

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Engineering
Arizona State University

CSE 520

Computer Architecture II

Advanced Memory Operations
Supplement

Prof. Michel A. Kinsy

1

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Engineering
Arizona State University

Paged Memory Systems

- Processor generated address can be interpreted as a pair <page number, offset>
- A page table contains the physical address of the base of each page

page number offset

0	1
1	0
2	3
3	2

Address Space of User-1 Page Table of User-1

2

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Engineering
Arizona State University

Private Address Space per User

- Each user has a page table
- Page table contains an entry for each user page

3

STAM Center SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Engineering A. J. VALDES SCHOOL OF ENGINEERING Arizona State University

Where Should Page Tables Reside?

- Space required by the page tables is proportional to the address space, number of users, ...
 - Space requirement is large too expensive to keep in registers
- Special registers just for the current user:
 - What disadvantages does this have?
 - may not be feasible for large page tables
- Main memory:
 - Needs one reference to retrieve the page base address and another to access the data word
 - doubles number of memory references!

4

STAM Center SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Engineering A. J. VALDES SCHOOL OF ENGINEERING Arizona State University

Page Tables in Physical Memory

5

STAM Center SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Engineering A. J. VALDES SCHOOL OF ENGINEERING Arizona State University

Address Translation and Protection

- Every instruction and data access needs address translation and protection checks
- A good VM design needs to be fast (~ one cycle) and space efficient

6

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

Linear Page Table

- Page Table Entry (PTE) contains
 - PPN (physical page number) of memory-resident page
 - DPN (disk page number) of page swapped to disk, or non-existent page
 - Status bits for protection and usage
- OS changes page table base register to point to base of page table for active user process

The diagram illustrates the linear page table mechanism. A virtual address is divided into a Virtual Page Number (VPN) and an Offset. The PT Base Register points to the start of a Page Table. Each entry in the Page Table contains a Physical Page Number (PPN) or a Disk Page Number (DPN). An arrow labeled 'Offset' points from the Offset field of the virtual address to the corresponding Data word in the Data Pages.

7

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

Size of Linear Page Table

- With 32-bit addresses, 4-KB pages, and 4-byte PTEs:
 - 2^{20} PTEs, i.e., 4 MB page table per user
 - 4 GB of swap needed to back up full virtual address space
- Larger pages?
 - More internal fragmentation (don't use all memory in page)
 - Larger page fault penalty (more time to read from disk)
- What about 64-bit virtual address space???
 - Even 1MB pages would require 2^{44} 8-byte PTEs (35 TB!)

8

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

Hierarchical Page Table

The diagram shows a hierarchical page table structure. A 32-bit virtual address is split into a 10-bit L1 Index (p1), a 10-bit L2 Index (p2), and an offset. The L1 Index points to a Level-1 Page Table (Root of the Current Page Table), which contains pointers to Level-2 Page Tables. The L2 Index points to a specific Level-2 Page Table, which then points to the Data Pages. The offset is used to find the specific Data word within the Data Pages.

Legend:

- White box: page in primary memory
- Green box: page in secondary memory
- Black box: PTE of a nonexistent page

9

STAM Center SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Engineering ARIZONA STATE UNIVERSITY

Translation Lookaside Buffers

- Address translation is very expensive!
 - In a two-level page table, each reference becomes
 - Best case is _____?
 - Worst case is _____?
- Solution: Cache translations in TLB
 - TLB hit → Single Cycle Translation
 - TLB miss → Page Table Walk to refill

10

STAM Center SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Engineering ARIZONA STATE UNIVERSITY

Translation Lookaside Buffers

- Address translation is very expensive!
 - In a two-level page table, each reference becomes
 - Best case is ___3 memory references ___?
 - Worst case is ___2-page faults ___?
- Solution: Cache translations in TLB
 - TLB hit → Single Cycle Translation
 - TLB miss → Page Table Walk to refill

11

STAM Center SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Engineering ARIZONA STATE UNIVERSITY

TLB Designs

- Typically, 32-128 entries
- Usually fully associative
 - Each entry maps a large page, hence less spatial locality across pages more likely that two entries conflict
 - Sometimes larger TLBs are 4-8 way set-associative
- Random or FIFO replacement policy
 - Typically, only one page mapping per entry
 - No process information in TLB
- TLB Reach: Size of largest virtual address space that can be simultaneously mapped by TLB
 - Example: 64 TLB entries, 4KB pages, one page per entry
 - TLB Reach = _____?

