Onboard Computer Case-Study on RISC-V Core

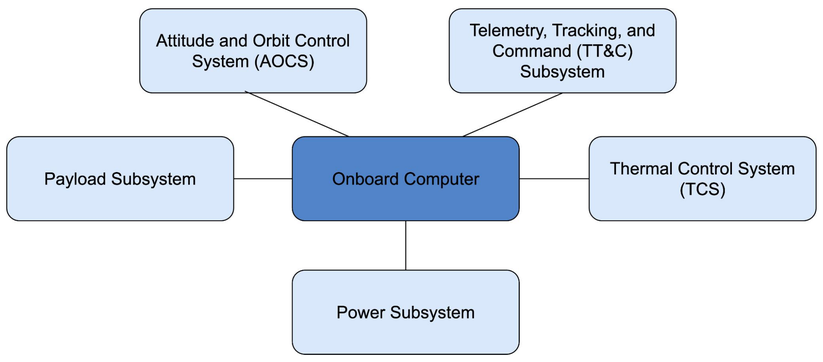
Luís França, Fabio Benevenuti, Fernanda Lima Kastensmidt
Federal University of Rio Grande do Sul (UFRGS), Brazil. E-mail: {luis.franca, fbenevenuti, fglima}@inf.ufrgs.br

Objectives

- Present the implementation of NASA's Core Flight System software architecture on a RISC-V platform.
- Open Source port of the Operating System Abstraction Layer (OSAL) to FreeRTOS.
- Execution of an Onboard Computer Case study Application on PolarFire FPGA board.
- Contribute to the use of RISC-V for space applications with open source software

1. Onboard Computer (OBC)

- Onboard Computer is responsible for the management of satellite subsystems.



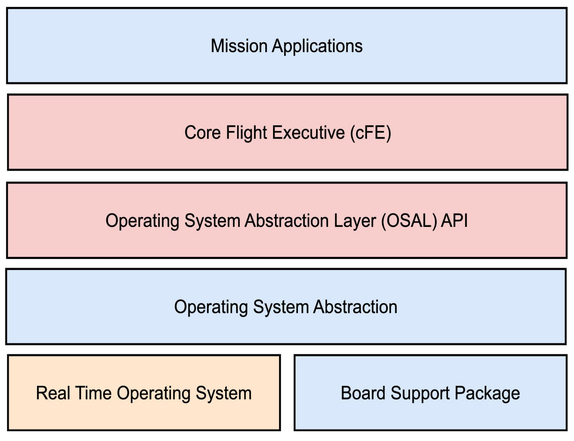
NANOSATC-BR2

- Onboard Computer (OBC):
  - Different processors (ARMs, RISC-V, …)
  - Different Operating Systems
- Goal: Independence of Hardware and Operating System
  - Reduction of cost and development time
  - More focus in software solutions, instead of software portability
- Use of RISC-V for OBC's
  - Open-source ISA
  - Can be implemented hardcore or softcore
  - Customizable architecture

2. NASA Core Flight System (cFS)

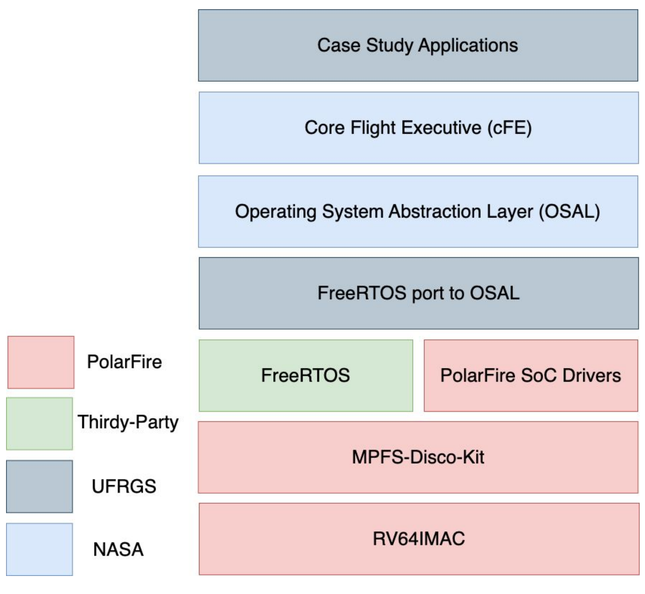
- The NASA Core Flight System (cFS) is a open source software architecture https://github.com/nasa/cFS.git
- It consists of a abstraction layer, a set of common services, and mission-specific libraries and applications.
- The use of cFS can bring reduction in cost and development time by making the solutions independent of hardware and operating system

- Operating System Abstraction Layer (OSAL): portability across different operating systems.
- Core Flight Executive (cFE): common services for flight software
- Application Layer: mission specific and reusable solutions.



