

STAM Center  
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU  
Arizona State University  
The Fulton School of  
Engineering

# CSE 520 Computer Architecture II

## Cache Coherence

Prof. Michel A. Kinsy

1

---

---

---

---

---

---

---

---

STAM Center  
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU  
Arizona State University  
The Fulton School of  
Engineering

## Architecture Taxonomy

Processor Organizations

- Single instruction, single data stream (SISD)
  - Uniprocessor
- Single instruction multiple data stream (SIMD)
  - Vector Processor
  - Array Processor
- Multiple instruction, single data stream (MISD)
  - Shared Memory (Tightly Coupled)
    - Symmetric Multiprocessor (SMP)
    - Nonuniform Memory Access (NUMA)
- Multiple instruction, multiple data stream (MIMD)
  - Distributed Memory (Loosely Coupled)
    - Cluster

Parallelism Paradigms: Instruction level, Data level and Task level Parallelisms

2

---

---

---

---

---

---

---

---

STAM Center  
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU  
Arizona State University  
The Fulton School of  
Engineering

## Architecture Taxonomy

Processor Organizations

- Single instruction, single data stream (SISD)
  - Uniprocessor

Parallelism Paradigms: Instruction level, Data level and Task level Parallelisms

3

---

---

---

---

---

---

---

---

**STAM Center** SECURE, TRUSTED, AND ASSURED MICROELECTRONICS **ASU Engineering** Arizona State University

### Parallel Architectures

- Single instruction, single data stream – SISD
  - Single processor
  - Single instruction stream
  - Data stored in single memory

```
graph LR; I[Instruction] -- Inst. Stream --> P[Processor]; P <--> |Data Stream| D[Data]
```

4

---

---

---

---

---

---

---

---

**STAM Center** SECURE, TRUSTED, AND ASSURED MICROELECTRONICS **ASU Engineering** Arizona State University

### Architecture Taxonomy

Processor Organizations

```
graph TD; A[Single instruction, single data stream (SISD)] --> B[Uniprocessor];
```

Presented RISC-V Architecture

Parallelism Paradigms: Instruction level, Data level and Task level Parallelisms

5

---

---

---

---

---

---

---

---

**STAM Center** SECURE, TRUSTED, AND ASSURED MICROELECTRONICS **ASU Engineering** Arizona State University

### Architecture Taxonomy

Processor Organizations

```
graph TD; A[Single instruction, single data stream (SISD)] --> B[Uniprocessor]; C[Single instruction multiple data stream (SIMD)] --> D[Vector Processor]; C --> E[Array Processor];
```

Parallelism Paradigms: Instruction level, Data level and Task level Parallelisms

6

---

---

---

---

---

---

---

---

**STAM Center** SECURE, TRUSTED, AND ASSURED MICROELECTRONICS **ASU Engineering** The Future School of Arizona State University

### Parallel Architectures

- Single instruction, multiple data stream – SIMD
- Single machine instruction
  - Each instruction executed on different set of data by different processors
- Number of processing elements
  - Machine controls simultaneous execution
    - Lockstep basis
  - Each processing element has associated data memory
- Application: Vector and array processing

7

---

---

---

---

---

---

---

---

**STAM Center** SECURE, TRUSTED, AND ASSURED MICROELECTRONICS **ASU Engineering** The Future School of Arizona State University

### Parallel Architectures

- Single instruction, multiple data stream – SIMD

```

    graph LR
      Inst[Instruction] -- Inst. Stream --> P1[Processor]
      Inst -- Inst. Stream --> P2[Processor]
      P1 <--> DS1[Data Stream]
      P2 <--> DS2[Data Stream]
      DS1 <--> Data[Data]
      DS2 <--> Data
  
```

8

---

---

---

---

---

---

---

---

**STAM Center** SECURE, TRUSTED, AND ASSURED MICROELECTRONICS **ASU Engineering** The Future School of Arizona State University

### Architecture Taxonomy

```

    graph TD
      PO[Processor Organizations] --> SISD[Single Instruction, single data stream (SISD)]
      PO --> SIMD[Single Instruction multiple data stream (SIMD)]
      PO --> MISD[Multiple Instruction, single data stream (MISD)]
      SISD --> UP[Uniprocessor]
      SIMD --> VP[Vector Processor]
      SIMD --> AP1[Array Processor]
      MISD --> AP2[Array Processor]
  
```

**Parallelism Paradigms: Instruction level, Data level and Task level Parallelisms**

9

---

---

---

---

---

---

---

---

**STAM Center** SECURE, TRUSTED, AND ASSURED MICROELECTRONICS **ASU Engineering** The Future School of Arizona State University

### Parallel Architectures

- Multiple instruction, single data stream – MISD
  - Sequence of data
  - Transmitted to set of processors
  - Each processor executes different instruction sequence
- Do not know any implemented case

