to recognize the possibility of constructing a hierarchy of memories, each of which has greater capacity than the preceding but which is less quickly accessible."

-- Burks, Goldstine, and von Neumann, 1946



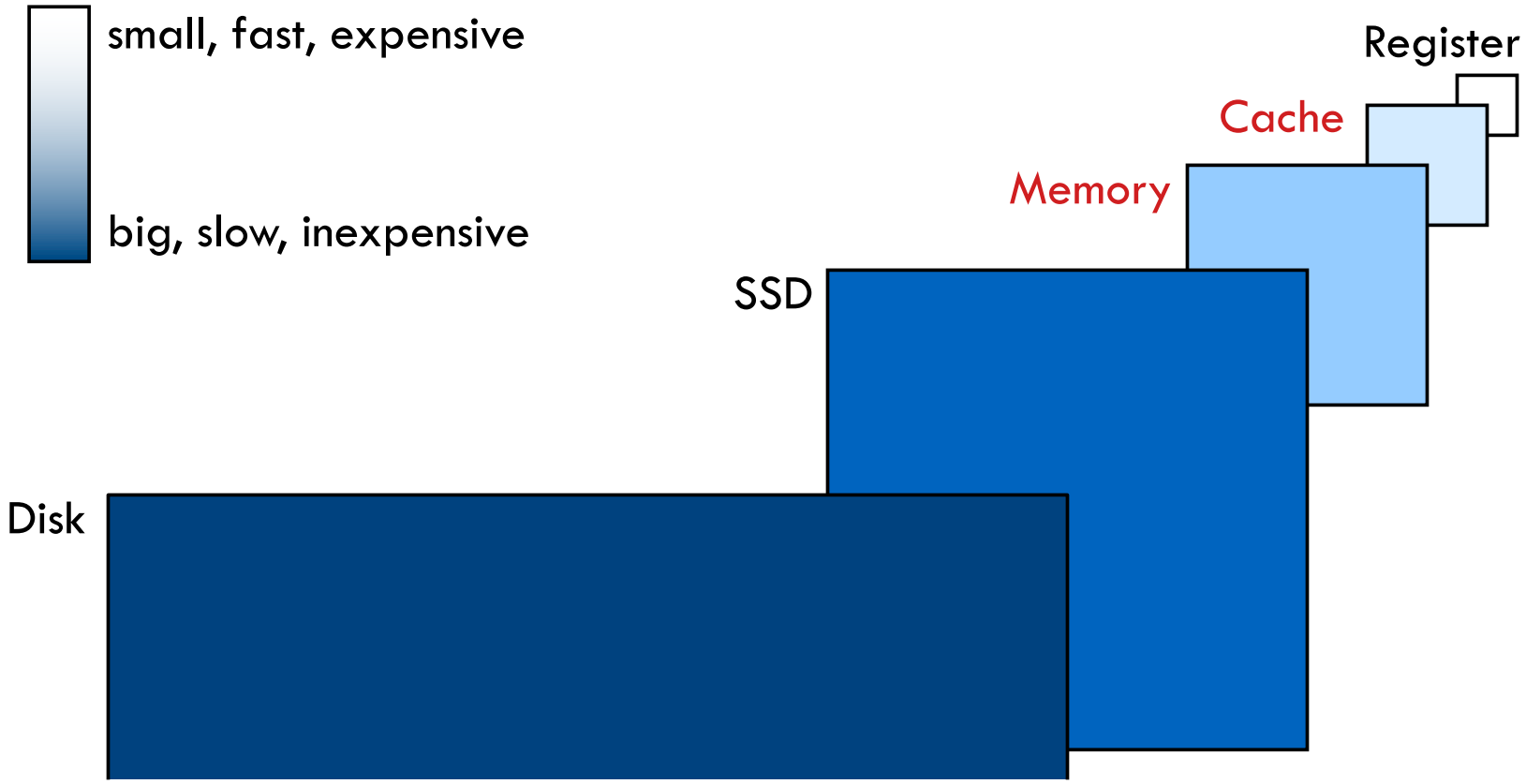
The Memory Wall

- Processor-memory performance gap increased over 50% per year
  - Processor performance historically improved ~60% per year
  - Main memory access time improves ~5% per year