12

STAM Center SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Engineering A CLAYTON SCHOOL OF Arizona State University

TLB Designs

- Typically, 32-128 entries
- Usually fully associative
 - Each entry maps a large page, hence less spatial locality across pages more likely that two entries conflict
 - Sometimes larger TLBs are 4-8 way set-associative
- Random or FIFO replacement policy
 - Typically, only one page mapping per entry
 - No process information in TLB
- TLB Reach: Size of largest virtual address space that can be simultaneously mapped by TLB
 - Example: 64 TLB entries, 4KB pages, one page per entry
 - TLB Reach = $64 \text{ entries} * 4 \text{ KB} = 256 \text{ KB}$ (if contiguous)...

13

STAM Center SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Engineering A CLAYTON SCHOOL OF Arizona State University

Handling A TLB Miss

- Software (MIPS, Alpha)
 - TLB miss causes an exception and the operating system walks the page tables and reloads TLB privileged "untranslated" addressing mode used for walk
- Hardware (SPARC v8, x86, PowerPC)
 - A memory management unit (MMU) walks the page tables and reloads the TLB
 - If a missing (data or PT) page is encountered during the TLB reloading, MMU gives up and signals a Page-Fault exception for the original instruction

14

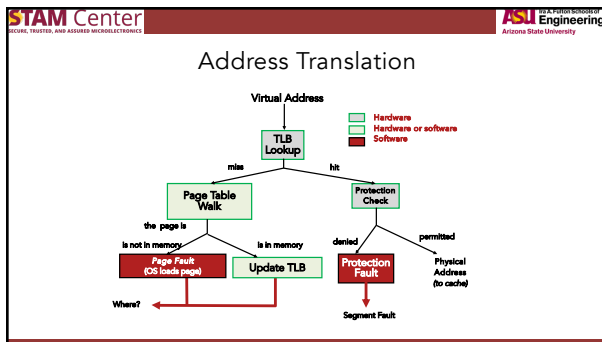
STAM Center SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Engineering A CLAYTON SCHOOL OF Arizona State University

Hierarchical Page Table Walk: SPARC v8

- MMU does this table walk in hardware on a TLB miss

15



16

Page Fault Handler

- When the referenced page is not in DRAM:
 - The missing page is located (or created)
 - It is brought in from disk, and page table is updated
 - Another job may be run on the CPU while the first job waits for the requested page to be read from disk
 - If no free pages are left, a page is swapped out
 - Approximate LRU replacement policy
- Since it takes a long time to transfer a page (msecs), page faults are handled completely in software by the OS
 - Untranslated addressing mode is essential to allow kernel to access page tables

17

Translation for Page Tables

Can references to page tables cause TLB misses?

The diagram shows a 'User PTE Base' (a small box) and a 'User Page Table (in virtual space)' (a larger box containing multiple entries). Arrows indicate that the PTE base and bits from the virtual address are used to access the page table.

- User VA translation causes a TLB miss
- Page table walk: User PTE Base and appropriate bits from VA are used to obtain virtual address VP for page table entry
- Get a TLB miss when we try to translate VP

18

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

Translation for Page Tables

The diagram illustrates the translation process. On the left, there are two boxes: 'User PTE Base' and 'System PTE Base'. Arrows from these boxes point to two corresponding tables: 'User Page Table (in virtual space)' and 'System Page Table (in physical space)'. From these tables, arrows point to a set of 'Data Pages' on the right. The User Page Table and System Page Table are shown as grids of colored cells, with lines connecting them to the Data Pages.

- When we get a TLB miss on VP translation, OS adds System PTE Base to bits from VP to find physical address of page table entry for VP

19

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

Swapping a Page of a Page Table

The diagram shows two page tables. The first table has three rows, with the middle row highlighted in red, representing a page in primary memory. The second table has three rows, with the middle row highlighted in grey, representing a page in secondary memory.

