Department of Electronics & Communication Engg.

Course : Real Time Operating Systems-10EC842 **Sem.: 8th (2017-18)**

Course Coordinator:

S.S.Malaj

UNIT 4 MEMORY

PHYSICAL HIERARCHY

•Physical memory hierarchy for an embedded processor can vary significantly based upon Hardware Architecture.
•Most often a Harvard Architecture is used.
•Evolved from GPCS & is often employed by embedded system as well.
•Typical Harvard Architecture
Separate L1 (Level-1) instruction & data caches .
But Unified L2 cache & either on chip SRAM or external DDR(Dynamic data RAM).

Local partitioning & segmenting of physical hierarchy by firmware

Memory is Global resource in a single address space with all other MMIO devices.

Memory system design most influenced Architecture & goals:

To Maximize throughput but necessarily to minimize the latency for any single memory access or operation.

Memory system design-----most influenced by GPC Architecture – goals—to maximize throughput ,but not necessary to minimize the latency for any single memory access or operation.

Multilevel cached memory hierarchy for GPC platforms now often includes Harvard L1 & unified L2 caches on chip with off chip L3 unified cache.

Caches for GPCs are most often set associative with aging bits for each cache set so that LRU sets are replaced when a cache line must be loaded.

N way set-associative cache –can load an address reference into any N ways in the cache allowing for the LRU line to be replaced.

LRU replacement policy—leads to a high cache hit to miss ratio so that a processor most often finds data in cache & does not have to suffer the penalty of a cache miss.

Direct mapped cache –each address can be loaded into one & only one cache line making the replacement policy simple.

Thrashing –occurs when two addresses are referenced & keep knocking each other out of cache,greatly decreasing cache efficiency.

Fig: Direct mapped cache where a main memory is four times the size of cache memory has memory sets which can only be loaded into one cache set.

Multilevel cached memory hierarchy for GPC platforms:
Includes Harvard L1 & unified L2 caches on chip with off chip L3 unified cache.
Caches for GPCS: Set associative with aging bits for each cache line loaded.
N way set associative cache :Can Load an address reference into any N ways in the cache allowing for LRU line to be replaced.

LRU Relacement policy:

Leads to a high cache hit to miss ratio so that a process most often finds data in cache & need not have to suffer penalty of a cache miss.

Set associative cache is an comprise between a direst ,mapped & fully associative cache.

Comparison between a direct mapped & fully associative cache

Direct Mapped	Fully Associative cache
Each address can be loaded into one & only one cache line making replacements policy,simple causing Cache thrashing	Cost of many aging bits Most caches are 4 way or 8 way set associative with 2 or 3 aging bits for LRU.

Thrashing occurs when two addresses are referenced & keep knocking each other out of cache ,greatly decreasing cache efficiency.

Capacity & Allocation

Most basic resource concern associated with memory----total capacity needed.

Algorithms(many) include space & time trade-offs & services---need(often) significant data context for processing.

---Cache does not contribute to total capacity because ---it stores only copies of data rather than unique data.
---another downside to cache for embedded systems where capacity is often limited.

Preventing exceptions rather than handling them is more proactive.

Code can check for conditions that would cause an exception & verify arguments to avoid the possibility of task suspension or target reboot.

Standard method:
For this type of defensive programming is to include assert checks in code to isolate errant conditions for debug.

In C Code:
Pointers are often passed to functions for efficiency.

A caller of a function might not pass a valid pointer,& this can cause an exception or errant behavior in the called function that can be difficult to trace back to bad pointer.

Are measured differently, but both provide some indication of system robustness & quality expectations.

Because system failures can be dangerous, both also relate to system safety.

Increasing availability & reliability of a system requires time & monetary investment ,increasing cost,& increasing time to market for products.

Is an unintentional decrease in overall availability & reliability.

RELIABILITY & AVAILABILITY:SIMILARITIES & DIFFERENCES

Availability is the percentage of time over a well defined period that a system or service is available for users.

Eg.if a system is said to have 99.999% ,or five nines availability,this system must not be unavailable more than five minutes over the course of a year.

Quick recovery & restoration of a service after a fault greatly increases availability.

High reliability is described by the old adage that a chain is only as strong as its weakest link.

A system built from components that have very low probability of failure leads to high system reliability.

Overall expected system reliability

Queries …?