10

---

---

---

---

---

---

---

---

**STAM Center** SECURE, TRUSTED, AND ASSURED MICROELECTRONICS **ASU Engineering** The Future School of Arizona State University

### Parallel Architectures

- Multiple instruction, single data stream – MISD

```

    graph LR
      I1[Instruction] -- Inst. Stream --> P1[Processor]
      I2[Instruction] -- Inst. Stream --> P2[Processor]
      P1 -- Data Stream --> D[Data]
      P2 -- Data Stream --> D
  
```

11

---

---

---

---

---

---

---

---

**STAM Center** SECURE, TRUSTED, AND ASSURED MICROELECTRONICS **ASU Engineering** The Future School of Arizona State University

### Architecture Taxonomy

Processor Organizations

- Single instruction, single data stream (SISD)
  - Uniprocessor
- Single instruction multiple data stream (SIMD)
  - Vector Processor
  - Array Processor
- Multiple instruction, single data stream (MISD)
- Multiple instruction, multiple data stream (MIMD)

Parallelism Paradigms: Instruction level, Data level and Task level Parallelisms

12

---

---

---

---

---

---

---

---

**STAM Center** SECURE, TRUSTED, AND ASSURED MICROELECTRONICS **ASU Engineering** The Future School of Arizona State University

### Parallel Architectures

- Multiple instruction, multiple data stream- MIMD
  - Set of processors
  - Simultaneously executes different instruction sequences
  - Different sets of data
- Examples: SMPs, NUMA systems, and Clusters

13

---

---

---

---

---

---

---

---

**STAM Center** SECURE, TRUSTED, AND ASSURED MICROELECTRONICS **ASU Engineering** The Future School of Arizona State University

### Parallel Architectures

- Multiple instruction, multiple data stream- MIMD

14

---

---

---

---

---

---

---

---

**STAM Center** SECURE, TRUSTED, AND ASSURED MICROELECTRONICS **ASU Engineering** The Future School of Arizona State University

### Architecture Taxonomy

Processor Organizations

- Single Instruction, single data stream (SISD)
  - Uniprocessor
  - Vector Processor
  - Array Processor
- Single Instruction multiple data stream (SIMD)
  - Shared Memory (Tightly Coupled)
    - Symmetric Multiprocessor (SMP)
    - Nonuniform Memory Access (NUMA)
  - Distributed Memory (Loosely Coupled)
    - Cluster
- Multiple Instruction, single data stream (MISD)
- Multiple Instruction, multiple data stream (MIMD)

Presented RISC-V Architecture

Parallelism Paradigms: Instruction level, Data level and Task level Parallelisms

15

---

---

---

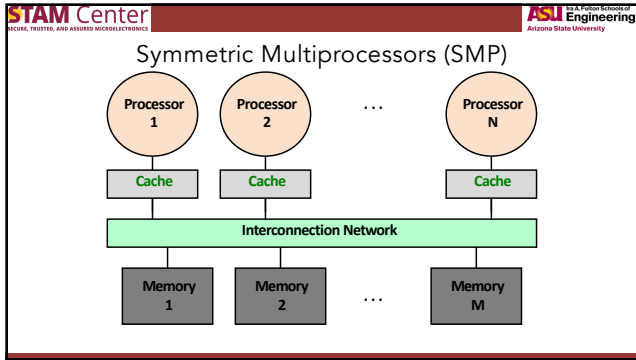
---

---

---

---

---



16

---

---

---

---

---

---

---

---

**Symmetric Multiprocessors (SMP)**

- A collection of processors and a collection of memory connected through some interconnect
- Symmetric because latency for any processor to access any memory is constant – uniform memory access (UMA)

The diagram below the text shows the same SMP architecture as slide 16, with processors, caches, an interconnection network, and memory.

17

---

---

---

---

---

---

---

---

**Shared Memory Architectures**

- Key differentiating feature: the address space is shared, i.e., any processor can directly address any memory location and access them with load/store instructions

The diagram below the text shows the same SMP architecture as slide 16, but with a red dashed box around the memory units (Memory 1, Memory 2, ..., Memory M) and the label 'Memory' next to it, indicating that the memory is shared.

18

---

---

---

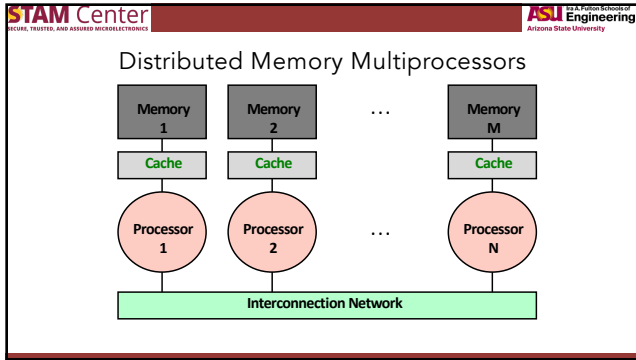
---

---

---

---

---



19

---

---

---

---

---

---

---

---

**Distributed Memory Multiprocessors**

- Each processor has local memory that is accessible through a fast interconnect
- The different nodes are connected as I/O devices with (potentially) slower interconnect
- Local memory access is a lot faster than remote memory
  - Nonuniform memory access (NUMA)

The slide includes a small diagram of the distributed memory multiprocessor architecture, similar to the one in slide 19.

