SERENITY

ZynqMP-based board-management mezzanines for the Serenity ATCA-blades

L. Ardila, M. Balzer, M. Fuchs, T. Mehner, O. Sander,
M. Schleicher, D. Tcherniakhovski @ KIT

G. Fedi, G. Gimas, G. Iles, D. Parker,
M. Pesaresi, A. Rose @ IC

T. Schuh @ RAL

L. Calligaris, A. Cascadan, B. Casu @ SPRACE

Current revision ATCA hardware

two prototyping platforms with different features.

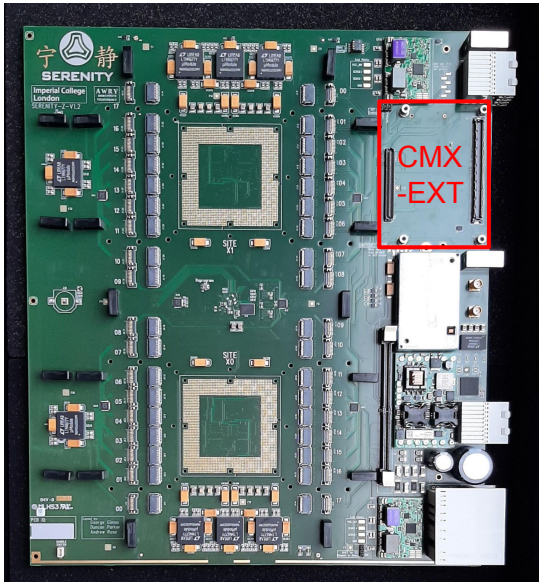
Serenity-Z:

- AUX connector outside CMX footprint
  - Fully compatible with PICMG com-express standard
  - Custom ZynqMP “CMX-EXT” can use both connectors
- IPMC functionality routed to both DIMM and AUX connector
- I2C and JTAG chains accessible via AUX connector

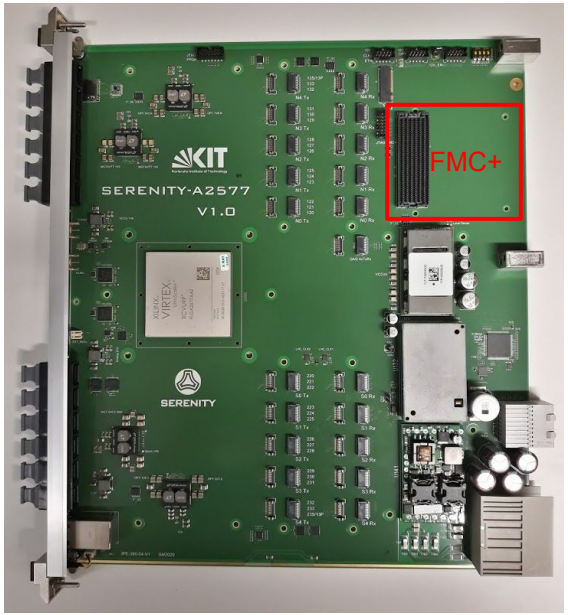
Serenity-A:

- FMC+ connector contains the IPMC signals; HA and I2C busses from the backplane
- JTAG, I2C, SPI and other required busses for management tasks available

Serenity-Z1.2



Serenity-A1.0



https://indico.cern.ch/event/921378/contributions/3912837/attachments/2067077/3469220/20200701_SoC_Meeting.pdf
https://indico.cern.ch/event/916720/contributions/3853811/attachments/2036066/3409066/2020-05-10_TK_DPS_v4.pdf

Custom ZynqMP SoM - Why?

SERENITY

Technical requirements (Serenity specific)

