ECE M116C - CS M151B Computer Architecture Systems - UCLA

Cheat Sheet

Fall 2023

Cache Memory

Types of Caches

- **Direct-Mapped:** Each block maps to exactly one cache line.
- **Set-Associative (A-way):** Each block maps to one of A cache lines in a set.
- **Fully Associative:** Any block can be placed in any cache line.

Cache Parameters

- **Number of Sets (S):** Determines the number of cache sets.
- **Associativity (A):** Number of ways per set.
- **Block Size (B):** Number of bytes per cache block.
- **Cache Size (C):** $C = S \times A \times B$

Address Breakdown

Virtual/Physical Address = Tag | Index | Offset

- **Offset Bits:** $\log_2(B)$
- **Index Bits:** $\log_2(S)$
- **Tag Bits:** Total Address Bits - (Index Bits + Offset Bits)

Cache Mapping Schemes

- **Direct-Mapped:** $A = 1$
- **Set-Associative:** $1 < A <$ Total Lines
- **Fully Associative:** $A =$ Total Lines

VIPT vs. PIPT

- **VIPT (Virtually Indexed, Physically Tagged):**
  - Index based on virtual address.
  - Tag based on physical address.
- **PIPT (Physically Indexed, Physically Tagged):**
  - Both index and tag based on physical address.

Virtual Memory

Key Concepts

- **Virtual Address Space (V)**: Total addressable memory by virtual addresses.
- **Physical Address Space (P)**: Total addressable memory by physical addresses.
- **Page Size (S)**: Size of a memory page.

Page Tables

- **Flat Page Table:** Single-level, simple but large.
- **Hierarchical Page Tables:** Multi-level (e.g., two-level), reduces memory overhead.

Number of Pages

$$\text{Virtual Pages} = \frac{V}{S}, \quad \text{Physical Pages} = \frac{P}{S}$$

Translation Lookaside Buffer (TLB)

Definition

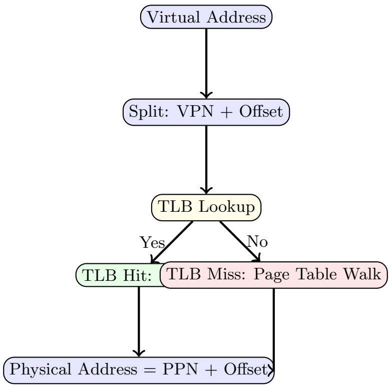
- A cache for virtual-to-physical address translations.

TLB Size Calculation

$$\text{TLB Size} = N_{\text{TLB}} \times (\text{VPN} + \text{PPN})$$

- N_{TLB}: Number of entries.
- VPN: Virtual Page Number bits.
- PPN: Physical Page Number bits.

TLB Workflow



Memory Consistency Models

Sequential Consistency (SC)

- Operations appear in a global sequential order.
- Easier to reason about but can limit performance.

Total Store Order (TSO)

- Relaxed model where stores are not immediately visible to all processors.
- Allows some reordering for performance.

Weak Consistency

- Fewer ordering guarantees.
- Requires explicit synchronization for ordering.

RISC-V Weak Memory Ordering (RVWMO)

- Optimistic load scheduling.
- Write-Read (W→R) constraints relaxed.
- Read-Read (R→R) constraints maintained.

Synchronization Mechanisms

Fences

- Ensure memory operations before the fence complete before those after.
- Example in RISC-V: fence rwio

Mutexes and Locks

- Ensure mutual exclusion for shared resources.
- Example: acquire(lock) while (lock != 0) /* busy wait */ lock = 1; release(lock) lock = 0;

Semaphores and Barriers

- **Semaphores:** Synchronize access to shared resources using counters.
- **Barriers:** Synchronize multiple threads to reach a certain point before proceeding.

Atomic Instructions

- Operations that execute indivisibly.
- Examples:
  - **Test-and-Set:** TS(int x) oldval = SWAP(x, 1); return oldval;
  - **Compare-and-Swap (CAS)**
  - **Load-Reserved/Store-Conditional (LR/SC):** loop: lr.w x2, 0(x1) addi x2, x0, 1 sc.w x2, 0(x1) bnez x2, loop

Cache Coherency

Coherency Protocols

- **MSI:** Modified, Shared, Invalid.
- **MESI:** Modified, Exclusive, Shared, Invalid.
- **MOESI:** Modified, Owned, Exclusive, Shared, Invalid.
- **MOESIF:** Modified, Owned, Exclusive, Shared, Invalid, Forwarder.

