Assessment of Hardware Based RTOS for Safety Critical System

Lee, Siaw Chen
siaw.chen.lee@intel.com
Functional Safety Engineering
Internet of Thing Group (IoTG)
Intel Corporation,
Bayan Lepas, Penang, Malaysia

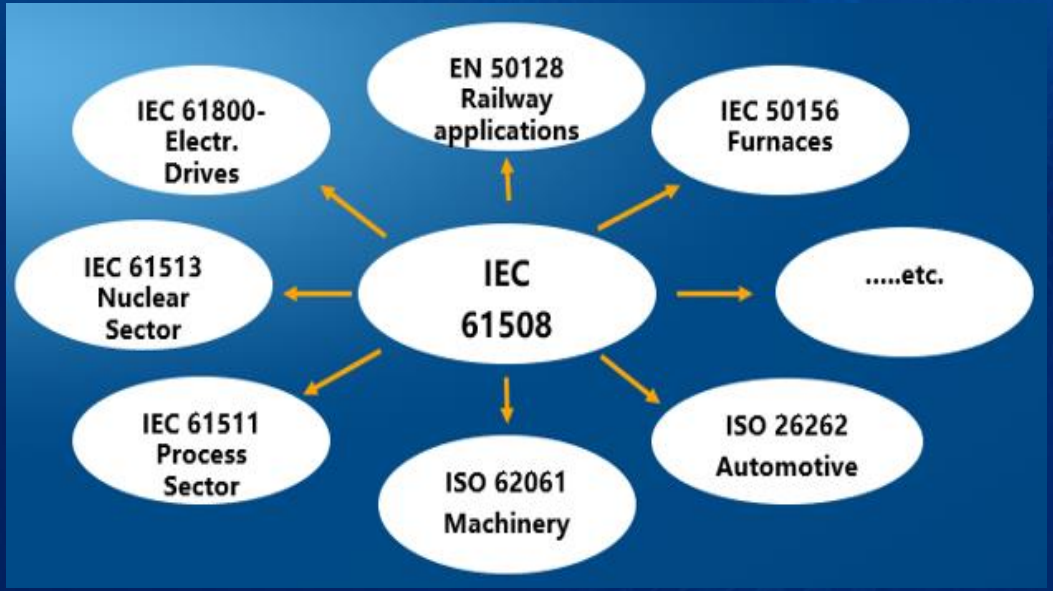
Ong, Soon Ee
soon.ee.ong@intel.com
Technologist
Internet of Thing Group (IoTG)
Intel Corporation,
Bayan Lepas, Penang, Malaysia

Internet of Things Group

Safety Critical System (SCS)

System whose **failure** or malfunction may **severely harm** human live, environment or equipment.

Functional Safety (FuSa)

