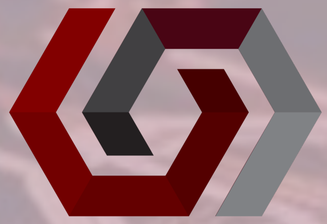


ZEPHYR AND OPEN AMP

Mohammed Billoo

MAB Labs Embedded Solutions

Zephyr Dev Summit 2024

AGENDA

- Motivation
- OpenAmp in Zephyr
- System Architecture
- Development Process (**WARNING: Yocto ahead!**)
- Common Issues and Resolutions
- Next steps

Mohammed Billoo

(mab@mab-labs.com)

/mab-embedded

@mabembedded

THE SPEAKER

- Embedded Software Consultant
- Design Work
  - Medical Devices
  - Scientific Instruments
  - LIDAR
  - Custom ASICs
- Experience/Expertise
  - Zephyr RTOS
  - Embedded Linux
  - GUI-based applications

BIOS FOOD NEWSLETTER

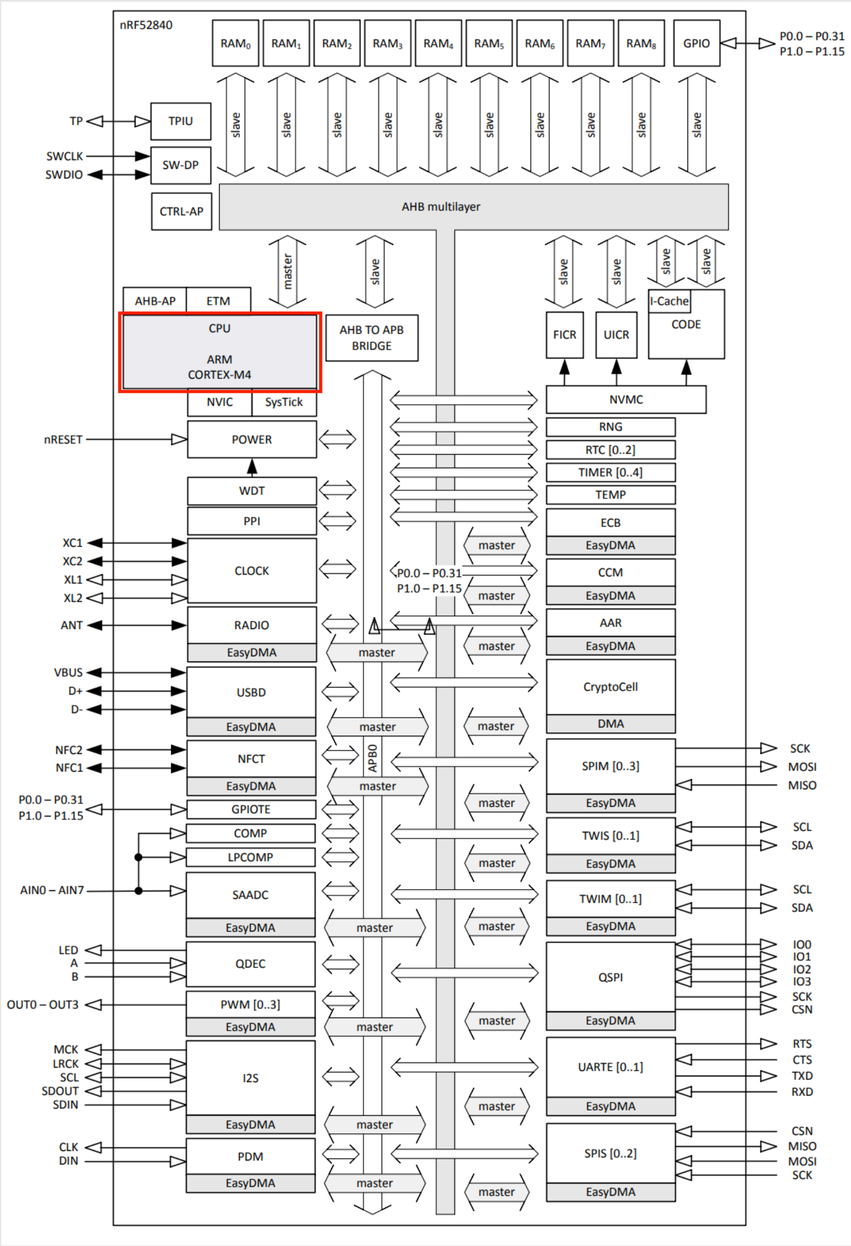
Training/Workshops

www.mab-labs.com

MOTIVATION

https://www.electronicproducts.com

- Heterogenous multi-processor systems
- Traditionally on distinct silicon dice
  - Communication using traditional busses
    - CAN
    - UART
    - SPI
- Each MCU responsible for a safety-critical component
- MPU responsible for non-critical components



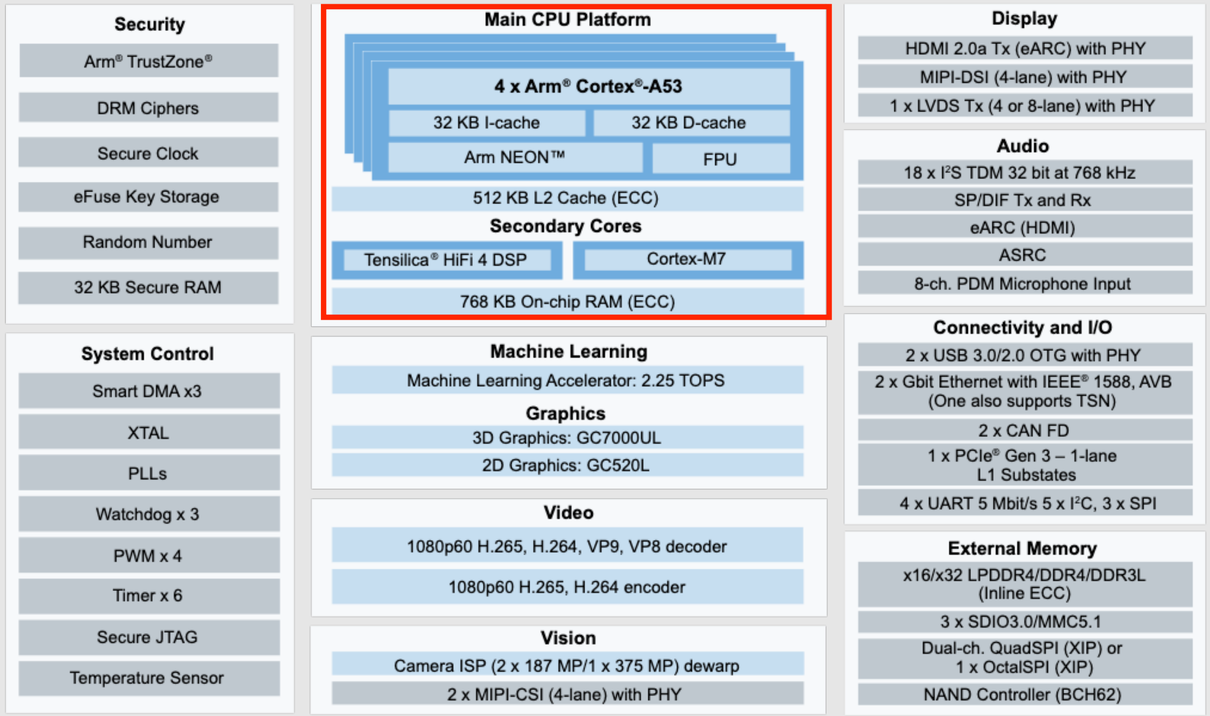
https://infocenter.nordicsemi.com/pdf/nRF52840_PS_v1.0.pdf