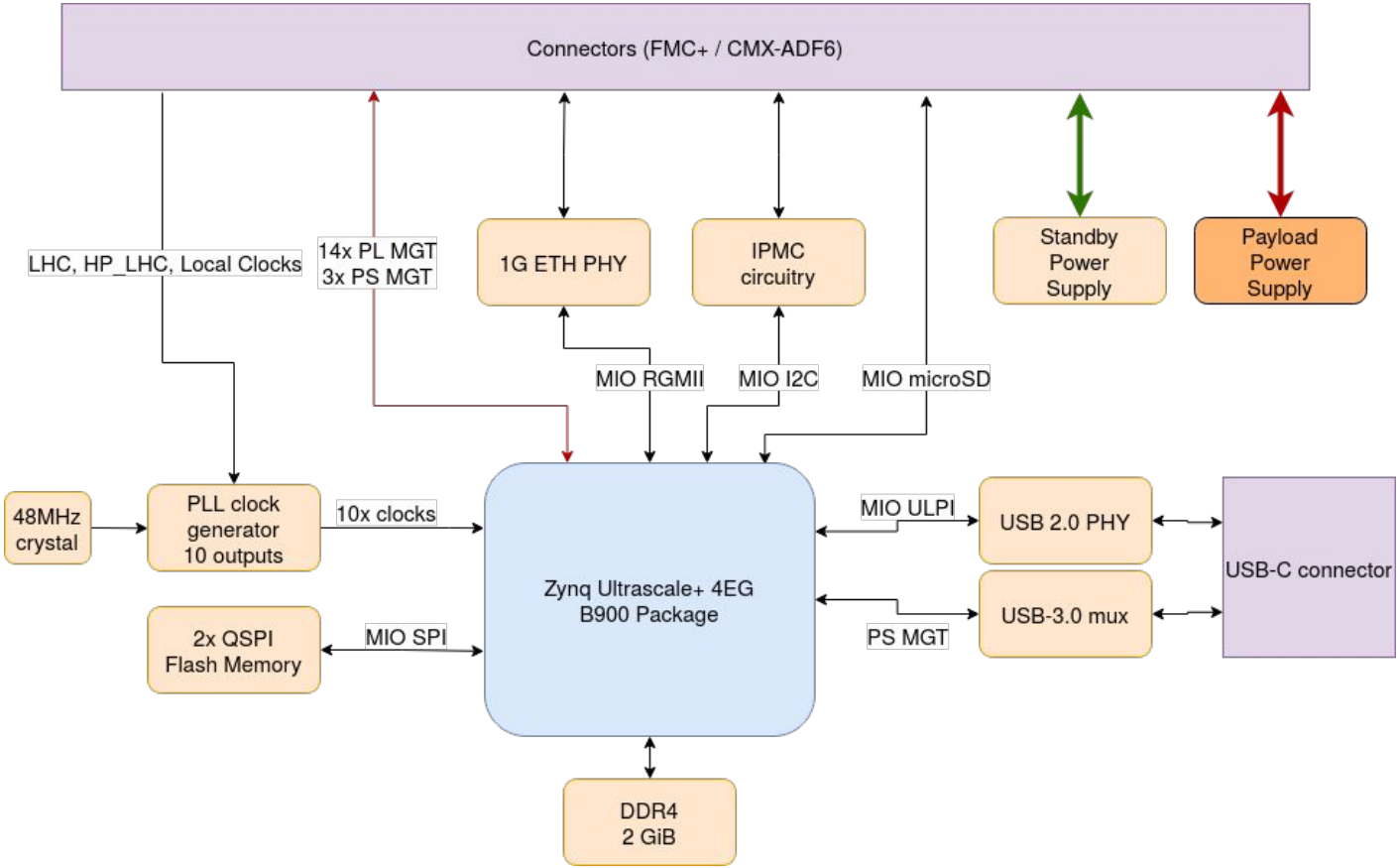
- compatibility with CMX and FMC+ form factor
  - flexible choice between x86 CMX boards and the ZynqMP SoM
- specific requirements related to the integration of IPMC into ZynqMP
  - individual powering of the domains (LPD, FPD, PL)
  - IPMB circuitry
- availability of high speed transceivers limited on commercial boards
  - ZU4EG with 16 lanes not available

Soft requirements

- full control of the design sources
- long term availability
- designed using KiCAD, an open source EDA software

Integrated ZynqMP SoM - Block Design

SERENITY



Integrated ZynqMP SoM - Layout

SERENITY

CMX-EXT module

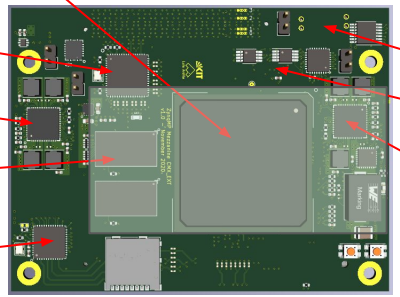
ZU4EG-B900

Si5395

IRPS5401 PMIC

2G DDR4

KSZ9031 ETH PHY



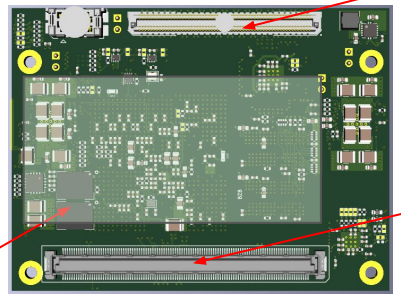
Integrated IPMC functionality

I2C IPMC I/O

USB PHY

IRPS5401 PMIC

QSPI FLASH



EXT
8 PL MGTs
2 PS MGT

CMX
6 PL MGTs
2 PS MGTs

FMC+ module

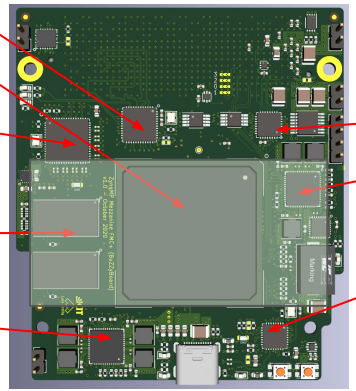
KSZ9031 ETH PHY

ZU4EG-B900

Si5395

2G DDR4

IRPS5401 PMIC



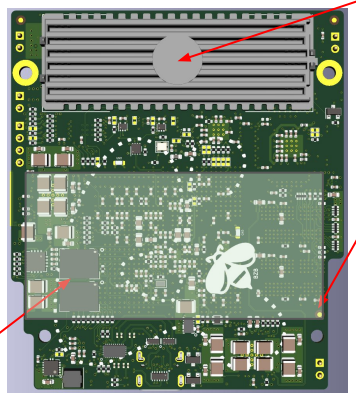
Integrated IPMC functionality

I2C IPMC I/O

IRPS5401 PMIC

USB PHY

QSPI FLASH



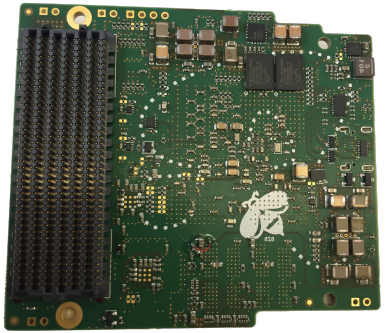
FMC+
5 PL MGTs
3 PS MGTs

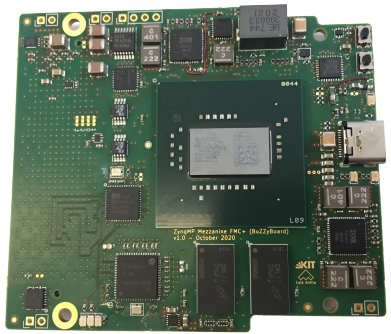
Highlighted area is common between them

https://indico.cern.ch/event/921378/contributions/3912837/attachments/2067077/3469220/20200701_SoC_Meeting.pdf

