Always a Step Ahead

ARDEC

ARMAMENTS

Use of Multi-core Technology in Fuzing Systems

Presented by:

Jeffrey Fornoff – ARDEC Fuze Division

NDIA 60th Annual Fuze Conference, May 9-11, 2017

UNPARALLELED COMMITMENT & SOLUTIONS

Act like someone's life depends on what we do.

U.S. ARMY RDECOM®

U.S. ARMY ARMAMENT RESEARCH, DEVELOPMENT & ENGINEERING CENTER

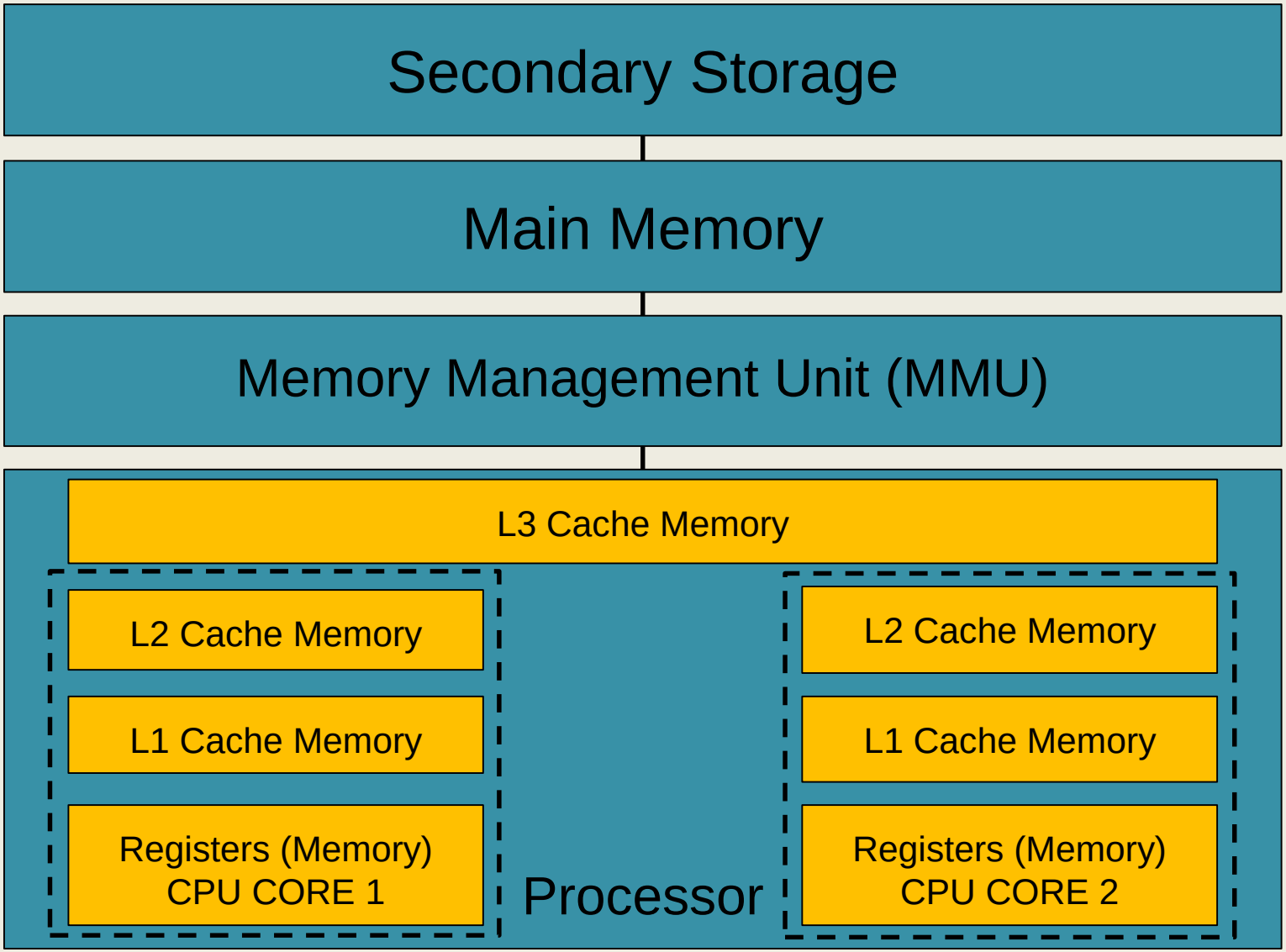
PROBLEM STATEMENT

- Microprocessors have been utilized in fuzes and fuzing systems for many years
- Use of today's multi-core technology may be attractive for some high level munition fuzing and initiation system applications requiring complex arm/disarm/rearm/continuous monitoring capabilities. However to adequately address safety standard requirements, attention needs to be given to the unique challenges posed by multi-core processing with respect to safety critical software that controls Safe and Arming functions.
- Typical software architectures used for control of fuze safety system, (ex., command and control of Safe and Arming functions) use some version of a virtual partition to isolate safety-critical functions from mission-critical functions. With the advent of Real-time Operation Systems (RTOS) that allows for software architecture partitioning, multi-core processor technologies now predominantly are used and attention needs to be given to the unique challenges of enabling multi-core processors with respect to RTOS and safety critical software.
- Safety critical software that executes on multi-core technologies must now consider temporal aspects that can arise with multi-threaded software executing on multiple CPU cores not only in terms of deterministic execution, but also on data integrity
- Currently, there are no standards by which developers can follow to implement safety functions in a system architecture containing multi-core technology that insures hardware and software failure modes are adequately identified and properly mitigated
- This discussion identifies unique engineering criteria that should be considered when implementing safe and arming functions utilizing multi-core technologies. These criteria involve both hardware and software design considerations.

Multi-core Unique Issues

- Compilers generate multiple execution threads to take advantage of multi-core (multi-processor) technology
- Temporal issues arise as a result of multi-threaded code
  - Different threads execute simultaneously on each CPU to increase speed of the application
  - Code becomes less deterministic because it has been broken down into multiple execution threads (as analyzed by the compiler)
  - Data that is accessed in multi-threaded code may end up in a race condition
  - In a multi-process environment (such as exists in operating systems) additional programming constraints must be considered
- Hardware complexity also plays a role
  - Shared memory between processors (CPUs)
    - Cache memory
    - Main memory
    - Secondary storage
  - Possible hardware race conditions