MOTIVATION

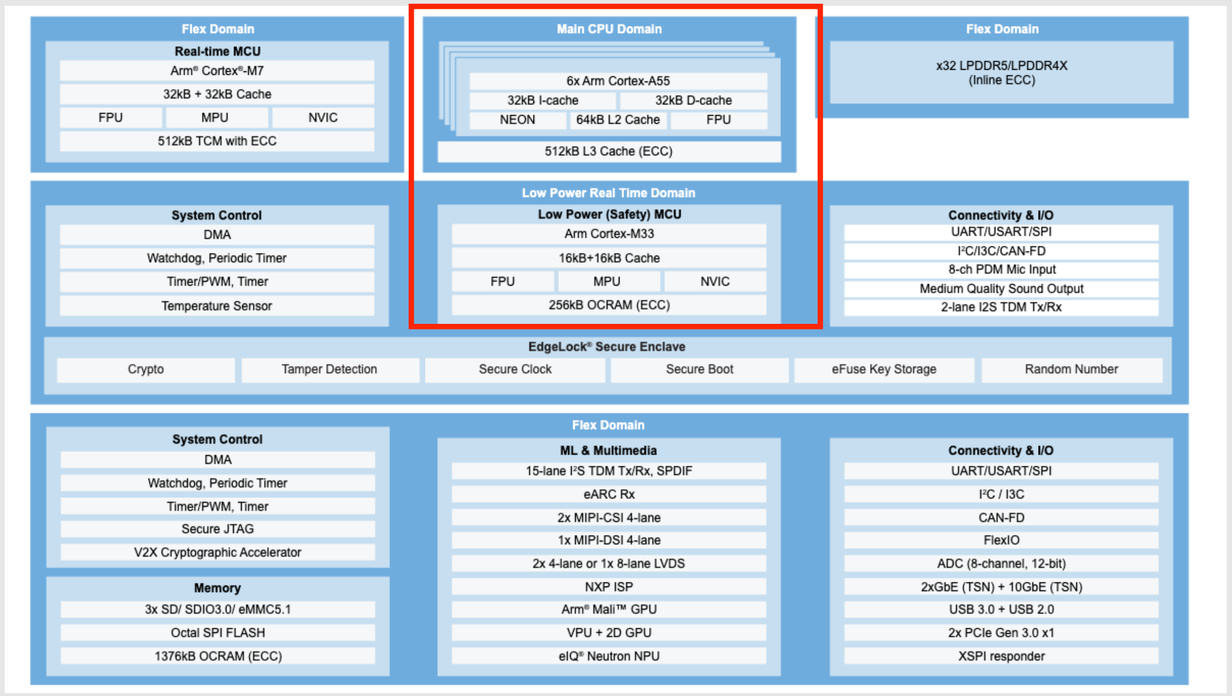
- Silicon density has exploded
- Single silicon die no longer just a single controller
- Entire **hardware** stacks now present on System-on-Chips (SoC)
  - Radio
  - USB
  - Peripherals
  - Etc
- Actual CPU itself occupies ~3% of silicon

MOTIVATION

- Add multiple microcontrollers and microprocessors to single die
- Can distribute the workload
  - MCU responsible for realtime operations
  - MPU responsible for non-realtime operations
- Robotics
  - MCU responsible for actuation
  - MPU responsible for GUI/display, communications, etc..



NXP iMX8M Plus



NXP iMX95

MOTIVATION

https://www.toradex.com/videos/taq-the-balancing-robot

Toradex
Swiss. Embedded. Computing.

ADVANTAGES

- Increased speed and reduced latency
  - Massive!
  - On-chip interconnects >> traditional protocols
- OTA simplified
  - MCU firmware resident on MPU filesystem
  - MCU OTA can be part of overall OTA