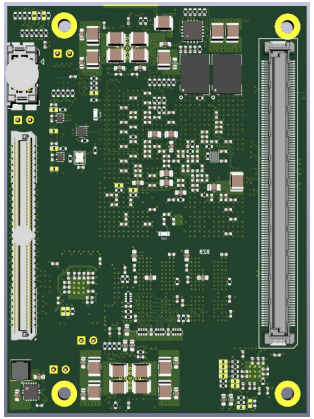
Zynq Ultrascale+ Mezzanines

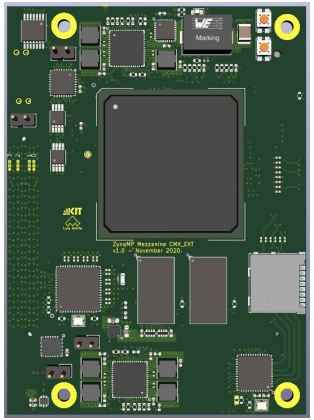
FMC+





CMX-EXT





- Both Mezzanines share the same stackup and routing for the MGT-CLKs, DDR4 interface, QSPI flash and Power delivery.
- Three FMC+ Mezzanines fabricated
- All interfaces of the SoM were tested, some bugs were found on the base-board and on the mezzanine, which should be fixed in a revision
- OpenIPMC firmware was ported to this platform with CentOS linux; standard booting from the SDcard and Network respectively

ZynqMP SoC DRAM-Layout-Verification

SERENITY

PCB layout DDR4 verification using the ZynqMP DRAM Diagnostics Test

- 2 GB Long-term-test
  - 3 TB written over 17h
  - **0 errors occured**
- 1 GB Long-term-test
  - 12 TB written over 55h
  - **0 errors occured**

Eye-Test

	Eye-Width in % (Average of 50 passes)	
	Read-Test	Write-Test
Byte-Lane 1	69.03	88.12
Byte-Lane 2	66.56	84.61
Byte-Lane 3	67.43	81.97
Byte-Lane 4	71.11	82.71

https://www.xilinx.com/html_docs/xilinx2019_1/SDK_Doc/SDK_references/sdk_u_zynq_dram.html

IPMCs

SERENITY

Commercial and **open source** solutions available

Pigeon Point IPMC software

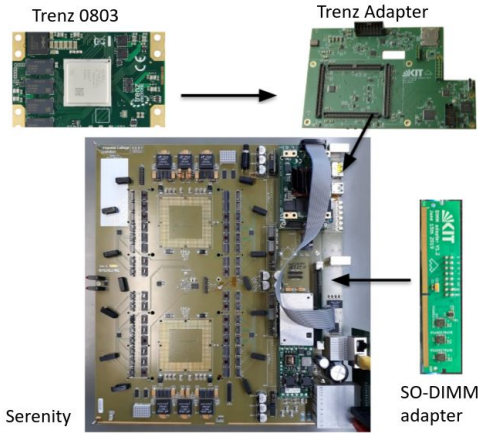
- Based on VPX version for ZynqMP (BMR-ZNQ-VPX)
- Extension by KIT and Pigeon Point for ATCA compliance

OpenIPMC software

- hardware independent software implemented on 7 platforms so far (ZynqMP x4, ESP32, STM32 x2), proven portability and hardware independence (based on FreeRTOS)
- subset of PICMG specifications developed and tested
- implemented into two “production” platforms
  - ZynqMP mezzanines with integrated solution (this presentation)
  - OpenIPMC-HW designed with DIMM format using STM32 microcontrollers

https://indico.cern.ch/event/1021679/contributions/4333600/attachments/2242916/3803473/20210511__xTCA_IG__OpenIPMC_Project.pdf

Polaris PICMG standard compliance tests



Pigeon Point IPMC on ZynqMP
via adapter on Serenity-Z1.1

PASSED	FAILED	SKIPPED	TOTAL
79 (58%)	34	24	137

OpenIPMC-HW on Serenity-Z1.1

PASSED	FAILED	SKIPPED	TOTAL
56 (56%)	17	26	99

using an ATCA compliance testing SW by Polaris Networks kindly provided by the CERN EP-ESE group at bldg 14.

https://indico.cern.ch/event/921378/contributions/3912837/attachments/2067077/3469220/20200701_SoC_Meeting.pdf
https://indico.cern.ch/event/1021679/contributions/4333600/attachments/2242916/3803473/20210511__xTCA_IG__OpenIPMC_Project.pdf

Current configuration

- R5 runs OpenIPMC-software
- A53-complex runs Petalinux Kernel with CentOS Root-Filesystem
- Programmable Logic
  - AXI-Chip2Chip-Master for EMP-Framework
- in the near future we hope this board can be as well tested for PICMG compliance