- A PTE in primary memory contains primary or secondary memory addresses
- A PTE in secondary memory contains only secondary memory addresses
- A page of a PT can be swapped out only if none its PTE's point to pages in the primary memory

20

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

Address Translation in CPU Pipeline

The diagram shows the CPU pipeline stages: Inst. TLB, Inst. Cache, Decode, Data TLB, and Data Cache. Arrows indicate the flow of data through these stages. Below the Inst. TLB and Data TLB stages, there are red arrows pointing to the text 'TLB miss? Page Fault? Protection violation?'.

- Software handlers need a restartable exception on page fault or protection violation
- Handling a TLB miss needs a hardware or software mechanism to refill TLB

21

STAM Center SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Engineering A CLAYTON SCHOOL OF Arizona State University

Address Translation in CPU Pipeline

- Need mechanisms to cope with the additional latency of a TLB
 - Slow down the clock
 - Pipeline the TLB and cache access
 - Virtual address caches
 - Parallel TLB/cache access

22

STAM Center SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Engineering A CLAYTON SCHOOL OF Arizona State University

Physical or Virtual Address Caches?

- One-step process in case of a hit (+)
- Cache needs to be flushed on a context switch unless address space identifiers (ASIDs) included in tags (-)
- Aliasing problems due to the sharing of pages (-)

23

STAM Center SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Engineering A CLAYTON SCHOOL OF Arizona State University

Aliasing in Virtual-Address Caches

Two virtual pages share one physical page

Virtual cache can have two copies of same physical data. Writes to one copy not visible to reads of other!

- General Solution: Disallow aliases to coexist in cache

24

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

Aliasing in Virtual-Address Caches

Tag	Data
VA ₁	1st Copy of Data at PA
VA ₂	2nd Copy of Data at PA

Two virtual pages share one physical page

Virtual cache can have two copies of same physical data. Writes to one copy not visible to reads of other!

- Software (i.e., OS) solution for direct-mapped cache
 - VAs of shared pages must agree in cache index bits; this ensures all VAs accessing same PA will conflict in direct-mapped cache (early SPARCs)

25

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

Concurrent Access to TLB & Cache

- Index L is available without consulting the TLB
 - Cache and TLB accesses can begin simultaneously

26

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

Concurrent Access to TLB & Cache

- Tag comparison is made after both accesses are completed
 - Cases: $L + b = k$ $L + b < k$
 - $L + b > k$ what happens here?
 - $L + b > k$: Partially VA cache. But it may be more effective to increase the way of the cache and decrease the index bits for same cache capacity

27

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

Virtual-Index Physical-Tag Caches

- After the PPN is known, W physical tags are compared
- Allows cache size to be greater than 2^{L+b} bytes

28

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

Anti-Aliasing Using L2

- Suppose VA1 and VA2 both map to PA and VA1 is already in L1, L2 (VA1 ≠ VA2)
 - After VA2 is resolved to PA, collision is detected in L2
 - Collision: VA1 will be purged from L1 and L2, and VA2 will be loaded → no aliasing!

Field a is different.

29

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

Virtually-Addressed L1

- Physically-addressed L2 can also be used to avoid aliases in virtually-addressed L1

L2 PA Cache
L2 "contains" L1

30

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Engineering
Arizona State University

Virtual Memory Use Today

- Desktops/servers have full demand-paged virtual memory
 - Portability between machines with different memory sizes
 - Protection between multiple users or multiple tasks
 - Share small physical memory among active tasks
 - Simplifies implementation of some OS features

31

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Engineering
Arizona State University

Virtual Memory Use Today

- Vector supercomputers have translation and protection but not demand-paging (Older Crays: base&bound, Japanese & Cray X1: pages)
 - Do not waste expensive CPU time thrashing to disk (make jobs fit in memory)
 - Mostly run in batch mode (run set of jobs that fits in memory)
 - Difficult to implement restartable vector instructions

32

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Engineering
Arizona State University

Virtual Memory Use Today

- Most embedded processors and DSPs provide physical addressing only
 - Cannot afford area/speed/power budget for virtual memory support
 - Often there is no secondary storage to swap to!
 - Programs custom written for particular memory configuration in product
 - Difficult to implement restartable instructions for exposed architectures

33

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Arizona State University
Engineering

Next Learning Module

- Cache Coherence

34