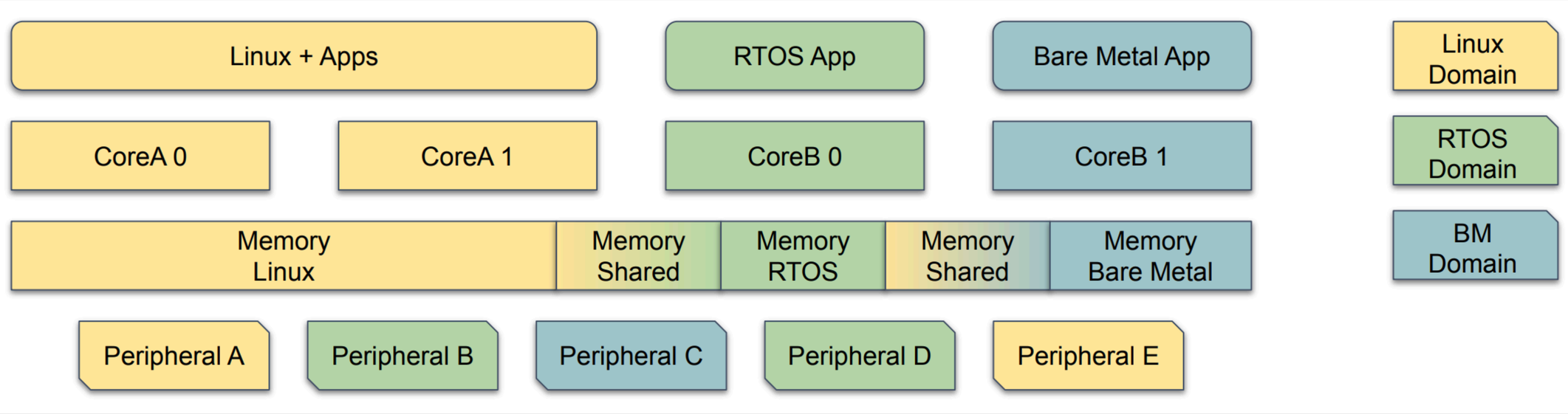
OPENAMP IN ZEPHYR

- OpenAMP (https://github.com/OpenAMP) **framework**
  - Multiple components
  - OpenAMP library
- Remoteproc and RPMsg
  - Core building blocks of library

OPENAMP IN ZEPHYR

- Remoteproc/RPMsg originally committed to Linux kernel by TI
  - Linux <—> SYS/BIOS (i.e. TI/RTOS)
- Picked up by OpenAMP
  - Expanded beyond just Remoteproc/RPMsg into a complete framework
- “OpenAMP Project”
  - openampproject.org

OPENAMP IN ZEPHYR



https://www.openampproject.org/docs/presentations/OpenAMP-Elevator-Pitch-2024-Q1.pdf

OPENAMP IN ZEPHYR

- Remoteproc
  - Parses ELF file
    - Ensure that it has appropriate memory segments to be loaded (**important later**)
  - Loads remote processor firmware
  - Starts the remote firmware

OPENAMP IN ZEPHYR

- RPMsg
  - IPC communication mechanism
  - Defines message format
  - RPMsg header
    - Defines source, destination, payload size

OPENAMP IN ZEPHYR

- VirtIO
  - Abstract out IPC
  - Use shared memory region between processors
- Metal
  - Abstraction between OpenAmp and underlying OS
  - Locks, interrupts, DMA, memory, sleep, etc…

OPENAMP IN ZEPHYR

- OpenAmp In Zephyr

```
[zephyrproject]$ ls -l modules/lib/open-amp/open-amp/
total 172
drwxrwxr-x 4 mab mab   4096 Jan  9 13:50 cmake
-rw-rw-r-- 1 mab mab    819 Jan  9 13:50 CMakeLists.txt
drwxrwxr-x 5 mab mab   4096 Jan  9 13:50 doc
-rw-rw-r-- 1 mab mab 122997 Jan  9 13:50 Doxyfile
drwxrwxr-x 8 mab mab   4096 Jan  9 13:50 lib
-rw-rw-r-- 1 mab mab   4696 Jan  9 13:50 LICENSE.md
-rw-rw-r-- 1 mab mab    524 Jan  9 13:50 MAINTAINERS.md
-rw-rw-r-- 1 mab mab  14983 Jan  9 13:50 README.md
-rw-rw-r-- 1 mab mab     54 Jan  9 13:50 VERSION
```

```
[zephyrproject]$ ls zephyr/lib/open-amp/
CMakeLists.txt  Kconfig  resource_table.c  resource_table.h
```

```
[zephyrproject]$ ls -l zephyr/samples/subsys/ipc/openamp_rsc_table/
total 24
drwxrwxr-x 2 mab mab 4096 Jan  9 13:50 boards
-rw-rw-r-- 1 mab mab  691 Jan  9 13:50 CMakeLists.txt
-rw-rw-r-- 1 mab mab  248 Jan  9 13:50 prj.conf
-rw-rw-r-- 1 mab mab 2272 Jan  9 13:50 README.rst
-rw-rw-r-- 1 mab mab  401 Jan  9 13:50 sample.yaml
drwxrwxr-x 2 mab mab 4096 Apr 17 12:46 src
```

