**Motivation**

- Shared hardware like caches & TLBs introduce timing unpredictability for real-time systems (RTS).
- Worst-case execution time (WCET) analysis for RTS with shared hardware resources often so pessimistic that extra processing capacity of multicore systems is negated.

**Problem**

- TLBs

**Solution & Results**

- **Cache Locking:**
  - Apply a multiprocessor real-time locking protocol to cache colors.
  - Treat each job as a critical section.

- **Cache Scheduling:**
  - Apply existing scheduling algorithms (e.g., Rate Monotonic) to cache accesses.
  - Allows for preemptions w.r.t. the cache (see example).

**Pseudo code for TLB reverse engineering**

```
1: ....
2: ....
3: //allocate a huge array
4: int * data = (int *) calloc(numOfElements,4);
5: pageOffset = 0;
6: ...
7: //access pages for the first time
8: PAPI read(eventSet, value1);
9: for i = 0 to noOfPagesToAccess ... pageOffset = pageOffset + (s * 1024);10: end for
11: PAPI read(eventSet, value2);
12: calculateMaxMisses();
13: calculateMinMisses();
14: ...
15: ...
```

**TLBs – Solutions & Results**

- Tighter WCET estimates can be established if we know which references hit in the cache and which do not.
- Other shared resources like TLBs show similar timing unpredictability.

**Solution**

- Our solutions focus on two shared resources: shared caches and TLBs.
- **Cache Locking:**
  - Apply a multiprocessor real-time locking protocol to cache colors.
  - Treat each job as a critical section.
- **Cache Scheduling:**
  - Apply existing scheduling algorithms (e.g., Rate Monotonic) to cache accesses.
  - Allows for preemptions w.r.t. the cache (see example).

**Conclusions**

- Developed 2 techniques based on cache coloring
  - eliminate cross-core cache evictions.
- Implemented in a mixed-criticality scheduler: LITMUS
- Evaluated on an ARM Tegra 3 platform
- Conducted overhead-aware schedulability study
  - based on measured overheads.
- Cache scheduling & cache locking improved schedulability
  - over a system with unmanaged cache.
- TLBs not using LRU replacement maybe PLRU (ongoing work)
- TLB-miss bounds not deterministic – even for accessing <4 pages.
- Current work: mechanisms & policies for TLB predictability