- As a result of cFS, a catalog of open-source solutions is available, allowing for customization and integration into the OBC without requiring additional portability efforts

3. Port FreeRTOS Abstraction Layer

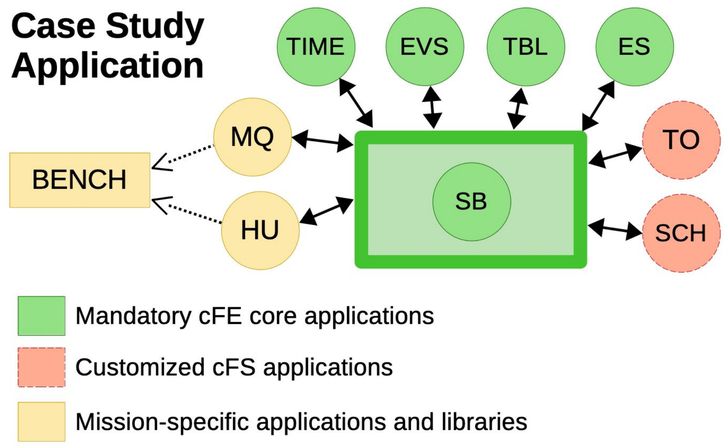


- The cFS supports: RTEMS, POSIX, and VxWorks operating systems.
- We ported OSAL to FreeRTOS
- Target hardware: RISC-V
- FreeRTOS:
  - Open source Real Time Operating System (RTOS)
  - Was used in Floripa-SAT1, NanosatC-BR1, and NanosatC-BR2

Open Source code from this work: https://github.com/projetoChi2p/fs-nasa-cfs-osal.git

4. Case study applications for the RISC-V Onboard Computer

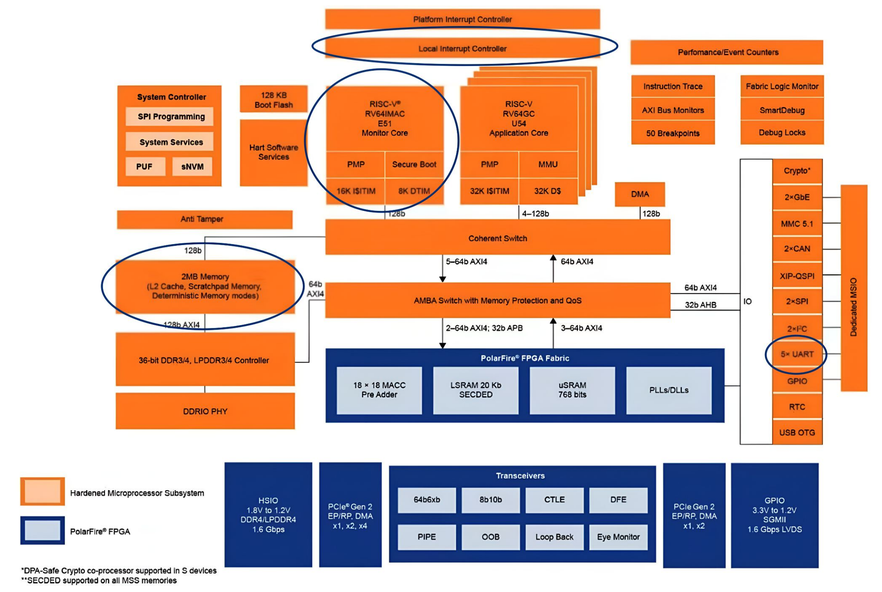
- Software Bus (SB): message router between cFE services and applications.
- Timer (TIME): time-related functionalities.
- Executive Services (ES): manages the application runtime environment.
- Event Services (EVS): event logging and messaging.
- Table Services (TBL): applications data structures.
- Telemetry Output (TO): presents messages to the console.
- Scheduler (SCH): periodically trigger execution of applications.



- Case study Applications and Libraries:
  - BENCH: A library with common utilities.
  - MQ (Fixed Point Matrix Multiplication): scientific operations
  - HU (Huffman Encoding and Decoding): Payload data management

5. RISC-V Characteristics

- The MPFS-Disco-Kit board was used to integrate a RISC-V processor with cFS and FreeRTO
- RISC-V Processor System:
  - 1× SiFive E51 (RV64IMAC) monitor core
  - 4× SiFive U54 (RV64IMAFDC) application cores
  - 625 MHz clock speed, 5-stage execution pipeline (7-stage for floating-point operations)
- The OBC's executes single core, in one of the E54's or the in E51 core.



6. Comparing OBC's Processors

	Nanosat-BR1	Nanosat-BR2	Study 1	Study 2
Model	ARM7TDMI	ARM92EJ-S	Arm-Cortex M7	RISC-V
Core	1 x Armv4T (32-bit)	1 x Armv5TEJ (32-bit)	1 x Armv7-M (32-bit)	1 X RV64IMAC (64-bit)
CPU Freq.	40 MHz	400 MHz	216 MHz	625 MHz
On-chip RAM	4 KiB	32 KiB	512 KiB	2 MiB
On-chip Flash	-	-	2 MiB	128 Ki
On-board RAM	2 MiB	64 MiB	-	1 GiB
On-board Flash	8 MiB	1 MiB	-	-
Mass Storage	microSD card	SD card	-	microSD card

- Nanosat-BR1 and Nanosat-BR2
  - Arm-based processor, OBC's developed using FreeRTOS

- This work adds value to the discussion of open-source flight software.
- The results show the feasibility of open-flight software stacks on embedded RISC-V for space.

Supported by:
FAPERGS · Office of Naval Research Global Science & Technology · CNPq · CAPES