Coherency States

- **Modified (M):** Dirty, exclusive copy.
- **Exclusive (E):** Clean, exclusive copy.
- **Owned (O):** Dirty, shared copy.
- **Shared (S):** Clean, shared copies.
- **Invalid (I):** No valid copy.
- **Forwarder (F):** Provides data to other caches.

Coherency Protocol Operations

- **Read Miss:** Load data into cache, set state based on existing copies.
- **Write Miss:** Invalidate other copies, set state to Modified.
- **Write to Shared:** Upgrade to Modified, invalidate others.
- **Eviction:** Write back if in Modified/Owned state.

False Sharing

- Occurs when multiple processors cache the same cache line with different variables.
- Leads to unnecessary invalidations and reduced cache efficiency.

Directory-Based Coherence

- Uses a centralized directory to track cache line states.
- Scales better for large multi-core systems.
- Reduces bus traffic compared to snooping protocols.

DMA and I/O

Direct Memory Access (DMA)

- Allows I/O devices to transfer data directly to/from memory without CPU intervention.
- **Steps:**
  1. CPU initializes DMA transfer by setting up DMA registers.
  2. DMA controller handles data transfer between I/O device and memory.
  3. Upon completion, DMA controller sends an interrupt to notify the CPU.

Benefits and Drawbacks

- **Benefits:**
  - Reduces CPU overhead for data transfers.
  - Enables simultaneous data transfers and CPU processing.
- **Drawbacks:**
  - Complexity in managing multiple DMA channels.
  - Potential for bus contention and performance bottlenecks.

Network-on-Chip (NoC)

Overview

- On-chip communication subsystem connecting multiple cores, memory controllers, GPUs, and I/O devices.
- Facilitates efficient data exchange and scalability in multicore processors.

Design Challenges

- **Performance Optimization:** Ensuring low latency and high throughput.
- **Scalability:** Supporting increasing numbers of cores and devices.
- **Energy Efficiency:** Minimizing power consumption.
- **Security:** Protecting against data breaches and unauthorized access.
- **Integration with Emerging Paradigms:** Adapting to new computing models and technologies.

Design Ingredients

- **Topology:** Network structure (mesh, torus, star, etc.).
- **Routing Logic:** Algorithms for data packet traversal.
- **Router Design:** Handling data packets, buffering, and flow control.
- **Bandwidth and Latency:** Ensuring sufficient data transfer rates and minimal delays.

Key Formulae

Concept	Formula
Number of Virtual Pages	$\frac{V}{S}$
Number of Physical Pages	$\frac{P}{S}$
Page Table Size (Flat)	$E \times \frac{V}{S}$
Page Table Size (Two-Level)	$2^{L_1} \times E + 2^{L_2} \times E$
Maximum VIPT Cache Size	$A \times S \times B \leq 2^{\log_2(S_{\text{page}})}$
Total Cache Size	$C = S \times A \times B$
TLB Size	$N_{\text{TLB}} \times (\text{VPN} + \text{PPN})$
Average Access Time (AAT)	$(1 - M_{\text{TLB}}) \times (H_{\text{TLB}} + H_{\text{Cache}}) + M_{\text{TLB}} \times (P_{\text{Walk}} + H_{\text{Ca}}$
Offset Bits	$\log_2(B)$
Index Bits	$\log_2(S)$
Tag Bits	Total Address Bits − (Offset Bits + Index Bits)

Table 1: Summary of Key Formulae

Memory Consistency

Consistency vs. Coherency

- **Consistency:** Ordering of parallel accesses between different addresses.
- **Coherency:** Ordering of parallel accesses to the same address.

Memory Operation Ordering

- **Write-Read (W→R):** Write must complete before subsequent read.
- **Read-Read (R→R):** Read must complete before subsequent read.
- **Read-Write (R→W):** Read must complete before subsequent write.
- **Write-Write (W→W):** Write must complete before subsequent write.

Memory Models

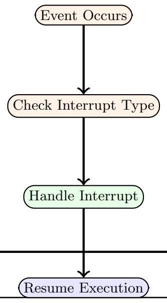
- **Sequential Consistency (SC):** Maintains all ordering constraints; simplest but can be slow.
- **Total Store Order (TSO):** Allows certain reordering for performance.
- **Weak Consistency:** More relaxed, requires explicit synchronization.