20

---

---

---

---

---

---

---

---

**Parallel Architectures**

- The parallel computers are classified as
  - Shared memory
  - Distributed memory
- Both shared and distributed memory systems have:
  - Processors: now generally commodity processors
  - Memory: now general commodity DRAM/DDR
  - Network/interconnect: between the processors or memory

21

---

---

---

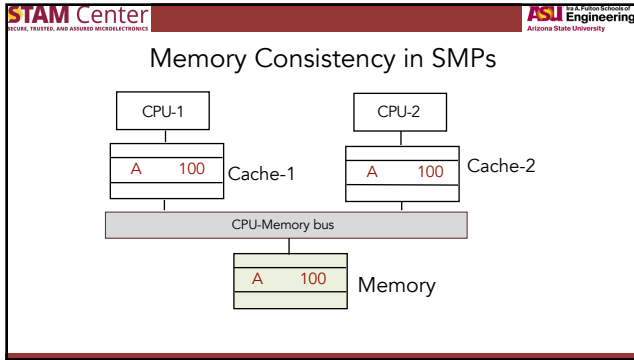
---

---

---

---

---



22

---

---

---

---

---

---

---

---

### Cache Reads & Writes

- Cache reads
  - With a hit, the data can be retrieved
  - On a miss, we need to take a set of actions
- Cache writes
  - On a hit
    - Write-Through: Every time we write to a variable in cache, we also need to update that variable in memory
    - Write-Back: When we write to a variable in the cache, the modified copy is written back to memory when it gets evicted or replaced in the cache.
      - Write-back can improve performance, but is more complex to implement
        - Dirty Bit: In a write-back cache, cache blocks have a dirty bit
        - Block Replacement: Block is written back to memory if it is dirty and replaced with a new memory block

23

---

---

---

---

---

---

---

---

### Cache Reads & Writes

- Cache reads
  - With a hit, the data can be retrieved
  - On a miss, we need to take a set of actions
- Cache writes
  - On a miss
    - Write Allocate - i.e., Fetch on Write
      - Load block into memory as on a read miss
      - Perform Write Hit actions
    - No Write Allocate - i.e., Write Around
      - Modify block in lower memory directly without loading into cache
  - Relation of write miss policies to write hit policies
    - Common pairings
      - Write Through with No Write Allocate
      - Write Back with Write Allocate

24

---

---

---

---

---

---

---

---

**STAM Center** SECURE, TRUSTED, AND ASSURED MICROELECTRONICS **ASU Engineering** Arizona State University

### Memory Consistency in SMPs

- Suppose CPU-1 updates A to 200
  - Write-back: memory and cache-2 have stale values
  - Write-through: cache-2 has a stale value

25

---

---

---

---

---

---

---

---

**STAM Center** SECURE, TRUSTED, AND ASSURED MICROELECTRONICS **ASU Engineering** Arizona State University

### Memory Consistency in SMPs

- Suppose CPU-1 updates A to 200
  - Do these stale values matter?
  - What is the view of shared memory for programming?

26

---

---

---

---

---

---

---

---

**STAM Center** SECURE, TRUSTED, AND ASSURED MICROELECTRONICS **ASU Engineering** Arizona State University

### Write-through Caches & Sequential Consistency

- T1 is executed
 

Cache-1
X=0
Y=10

Memory
X=0
Y=10
X=
Y=

Cache-2
Y=
X=
X=

Prog T1
ST X, R1
ST Y, R1

Prog T2
LD Y, R1
ST Y', R1
LD X, R2
ST X', R2

27

---

---

---

---

---

---

---

---

**STAM Center** SECURE, TRUSTED, AND ASSURED MICROELECTRONICS **ASU Engineering** The Future School of Arizona State University

### Write-through Caches & Sequential Consistency

- T1 is executed
- Cache-1 writes back Y
- T2 executed
- Cache-1 writes back X

	Cache-1	Memory	Cache-2	Prog T2
Initial	X=0 Y=10	X=0 Y=10	Y=- X=-	LD Y, R1
T1 executed	X=1 Y=11	X=0 Y=10	Y=- X=-	ST Y, R1
Cache-1 writes back Y	X=1 Y=11	X=0 Y=11	Y=- X=-	LD X, R2
T2 executed	X=1 Y=11	X=0 Y=11	Y=11 X=0	ST X, R2
Cache-1 writes back X	X=1 Y=11	X=1 Y=11	Y=11 X=0	