Zephyr OpenAMP sample using resource table

OPENAMP IN ZEPHYR

- Relevant Kconfig options

```
CONFIG_IPM=y
CONFIG_OPENAMP=y
CONFIG_OPENAMP_RSC_TABLE_NUM_RPMSG_BUFF=8
CONFIG_OPENAMP_RSC_TABLE=y
CONFIG_OPENAMP_MASTER=n
```

OPENAMP IN ZEPHYR

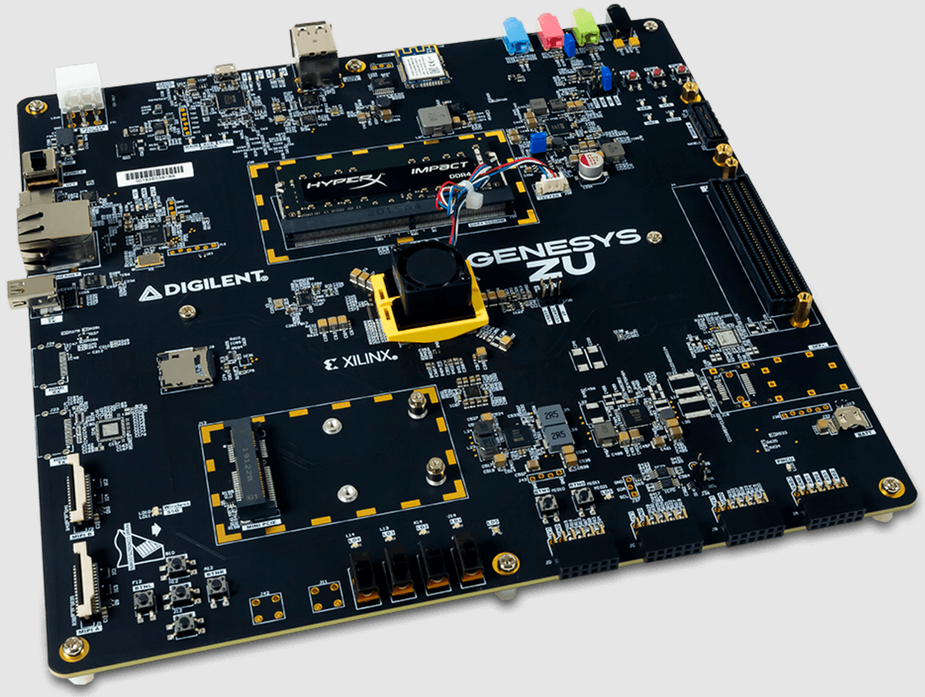
- Devicetree configuration

```
/*
 * Copyright (c) 2020, STMICROELECTRONICS
 *
 * SPDX-License-Identifier: Apache-2.0
 */

/ {
        chosen {
                /*
                 * shared memory reserved for the inter-processor communication
                 */
                zephyr,ipc_shm = &mcusram3;
                zephyr,ipc = &mailbox;
        };

        mcusram3: memory1@10040000 {
                compatible = "mmio-sram";
                reg = <0x10040000 DT_SIZE_K(64)>;
        };
};

&mcusram {
        reg = <0x10000000 DT_SIZE_K(256)>;
};
```