Highlight: ZynqMP Split PSU Config

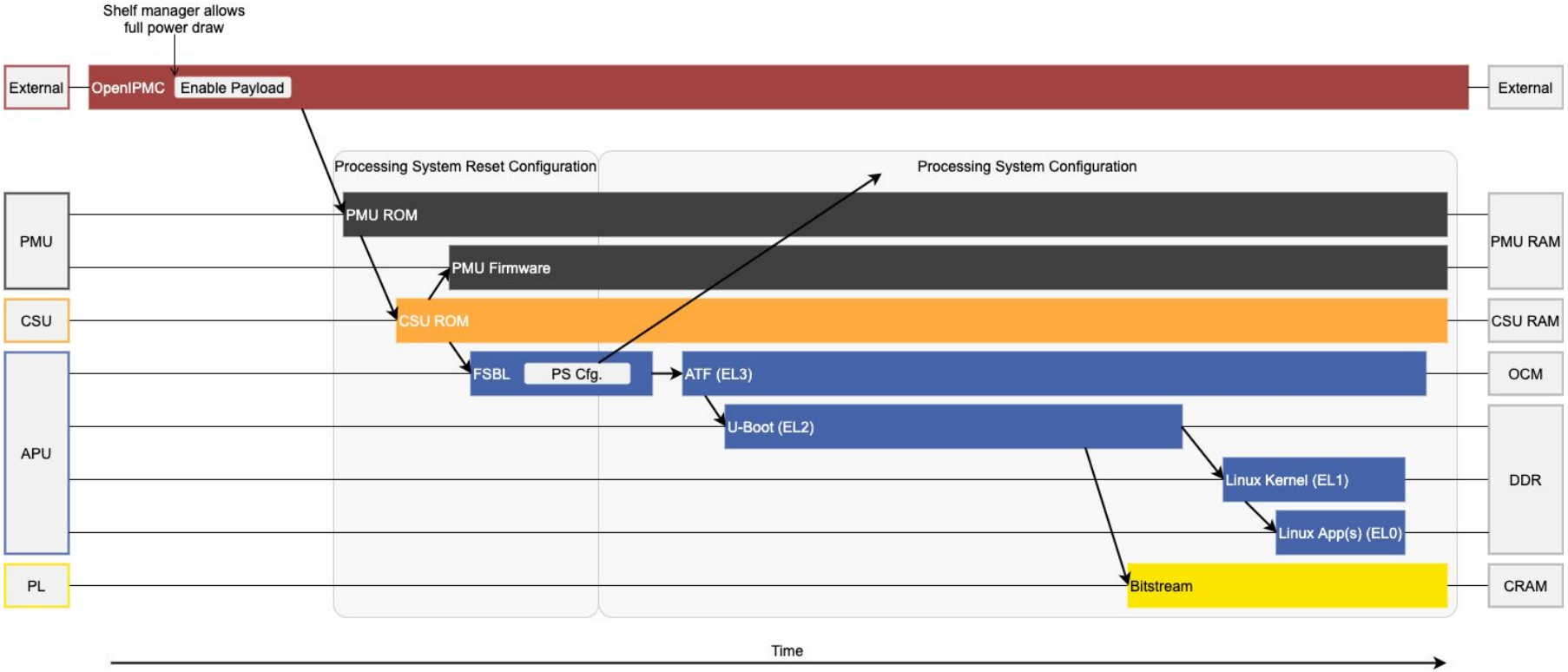
- Potentially hundreds of ZynqMP to boot at a powerup in CMS applications
- There is a trend to get as much configuration as possible from the network
  - Linux kernel → ok, can be loaded via tftp by uboot
  - Root filesystem → ok, can be accessed as NFS rootfs
  - Bitstream → ok, can be loaded and configured via tftp by uboot
- However, initialization of the Processing System is done very early(1) at boot.
- It includes
  - MIO/EMIO configuration
  - Peripheral configuration
  - Clock configuration
  - Serdes configuration
  - PS/PL interfaces
  - Isolation, MPU configuration
  - ...

Consequence
A lot of **application specific information** from the Vivado project needs to be **stored locally** on the ZynqMP board (e.g. QSPI, emmc, sdcard) and can not be fetched from network.
= not good maintainability / updateability

Good news: We found a way to fetch that information via network during the boot process.

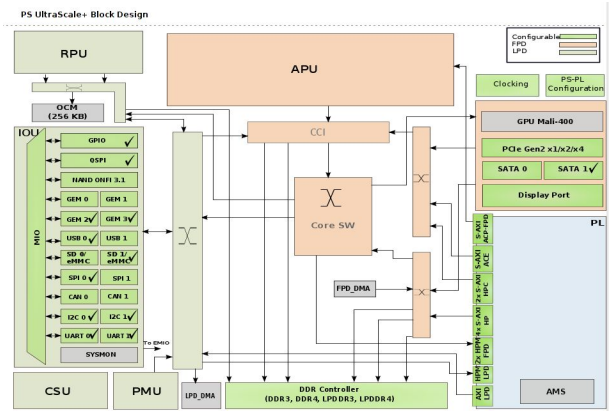
(1) for experts: FSBL stage one as psu_init which is generated by Vivado

