

# Memory Consistency and Cache Coherence

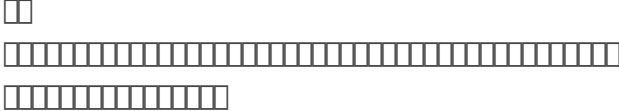

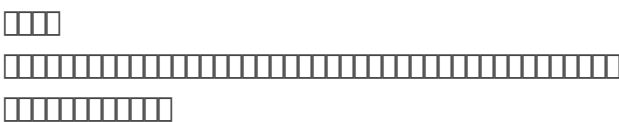
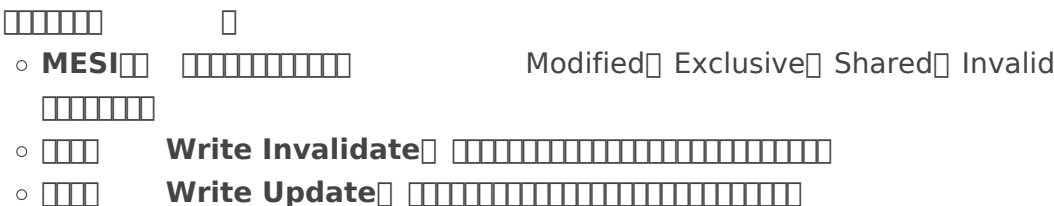









- Memory Consistency and Cache Coherence [1]
- TSO [1] Total Store Ordering [1][1][1][1]
- x86 [1][1][1][1][1][1][1][1][1][1]

# Memory Consistency and Cache Coherence

# 1. Memory Consistency

- [illegible]

## 2. Cache Coherence

- 
- 
- 
- 
  - **MESI**  Modified  Exclusive  Shared  Invalid
  -  **Write Invalidate** 
  -  **Write Update** 
- 

[illegible]

3. 

--	--	--	--

□□	Memory Consistency□□□□□□	Cache Coherence□□□□□□
□□□	□□□□□□□□□□	□□□□□□□□□□
□□	□□□□□□□□ / □□□□	□□□□□□□□□□□□
□□□□	□□□□□□□□□□□□□□	□□□□□□□□□□□□
□□□□	□□□□□□□□□□□□□□□□	□□□□□□□□ MESI□□□□□□□□

4. 

--	--

- **Memory Consistency**
- **Cache Coherence**

# TSO Total Store Ordering



- 1. TSO Total Store Ordering
- 2. x86 RISC-V TSO RVTSO x86 TSO  
Intel AMD
- 3. x86 SC sequential consistency FIFO write buffer
- 4. TSO SC store write buffer load bypass
  - 1. SC
  - 2. memory consistency
- 5. TSO store-load FENCE FENCE  
FENCE write buffer FENCE  
load

# x86

1. 

“ ”

1. “XCHG” “XADD”

2. LOCK 

LOCK CMPXCHG

3. A 

#LOCK

A

A

A
2. fence

1. sfence: sfence sfence

2. lfence lfence lfence

3. mfence mfence mfence
3. C++11

1. Acquire-Release Synchronizes-With

2. Release-Consume carry-a-dependency

```
enum memory_order {  
    memory_order_relaxed, // Relaxed  
    memory_order_consume, // Release-Consume  
    memory_order_acquire, // Acquire-Release  
  
    memory_order_release, // Acquire-Release  
  
    memory_order_acq_rel, // Acquire-Release memory_order_acquire memory_order_release  
    memory_order_seq_cst //  
};
```