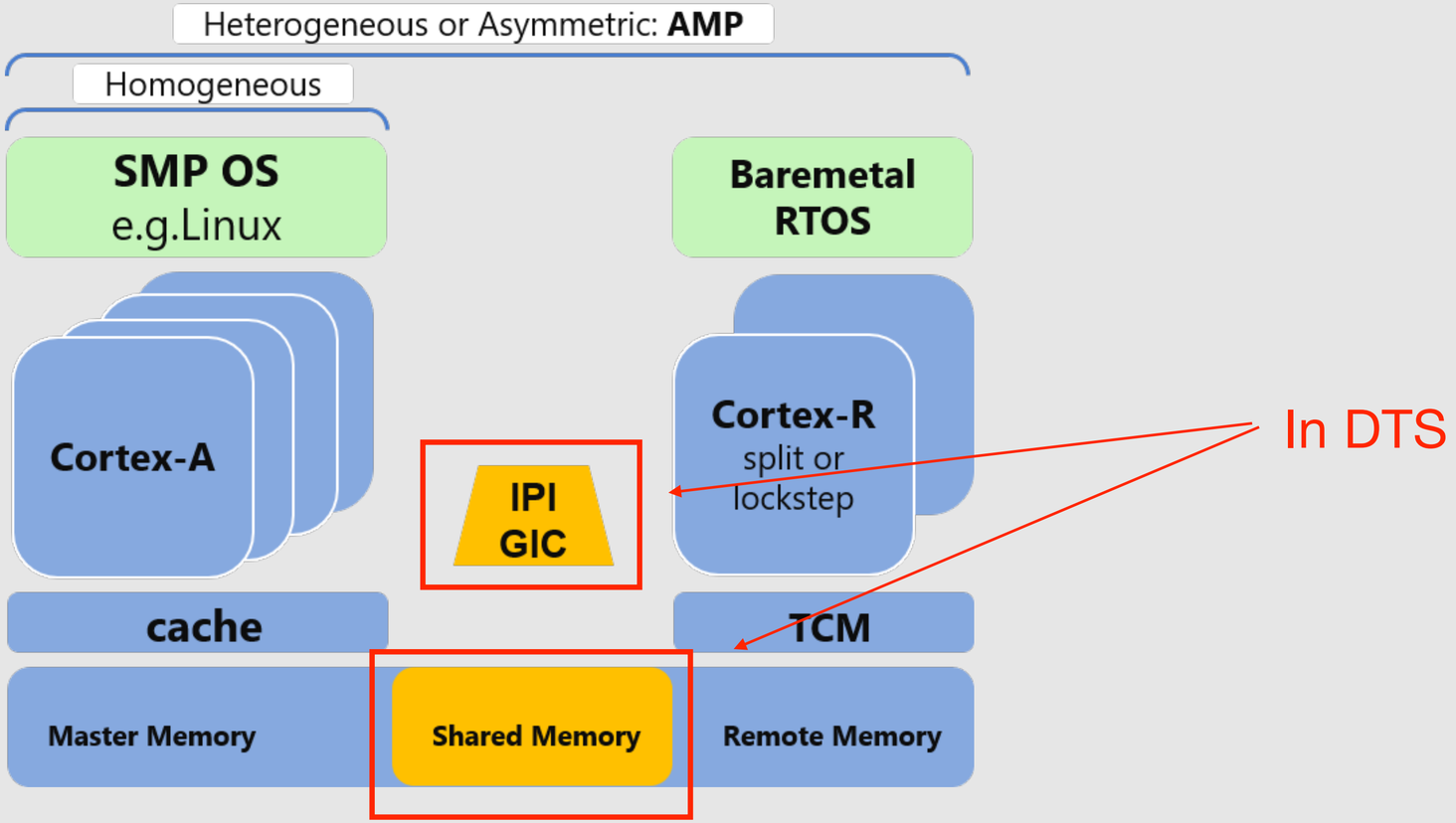
STM32MP157C_DK2

SYSTEM ARCHITECTURE

- Digilent Genesys ZU
  - Zynq MPSoC
    - Quad-core ARM Cortex A53
    - Dual-core ARM Cortex R5