Writing Correct Programs

- **Race-Free Programming:** Ensure no data races.
- **Synchronization Primitives:** Use fences, mutexes, semaphores, etc.

Example Diagrams

Interrupt Handling Flowchart



Atomic Operations

Compare-and-Swap (CAS)

bool CAS(int* addr, int expected, int new$_v al) if (*addr == expected) *addr = new_v al; re$

Load-Reserved/Store-Conditional (LR/SC)

loop: lr.w x2, 0(x1) addi x2, x0, 1 sc.w x2, 0(x1) bnez x2, loop // Retry if store-conditional failed

Tables and Figures

Cache Performance Metrics

Cache Level	**Hit Time**	**Miss Penalty**
L1	H_1	P_1
L2	H_2	P_2
L3	H_3	P_3
Main Memory	H_m	-

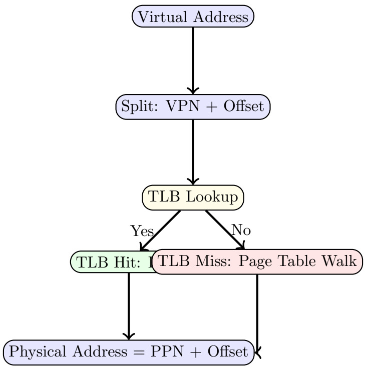
Table 2: Cache Performance Metrics

Comparison of Cache Mapping Schemes

Mapping Scheme	**Flexibility**	**Complexity**	**Speed**
Direct-Mapped	Low	Low	Fast
Set-Associative	Medium	Medium	Moderate
Fully Associative	High	High	Slow

Table 3: Comparison of Cache Mapping Schemes

TLB Workflow Diagram

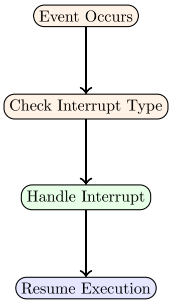


Summary of Key Formulae

Concept	**Formula**	**Variables**
Number of Virtual Pages	$\frac{V}{S}$	V (Virtual Address Space), S (Page Size)
Number of Physical Pages	$\frac{P}{S}$	P (Physical Address Space), S (Page Size)
Page Table Size (Flat)	$E \times \frac{V}{S}$	E (Page Table Entry Size), V, S
Page Table Size (Two-Level)	$2^{L_1} \times E + 2^{L_2} \times E$	L_1, L_2 (Levels), E, V, S
Maximum VIPT Cache Size	$A \times S \times B \leq 2^{\log_2(S_{\text{page}})}$	A (Associativity), S (Sets), B (Block Size), S_{page} (Page Size)
Total Cache Size	$C = S \times A \times B$	S, A, B
TLB Size	$N_{\text{TLB}} \times (\text{VPN} + \text{PPN})$	N_{TLB}, VPN, PPN
Average Access Time (AAT)	$(1 - M_{\text{TLB}}) \times (H_{\text{TLB}} + H_{\text{Cache}}) + M_{\text{TLB}} \times (P_{\text{Walk}} + H_{\text{Cache}})$	M_{TLB}, H_{TLB}, H_{Cache}, P_{Walk}
Offset Bits	$\log_2(B)$	B
Index Bits	$\log_2(S)$	S
Tag Bits	Total Address Bits−(Offset Bits+Index Bits)	Total Address, Offset, Index Bits

Table 4: Summary of Key Formulae

Interrupt Handling Flowchart



Cache Coherency Example

MOESIF Coherence Protocol Example

Initial States: All Cores Invalid

Access Sequence: R1, R2, W1, R2, W3

Final States:

P1: S

P2: S

P3: M

P4: I

Access	**P1**	**P2**	**P3**	**P4**
R1	E	I	I	I
R2	Fˆ	S	I	I
W1	M	I	I	I
R2	Oˆ	S	I	I
W3	I*	I	M	I

Table 5: MOESIF Coherence Protocol State Transitions

Atomic Operations

Compare-and-Swap (CAS)

bool CAS(int* addr, int expected, int new$_{v}al)if(*addr == expected)*addr = new_{v}al;returntrue;returnfalse;$

Load-Reserved/Store-Conditional (LR/SC)

loop: lr.w x2, 0(x1) addi x2, x0, 1 sc.w x2, 0(x1) bnez x2, loop // Retry if store-conditional failed