Modern Memory Hierarchy

- Trade-off among memory speed, capacity, and cost

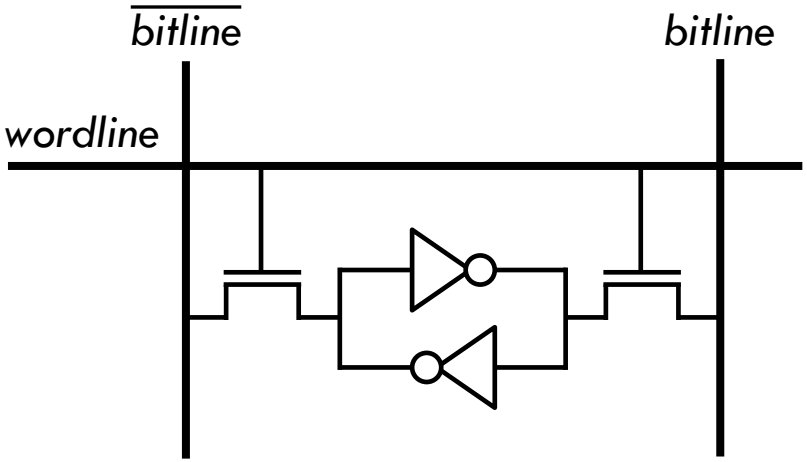


Memory Technology

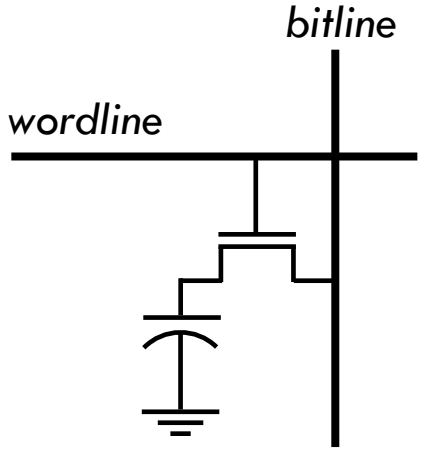
- Random access memory (RAM) technology
  - access time same for all locations (not so true anymore)
  - Static RAM (SRAM)
    - typically used for caches
    - 6T/bit; fast but – low density, high power, expensive
  - Dynamic RAM (DRAM)
    - typically used for main memory
    - 1T/bit; inexpensive, high density, low power – but slow

RAM Cells

- 6T SRAM cell
  - internal feedback maintains data while power on



- 1T-1C DRAM cell
  - needs refresh regularly to preserve data



Processor Cache

- Occupies a large fraction of die area in modern microprocessors

3-3.5 GHz
~$1000 2014)

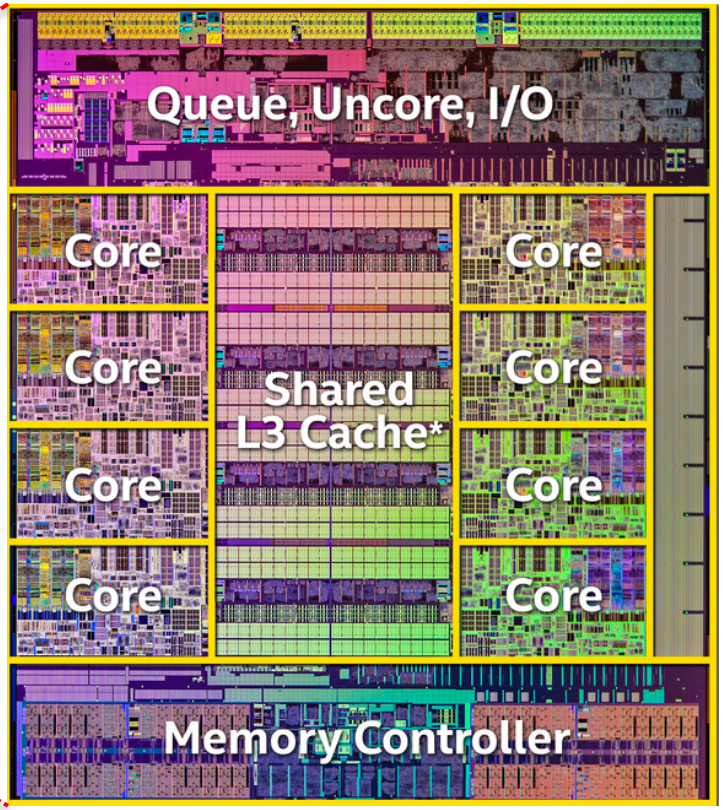
Source: Intel Core i7

Processor Cache

- Occupies a large fraction of die area in modern microprocessors

3-3.5 GHz
~$1000 2014)