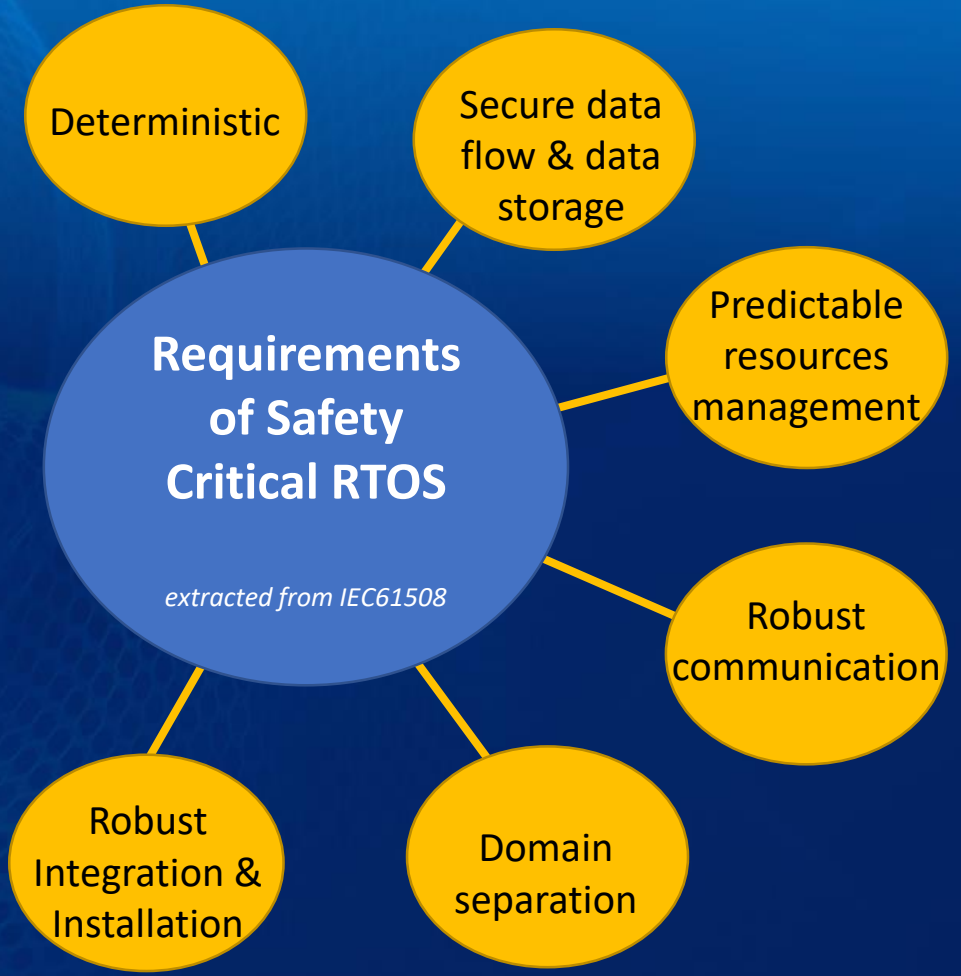


International Standards for FuSa

Absence of **unreasonable risk** due to hazards caused by malfunctioning behavior of **E/E system**

Using RTOS for SCS in Edge IoT

- Tremendous amount of Safety Critical Systems are implemented on Edge IoT nowadays.
- RTOS is one of the key components in Edge IoT system whereby any failure or malfunction of RTOS could potentially crash the entire system.



Safety Challenges of SW Based RTOS

Robustness

- Random **SW glitch**
- Multiple point attack
- **Hyper-jacking** cause crashes of RTOS
- **lock up** of critical system resources preventing
- crashing entire system

Domain separation

- lower barrier on memory separation, e.g. no memory virtualization.
- 1 task/apps to **temper** other task/apps; memory; critical section of system

Verification & Installation

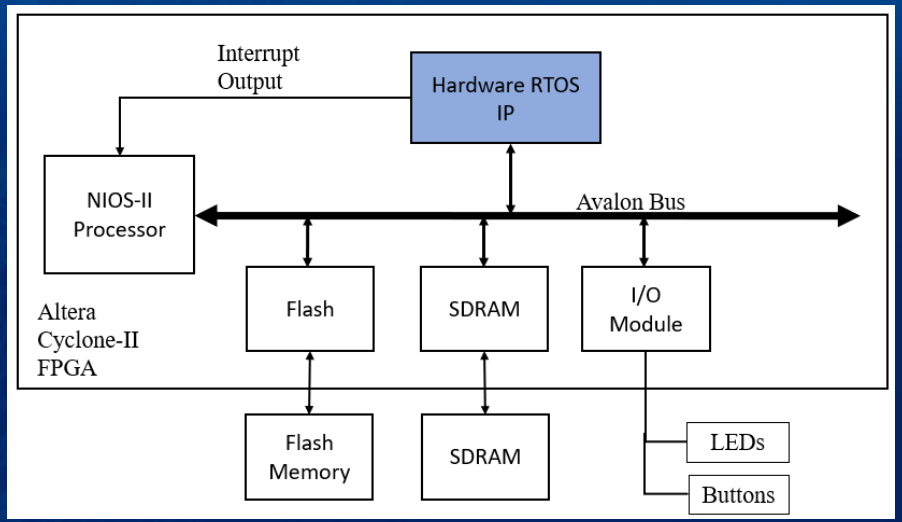
- **Exhaustive** SW testing requires enormous effort to achieve higher SIL level
- **Complex** and error-prone installation is a complex process

Safety Standard Compliances

- **Tougher** to achieve higher SIL eg: SIL3 is the highest SIL level for software-only component with IEC 61508