Deep dive: ZynqMP boot process

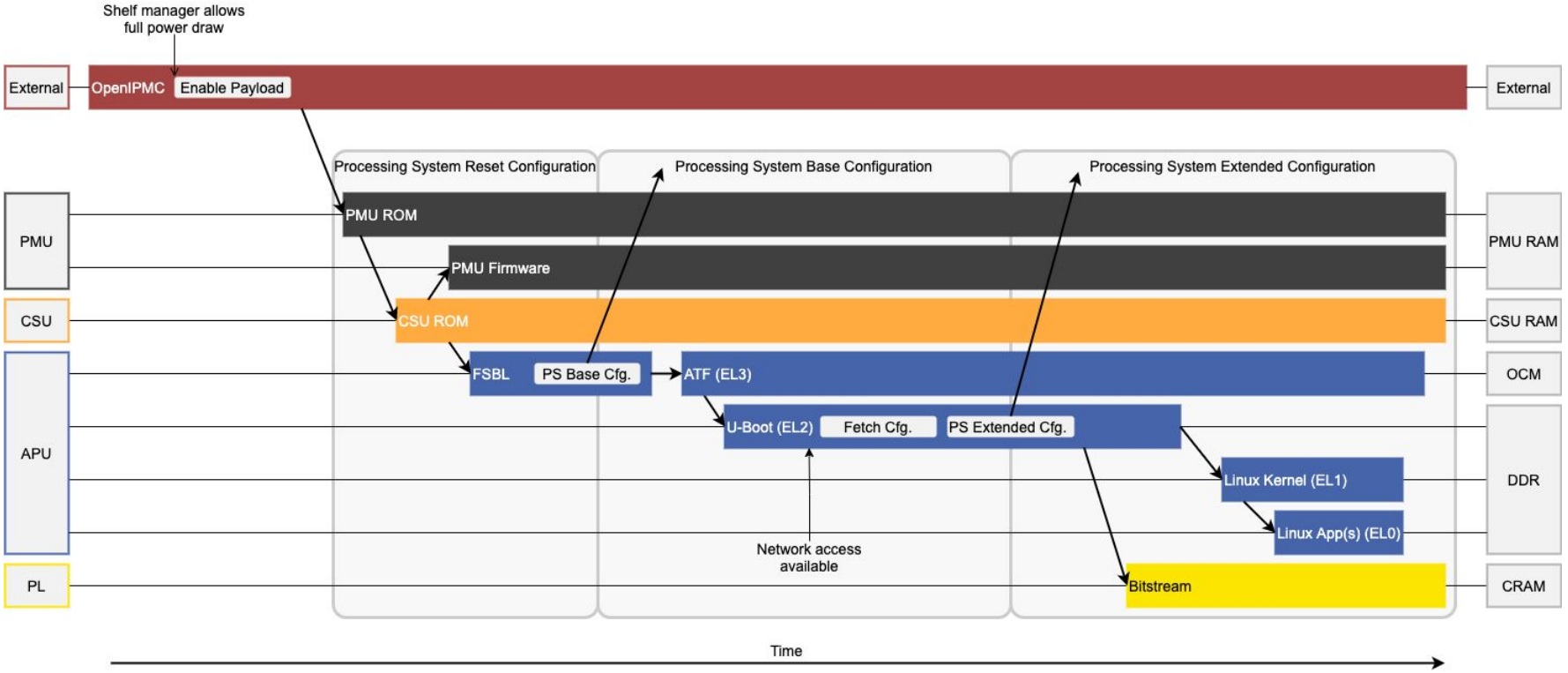


ZynqMP Split Boot - principle idea

- Idea: Separate configuration in two stages
- First stage (FSBL) loads basic configuration
  - **Minimal Vivado PS configuration** (e.g. DDR, ETH, UART)
  - allows to execute FSBL + ATF + uboot
- Second stage (uboot) loads full configuration
  - tftp load and apply **full application specific Vivado PS configuration**
  - SoC is mostly reconfigurable (exception are parts already in use like DDR)
  - Bitstream, linux kernel and rootfs are already tftp / nfs capable