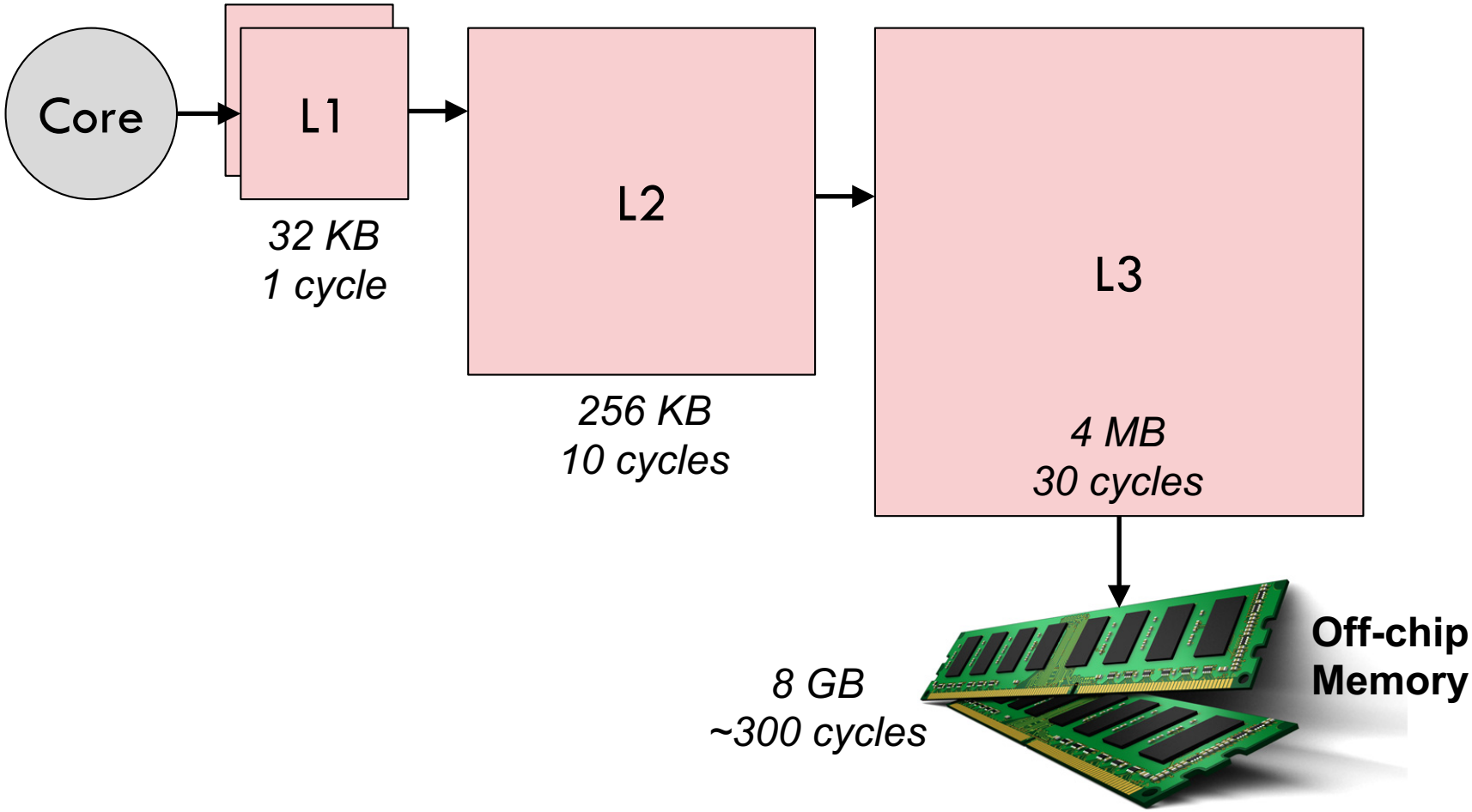
20MB of cache



Source: Intel Core i7

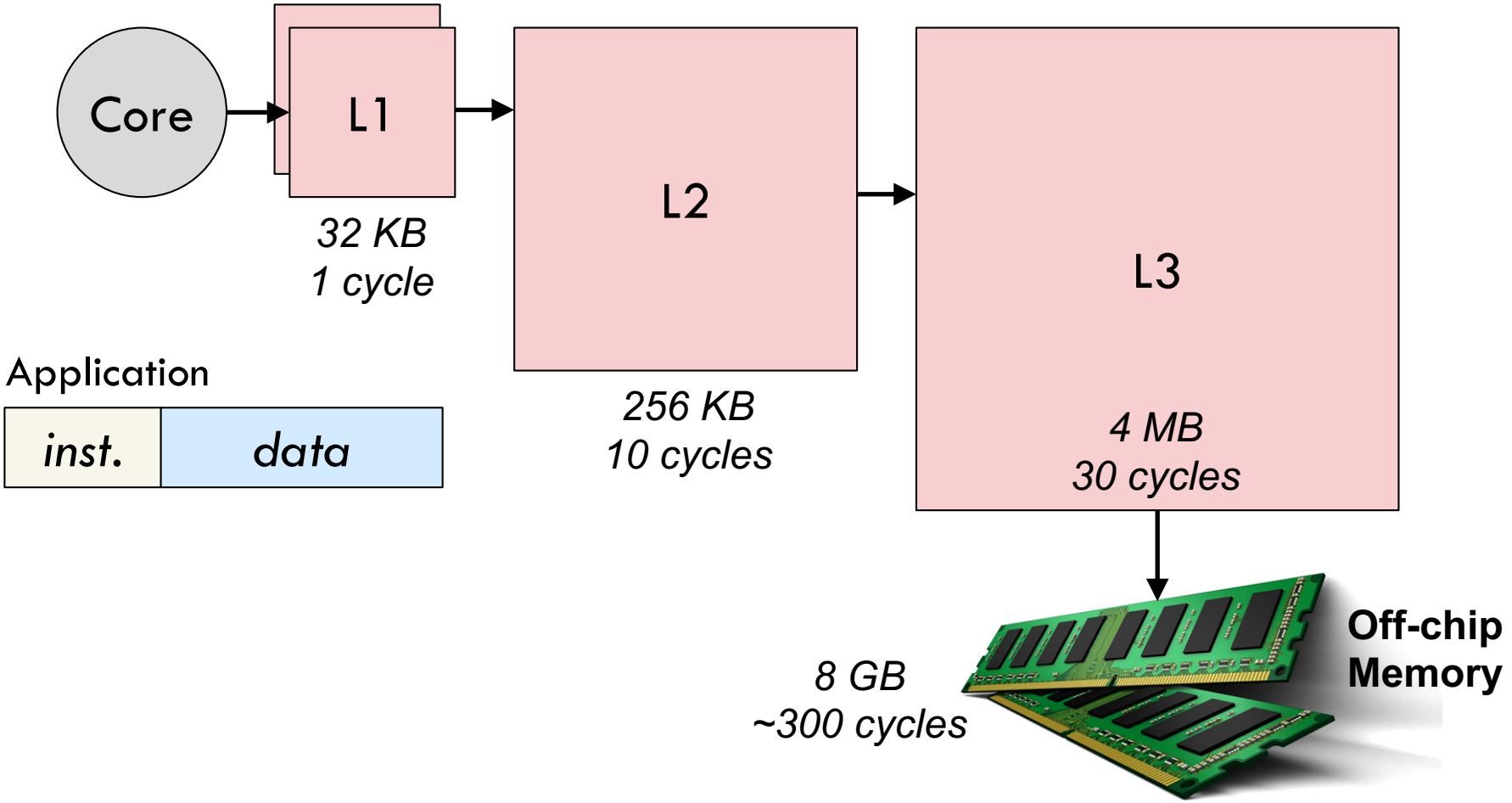
Cache Hierarchy

- Example three-level cache organization



Cache Hierarchy

- Example three-level cache organization



Cache Hierarchy

- Example three-level cache organization