28

---

---

---

---

---

---

---

---

**STAM Center** SECURE, TRUSTED, AND ASSURED MICROELECTRONICS **ASU Engineering** The Future School of Arizona State University

### Write-through Caches & Sequential Consistency

- T1 is executed
- Cache-1 writes back Y
- T2 executed
- Cache-1 writes back X
- Cache-2 writes back X' & Y'

	Cache-1	Memory	Cache-2	Prog T2
Initial	X=0 Y=10	X=0 Y=10	Y=- X=-	LD Y, R1
T1 executed	X=1 Y=11	X=0 Y=10	Y=- X=-	ST Y, R1
Cache-1 writes back Y	X=1 Y=11	X=0 Y=11	Y=- X=-	LD X, R2
T2 executed	X=1 Y=11	X=1 Y=11	Y=11 X=0	ST X, R2
Cache-1 writes back X	X=1 Y=11	X=1 Y=11	Y=11 X=0	
Cache-2 writes back X' & Y'	X=1 Y=11	X=1 Y=11	Y=11 X=0	

Incoherent

29

---

---

---

---

---

---

---

---

**STAM Center** SECURE, TRUSTED, AND ASSURED MICROELECTRONICS **ASU Engineering** The Future School of Arizona State University

### Write-through Caches & Sequential Consistency

- T1 is executed
- T2 executed
- Write-through caches don't preserve sequential consistency either

	Cache-1	Memory	Cache-2	Prog T2
Initial	X=0 Y=10	X=0 Y=10	Y=- X=-	LD Y, R1
T1 executed	X=1 Y=11	X=0 Y=10	Y=- X=-	ST Y, R1
T2 executed	X=1 Y=11	X=1 Y=11	Y=11 X=0	LD X, R2
Cache-1 writes back X	X=1 Y=11	X=1 Y=11	Y=11 X=0	ST X, R2

30

---

---

---

---

---

---

---

---

**STAM Center** SECURE, TRUSTED, AND ASSURED MICROELECTRONICS **ASU Engineering** The Future School of Arizona State University

### Maintaining Sequential Consistency

- Problem: Multiple copies of a location in various caches can cause SC to break down.
- Hardware support is required such that
  - Only one processor at a time has write permission for a location
  - No processor can load a stale copy of the location after a write
    - Cache coherence protocols

31

---

---

---

---

---

---

---

---

**STAM Center** SECURE, TRUSTED, AND ASSURED MICROELECTRONICS **ASU Engineering** The Future School of Arizona State University

### A System with Multiple Caches

```

    graph TD
      P1[P L1] --- L2[L2]
      P2[P L1] --- L2
      P3[P L1] --- L2
      P4[P L1] --- L2
      L2 --- Interconnect[Interconnect]
      Interconnect --- M[M]
      P5[P L1] --- L1_2[L1]
      L1_2 --- L2
      P6[P L1] --- L1_2
      P7[P L1] --- L1_2
      P8[P L1] --- L1_2
  
```

- Modern systems often have hierarchical caches
- Each cache has exactly one parent but can have zero or more children
- Only a parent and its children can communicate directly

32

---

---

---

---

---

---

---

---

**STAM Center** SECURE, TRUSTED, AND ASSURED MICROELECTRONICS **ASU Engineering** The Future School of Arizona State University

### A System with Multiple Caches

```

    graph TD
      P1[P L1] --- L2[L2]
      P2[P L1] --- L2
      P3[P L1] --- L2
      P4[P L1] --- L2
      L2 --- Interconnect[Interconnect]
      Interconnect --- M[M]
      P5[P L1] --- L1_2[L1]
      L1_2 --- L2
      P6[P L1] --- L1_2
      P7[P L1] --- L1_2
      P8[P L1] --- L1_2
  
```

- Inclusion property is maintained between a parent and its children, i.e.,
  - a in  $L_i$  implies that a in  $L_{i+1}$

33

---

---

---

---

---

---

---

---

**STAM Center** SECURE, TRUSTED, AND ASSURED MICROELECTRONICS **ASU Engineering** Arizona State University

### Cache Coherence Protocols for SC

- Write request:
  - The address is invalidated in all other caches before the write is performed, or
  - The address is updated in all other caches after the write is performed
- Read request:
  - If a dirty copy is found in some cache, a write-back is performed before the memory is read
  - We will focus on Invalidation protocols as opposed to Update protocols

34

---

---

---

---

---

---

---

---

**STAM Center** SECURE, TRUSTED, AND ASSURED MICROELECTRONICS **ASU Engineering** Arizona State University

### Next Learning Module

- Cache Coherence Protocols

35

---

---

---

---

---

---

---

---