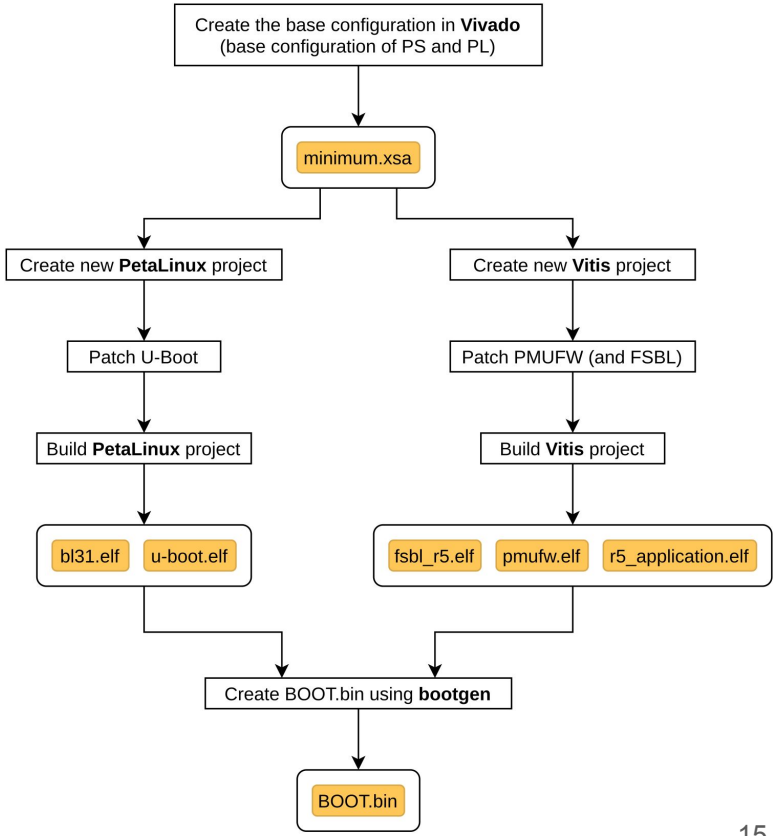


ZynqMP Split Boot in detail



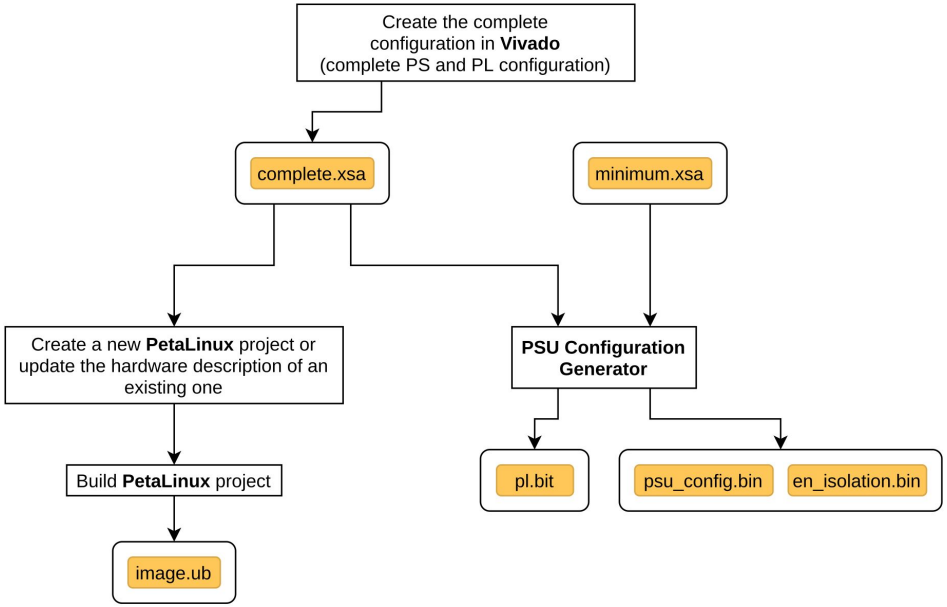
Toolflow - Base Config

- Create Vivado-project
  - Basic PS- and PL-configuration
  - Export hardware
- Generate executables using hardware
  - New: patch needed functions into U-Boot source in PetaLinux-project
  - New: patch PMU firmware and FSBL
  - PetaLinux creates ATF and U-Boot
  - Vitis creates FSBL, PMU-Firmware, IPMC
- Generate boot-file
  - Combine executables with bootgen

