A Safe, Secure, and Adaptive Mixed-Criticality System

Youssef Zaki (youssef.zaki@alten.se), Andreas Hammar, Detlef Scholle, and Jens Bergman
Alten Sverige AB, Knarrarnäsgatan 7, 164 40 Kista, Stockholm

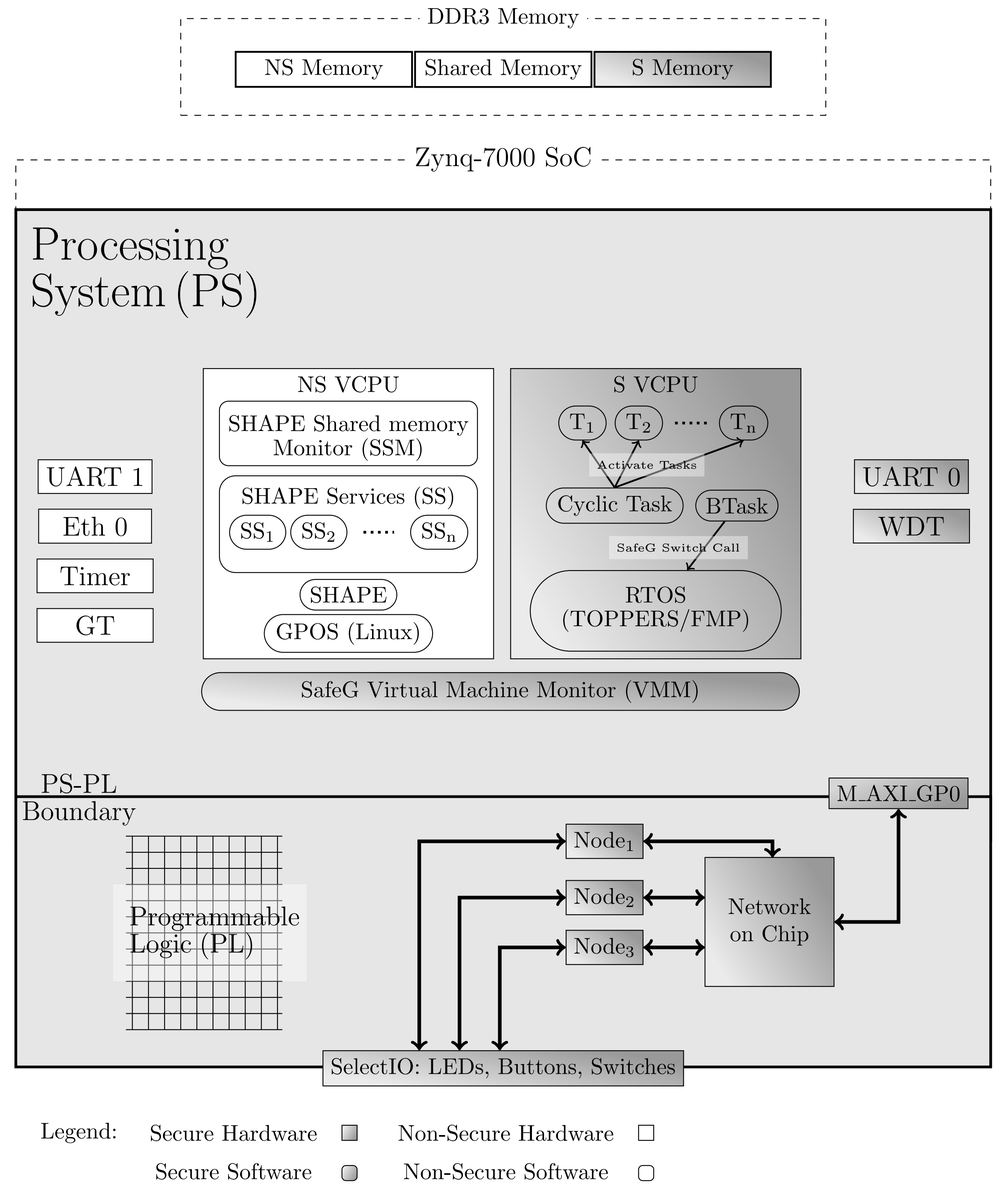
Summary

- Safe and secure integration of critical and non-critical applications into a single computing multi-processor platform.
- Concurrent execution of heterogeneous operating systems.
- System adaptability based on Service-Oriented Architecture.
- System optimization based on available hardware resources.
- Prototype implementation on the EMC2-DP board from Sundance and the Zedboard from Digilent.

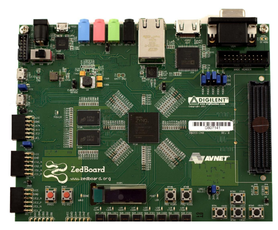
System Description

- ARM TrustZone technology enforces isolation between critical and non-critical resources in the system by marking system resources as either Secure (S) or Non-Secure (NS).
- The Virtual Machine Monitor initializes the TrustZone-based system, and controls the switching activities between the Secure and Non-Secure software and hardware components.
- Each Cortex-A9 processor in the PS is converted into two virtual CPUs: Secure VCPU and Non-Secure VCPU.
- Peripheral devices are marked as Secure and Non-Secure, and access is restricted only for the secure resources.
- The external memory is partitioned into Secure, Non-Secure, and Shared regions. This partitioning ensures isolation between operating systems, and allows inter-OS communication through the shared memory region.
- The DualOSCom protocol is integrated into the SHAPE framework as a SHAPE Shared memory Monitor (SSM).
- Employing Network-on-Chip (NoC) for efficient inter-processor communication in the FPGA region, and spatial partitioning.
- NoC is accessed through the M_AXI_GP0 port.

Hardware and Software Synopsis of Implemented System



Zedboard



EMC2-DP Board

Overview of Implemented Demo System

- Implemented system contains 4 system nodes: 2 Zedboards, 1 EMC2-DP board, and the control station.
- All devices are connected through a networking switch with the Ethernet medium.
- Buttons, switches, and LEDs simulate system sensors.

Implementation Statistics

System File	Size (Bytes)
GPOS(Linux v3.6.10)	2,368,472
SOA(SHAPE)	248,164
RTOS(FMP)	50,160
VMM(SafeG)	4,920

- RTOS to GPOS switch time: 2μs.

Illustration of Demo System