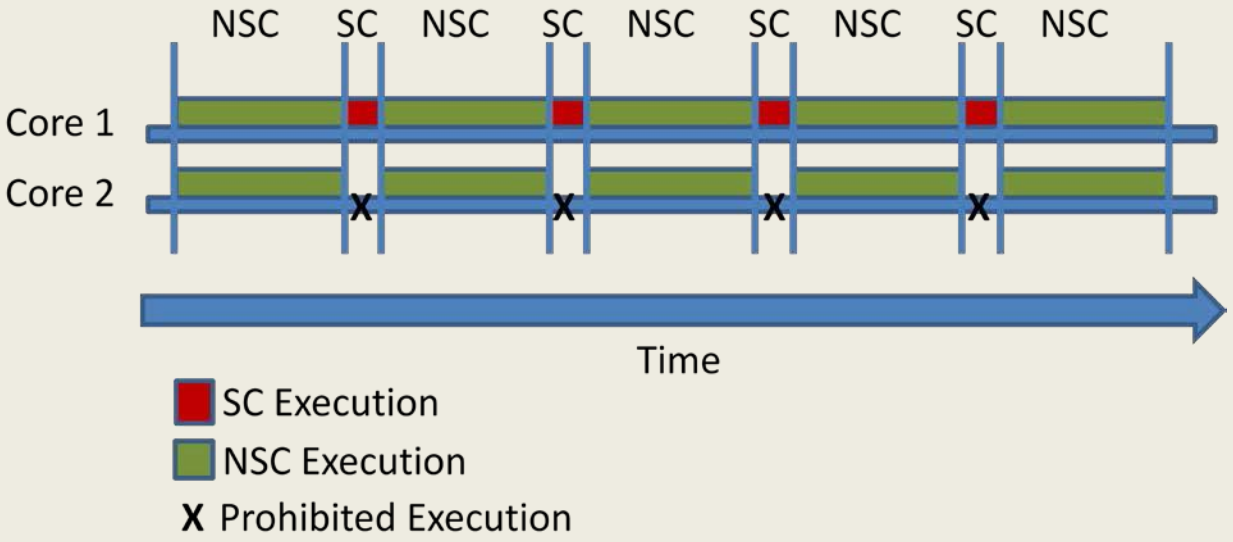
Typical Memory Hierarchy



PARTITIONING STRATEGIES

- Partitioning is required to separate safety critical processing from non-safety critical processing
  - Software
    - Safety functions should be designed and compiled as single-threaded
    - Safety functions isolated from other code
    - Safety data isolated from other data
    - Each safety function and its associated data is segregated as well
  - Hardware
    - Since microprocessor hardware cannot be physically separated (as it is contained in a single die), the use (or execution) of the hardware must be separated
    - Safety should have exclusive use of the hardware when executing as much as physically possible

PARTITIONING CPU EXECUTION



Example shows how safety critical code execution is isolated from non-safety critical execution. Safety critical code is single-threaded and executes on only 1 core while other multi-threaded non-safety critical code is allowed to execute on multi-cores simultaneously

USING EXISTING STANDARDS

- Hardware
  - JOTP-51 Safety Features (SF) shall be functionally and physically separated
  - MIL-STD-882E Appendix B.2.2.5 Design Requirements to consider physical partitioning of processors
- Software (code)
  - MIL-STD-882E Task 208.1 Functional Hazard Analysis (FHA) describes the need to partition Safety Critical Functions (SCFs) and Safety Related Functions (SRFs) in the design architecture
  - MIL-STD-882E Appendix B.2.2.5 Design Requirements to consider the need to partition safety functions (software modules)
- Software (data)
  - AOP-52 Section 4.10.21 Specifies that safety related data shall be partitioned away from other non-safety related data
- There are additional design considerations needed when utilizing multi-core technology where safety functions are implemented in software

ADDITIONAL CONSIDERATIONS

- Is the code running standalone or under the control of an operating system?
  - Standalone code is easier to analyze and test
  - If using an operating system, a Real-time Operating System is required
    - Windows is NOT a RTOS – Microsoft actually warns against its use for safety critical operations
    - Linux is NOT a RTOS
- Is the code single-threaded or multi-threaded?
  - Restrict the compiler from generating multi-threaded code
- Design the system architecture to partition hardware and software elements of safety functions
- In addition to the verifying and validating the application, certification may be needed for other software elements of the system such as
  - The Operating System (OS)
  - Compilers and Assemblers
  - Application development software such as pre-processors and deployment tools

REFERENCES

- **MIL-STD-882E** Department of Defense Standard Practice System Safety
- **JOTP-51** Technical Manual for the use of Logic Devices in Safety Features
- **AOP-52** Guidance of Software Safety Design and Assessment of Munition-Related Computing Systems

- It should be noted that all software contained in fuzes or fuzing system needs to be reviewed by the Army Fuze Safety Review Board (AFSRB)
  - Safety Critical Code must be reviewed by the AFSRB Software Safety Panel
  - Requirements specified in AOP-52 must be satisfied
- If a fuze or fuzing system is identified as a joint program then software must be reviewed by the Joint Services Software Safety Authorities

CONCLUSIONS / QUESTIONS

Questions?