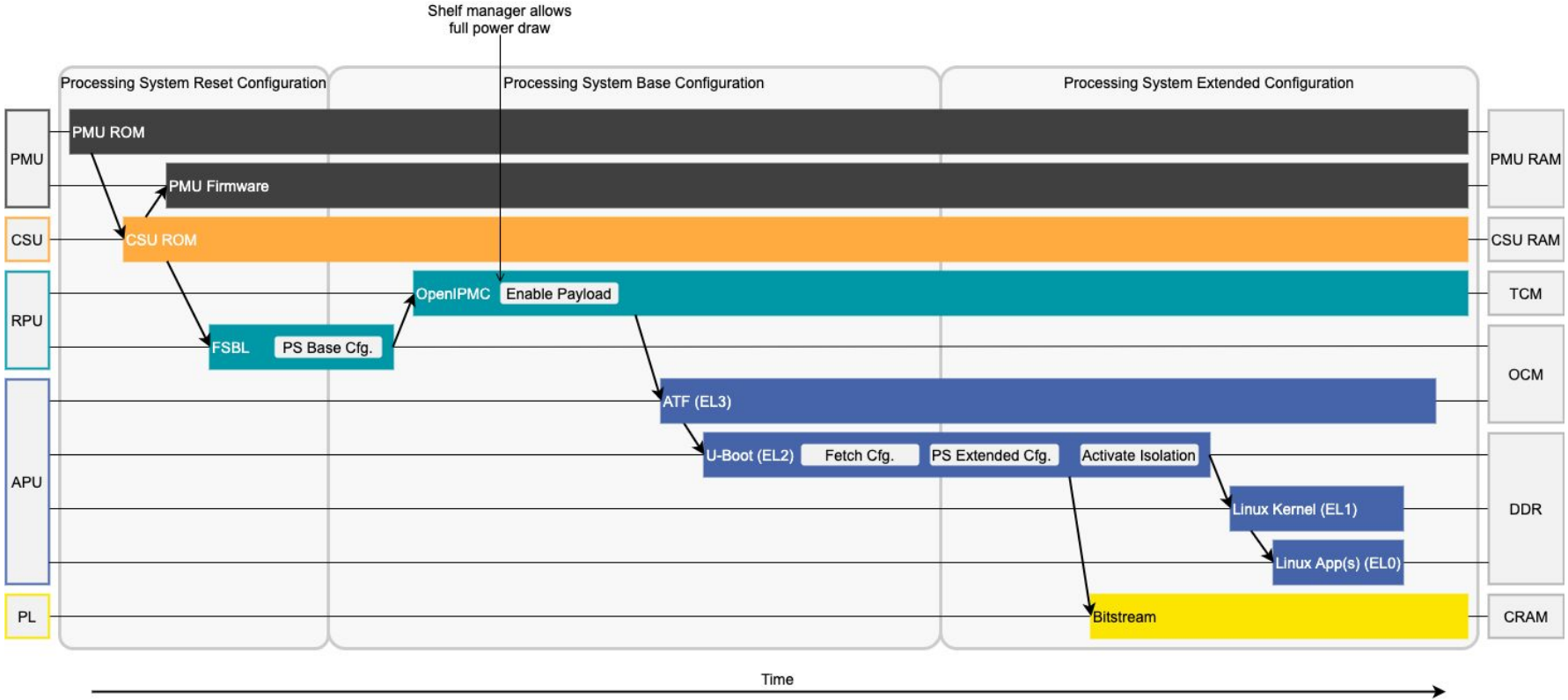


Toolflow - Extended/Full Config

- Create complete Vivado-project
  - Complete PS- and PL-configuration
  - Export hardware
- Build Petalinux
- New: PSU-Config-Generator
  - Compares full- and base-configuration
  - Compiles difference to config-files
- Deploy image, config and bitstream
  - Upload to TFTP-server



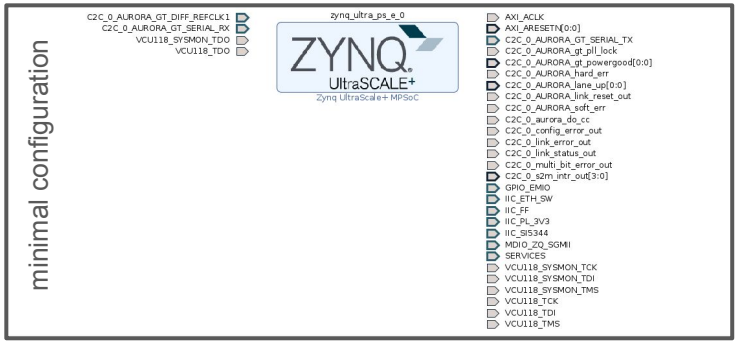
ZynqMP Split Boot with internal OpenIPMC

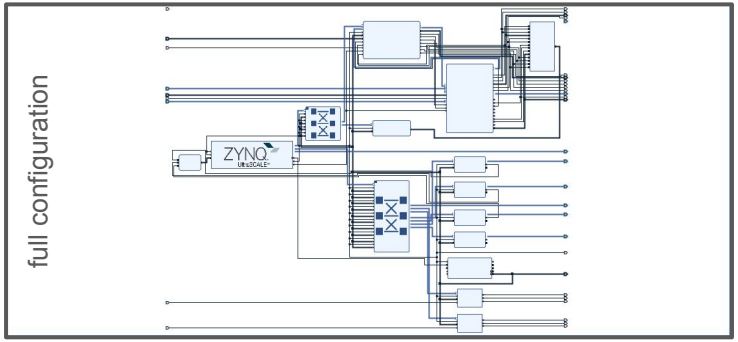


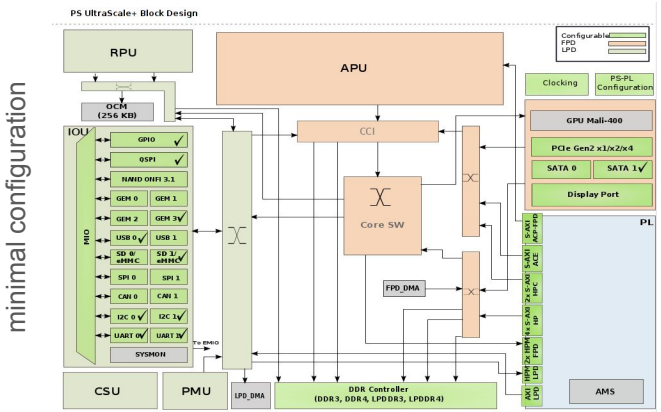
Conclusion

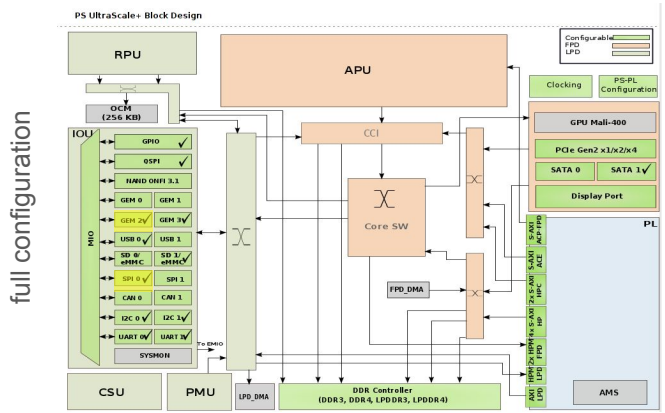
- Two ZynqMP mezzanines have been designed to serve the board management functionalities of the Serenity ATCA boards.
- One of those mezzanines was fabricated, some bugs were found and fixed in preparation for a revision fabrication of both form factors.
- Both commercial and open-source IPMCs have been demonstrated to run using the integrated approach on the ZynqMP devices.
- The subset of implemented functions in OpenIPMC was proven to be conformant with the standard, benefiting from its open nature, platform independence and configurability
- A split boot mode has been designed with the aim of reducing the application-specific PS configuration stored on the early boot images, stored locally in the ZynqMP. This mechanism allows the full configuration to be loaded via the network.