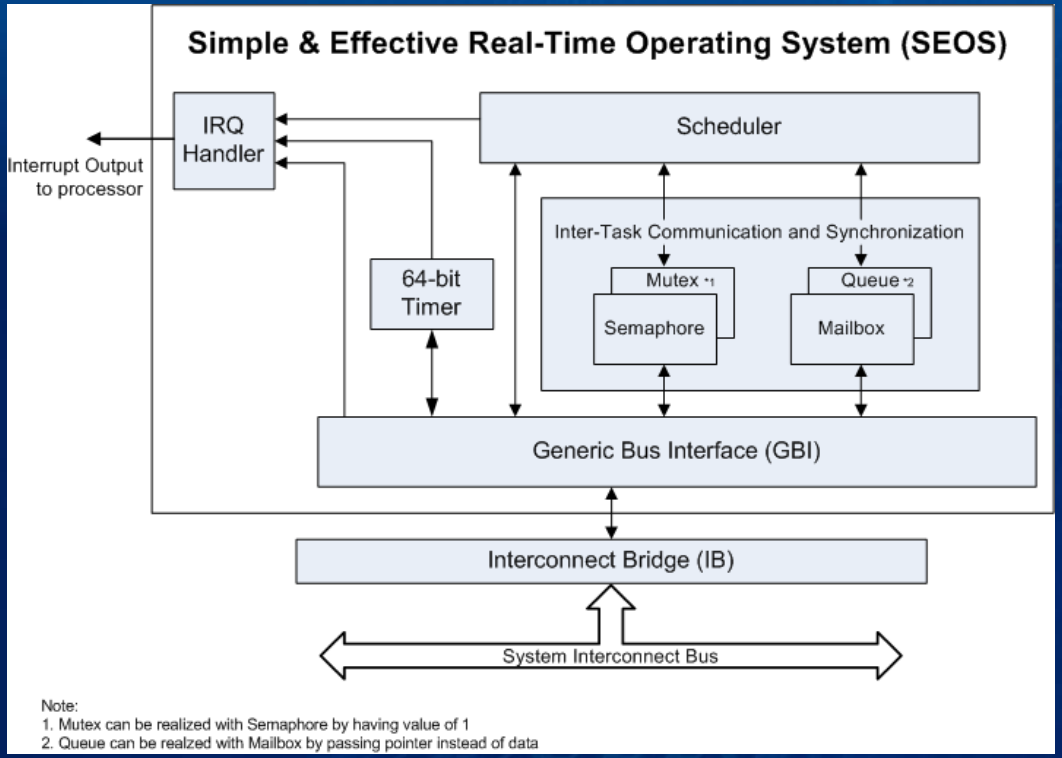
Hardware Based RTOS Approach

- **Improve** short comings of software based RTOS in safety aspect.
- **Implemented as IP** in SoC/FPGA that provides similar functionalities including task scheduling, inter-task communication and resource management.
- Also comes with **better latency** and **jitter performance** as compared to SW based RTOS.



Example of HW RTOS implementation in SoC/FPGA

Internal Design of Hardware Based RTOS



- **Scheduler** - priority-based pre-emptive kernel, rate-monotonic scheduling algorithm
- **Timer** - 64-bit programmable countdown timer for task scheduler and general purpose
- **Inter Task Communication and Synchronization** –facilitate software for tasks synchronization and communication.
- **Interconnect Bridge and General Bus Interface** - IB is a protocol dependent slave port bridge that provides connection to system. Configurable to support various interconnect types.

Benefits of Hardware Based RTOS

1. Deterministic & Robustness
   - **High predictable behavior**.
   - **Immune** to random SW glitch, virus or malware attack, hyper-jacking etc that hold up resources.
2. Secure data flow & data storage
   - **Critical OS data** includes scheduling, context switch stack etc. are **kept in HW register** inside SEOS IP that is accessible only by the IP.
3. Predictable resources management
   - **Scheduling** handled at **HW level** prevents crashes & lock-ups.
   - **Shared resources** are **managed by hardware**, thus invulnerable to SW bug.
4. Robust internal communication
   - Inter-task communication is executed by the SEOS hardware IP gives **robust transition of data**.
5. Domain separation
   - **Critical tasks register data & stacks** are managed by HW, not accessible by SW and is **total isolated** from one task to another.
6. Robust Integration & Installation
   - HW IP can be easily integrated into SoC during HW development cycle. Critical OS data are kept at register level, **does not required memory mapping**.
7. Verification
   - Hardware based RTOS is verified as **HW IP** which has **higher level of safety coverage and lower effort**, as compare to SW verification of RTOS.
8. Safety Standard Compliances
   - Higher safety coverage of verification for HW based RTOS made it **easier to comply to IEC61508** safety standard.

Thank You