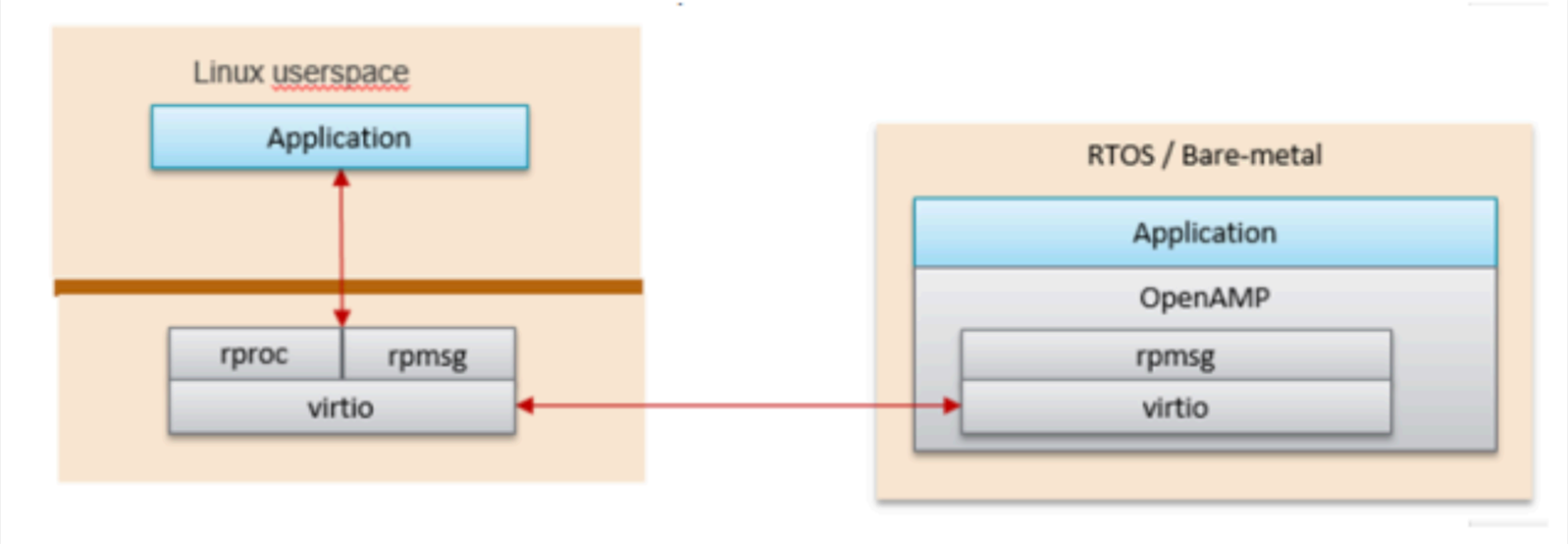


SYSTEM ARCHITECTURE



https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/18841718/OpenAMP

SYSTEM ARCHITECTURE



SYSTEM ARCHITECTURE

- Goal
  - Zephyr firmware for R5 on A53 root filesystem
  - Load Zephyr firmware from (Peta) Linux
  - “Start” Zephyr firmware from Linux
  - Send message from Linux to Zephyr
    - Echo back to Linux

DEVELOPMENT PROCESS

- Use KV260 R5 board in Zephyr as starting point
- Get blinky working
  - Confirm that clock configuration and pinout modifications are correct
  - Use Xilinx tools (i.e. Vitis) to load firmware
  - Use o-scope to confirm blinky frequency
- Test!