ZynqMP Minimal vs Full Configuration









ZynqMP FMC+ Power

+3V3_STBY @ 2.68A (7.43W) (67of 11W)

- VDDO+VDDA_Si5395@ 0.433A
- IRPS5401MTRPBF @ 2.25A (70eff)
  - 3) +2V5_STBY @ 0.060A
    - VPP_DDR4x2 @ 0.060A
  - 3) +1V2_STBY @ 1.235A
    - VCCO_PSDDR_504 @ 0.354A
    - VDD_DDR4x2 @ 0.660A
    - ETH_PHY @ 0.221A
  - 1) +0.85V_STBY @ 2.62A
    - VCCINT_IO @ 0.058A
    - VCCRAM @ 0.011A
    - VCC_PSINTFP @ 1.062A
    - VCC_PSINTLP @ 0.171A
    - VCC_PSINTFP_DDR @ 0.677A
    - ▒▒▒ - PS_MGTRAVCC @ 0.641A
  - 2) +1V8_STBY @ 0.690A
    - VCCAUX @ 0.117A
    - VCCAUX_IO @ 0.041A
    - VCC_PSAUX @ 0.002A
    - VCCO @ 0.053A
    - ▒▒▒ - VCC_PSDDR_PLL @ 0.026A
    - ▒▒▒ - VCC_PSADC @ 0.011A
    - ▒▒▒ - VCCADC @ 0.008A
    - ▒▒▒ - PS_MGTRAVTT @ 0.1A
    - VDD_Si5395 @ 0.270A
    - VCC_QSPI @ 0.062A
  - 2) +1V2_PS_PLL @ 0.026A
    - VCC_PS_PLL @ 0.026A

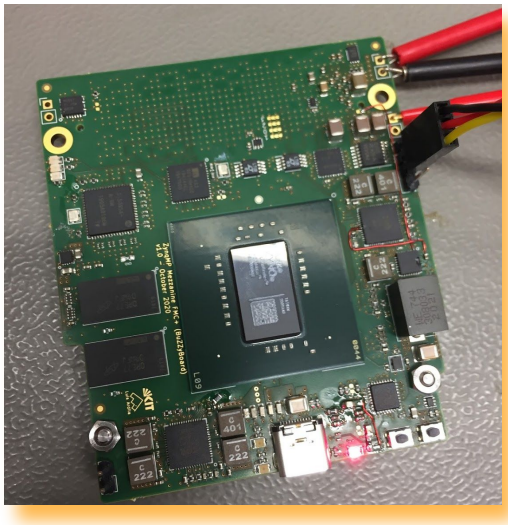
12V @ 2 A

- IRPS5401MTRPBF
  - 1) +0.85_PL_VCCINT
    - VCCINT @ 6A
  - 4) +5V_USB_VBUS
  - 3) +1V2_PL_MGTAVTT
    - MGTAVTT_R @ 1.892A
  - 2) +0.9V_PL_MGTAVCC
    - MGTAVCC @ 1.339A
  - 3) +1V8_PL_MGTVCCAUX
    - MGTAVCCAUX_R @ 0.049A

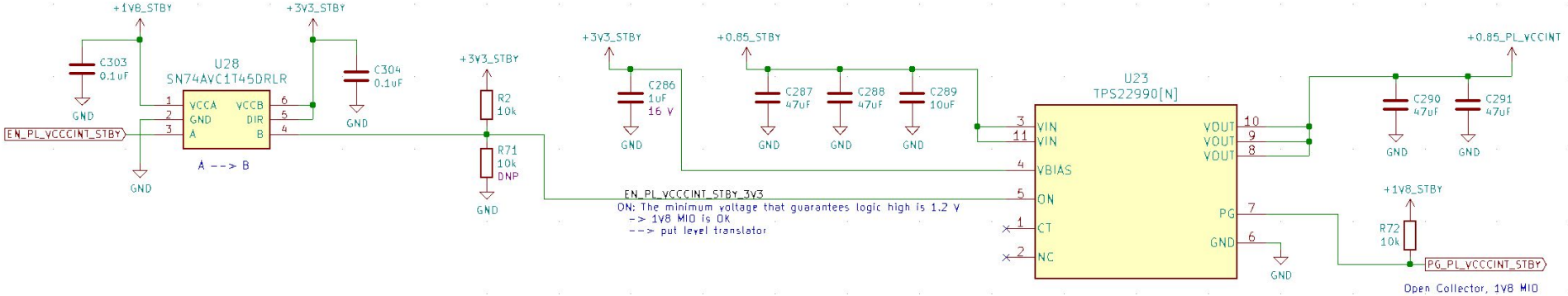
Case1: 3V3_STBY ON, DC-DC unconfigured
3V3_STBY = 208mA

Case2: DC-DC configured
3V3_STBY = 743 mA (2.45W)
12V = 72.8 mA

Case3: DC-DC configured Plugged-In
3V3_STBY = 743 mA (2.45W)
48V = 127mA (6.1W)



ZynqMP FMC+ Power



The PL_VCCINT 0.85V rail can be supplied from the 3V3_Standby power by using the power switch TPS2290N (U23)

Quiescent current for PL_VCCINT is ~700mA for 4EG, 5EG and ~ 1200mA for 7EG, currently with 4EG total current in 0.85V is 2A (4A possible)

OpenIPMC and CentOS on ZynqMP

- A53-complex runs Petalinux Kernel with CentOS Root-Filesystem
- R5 runs IPMC-software
  - OpenIPMC
- Programmable Logic
  - AXI-Chip2Chip-Master for EMP-Framework