DEVELOPMENT PROCESS

- Add remoteproc drivers in Peta Linux

```
Device Drivers --->

Remoteproc drivers --->

# for R5:

<M> ZynqMP_r5 remoteproc support

# for Zynq A9

<M> Support ZYNQ remoteproc
```

DEVELOPMENT PROCESS

- Create recipe to add Zephyr binary to RFS

```
#
# This file is the openamp-test recipe.
#

SUMMARY = "Simple openamp-test application"
SECTION = "PETALINUX/apps"
LICENSE = "MIT"
LIC_FILES_CHKSUM = "file://${COMMON_LICENSE_DIR}/MIT;md5=0835ade698e0bcf8506ecda2f7b4f302"

SRC_URI = "file://zephyr.elf \
        "

S = "${WORKDIR}"

INSANE_SKIP_${PN} = "arch"

RDEPENDS_${PN} = " \
        libmetal \
"

do_install() {
             install -d ${D}/lib/firmware
             install -m 0644 ${S}/zephyr.elf ${D}/lib/firmware/zephyr.elf
}

FILES_${PN} = "/lib/firmware/zephyr.elf"
```

DEVELOPMENT PROCESS

- Use Vitis as a debugger
  - Copy over source files
  - Inform Vitis how to reconcile final binary with source files
    - Many-step process
    - Not focus of this talk (maybe another)
  - Have step through debugging!

DEVELOPMENT PROCESS

- After boot, instruct Peta Linux to load Zephyr firmware and start

```
echo zephyr > /sys/class/remoteproc/remoteproc0/firmware

echo start > /sys/class/remoteproc/remoteproc0/state
```

COMMON ISSUES AND RESOLUTIONS

- Issue when loading and starting firmware
  - `CONFIG_OPENAMP_RSC_TABLE=y`
    - Adds resource table section in final ELF
    - Checked by Linux kernel module

COMMON ISSUES AND RESOLUTIONS

- Segfault when sending commands from Linux to Zephyr
  - Zephyr IPC connected to mailbox in DTS?
  - Zephyr SRAM updated to reflect new value?
  - Zephyr SHM node connected to SRAM?

COMMON ISSUES AND RESOLUTIONS

```
rpu0_ipi: zynqmp-ipi@ff310000 {
        status = "disabled";
        compatible = "xlnx,zynqmp-ipi-mailbox";
        #address-cells = <1>;
        #size-cells = <1>;

        reg = <0xff310000 0x10000>;
        reg-names = "host_ipi_reg";
        interrupts = <GIC_SPI 33 IRQ_TYPE_LEVEL
                        IRQ_DEFAULT_PRIORITY>;
        local-ipi-id = <1>;

        rpu0_apu_mailbox: mailbox@ff990200 {
                remote-ipi-id = <0>;
                reg = <0xff990200 0x20>,
                      <0xff990220 0x20>,
                      <0xff990040 0x20>,
                      <0xff990060 0x20>;
                reg-names = "local_request_region",
                            "local_response_region",
                            "remote_request_region",
                            "remote_response_region";
        };
```

```
&rpu0_ipi {
        status = "okay";
};
```

```
zephyr,ipc = &rpu0_apu_mailbox;
```

COMMON ISSUES AND RESOLUTIONS

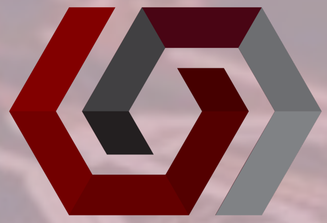
```
#ifndef SHARED_MEM_PA
#if XPAR_CPU_ID == 0
#define SHARED_MEM_PA  0x3ED40000UL
#else
#define SHARED_MEM_PA  0x3EF40000UL
#endif /* XPAR_CPU_ID */
#endif /* !SHARED_MEM_PA */

#ifndef SHARED_MEM_SIZE
#define SHARED_MEM_SIZE 0x100000UL
#endif /* !SHARED_MEM_SIZE */

#ifndef SHARED_BUF_OFFSET
#define SHARED_BUF_OFFSET 0x8000UL
#endif /* !SHARED_BUF_OFFSET */
```

```
&sram0 {
        reg = <0x3ed40000 0x100000>;
};
```

```
zephyr,ipc_shm = &sram0;
```

NEXT STEPS

- Upstream!
  - Add Digilent board to Zephyr
  - Add necessary board overlay to IPC sample
- Document steps to debug Zephyr in Vitis



THANK YOU!

Mohammed Billoo

MAB Labs Embedded Solutions

Zephyr Dev Summit 2024