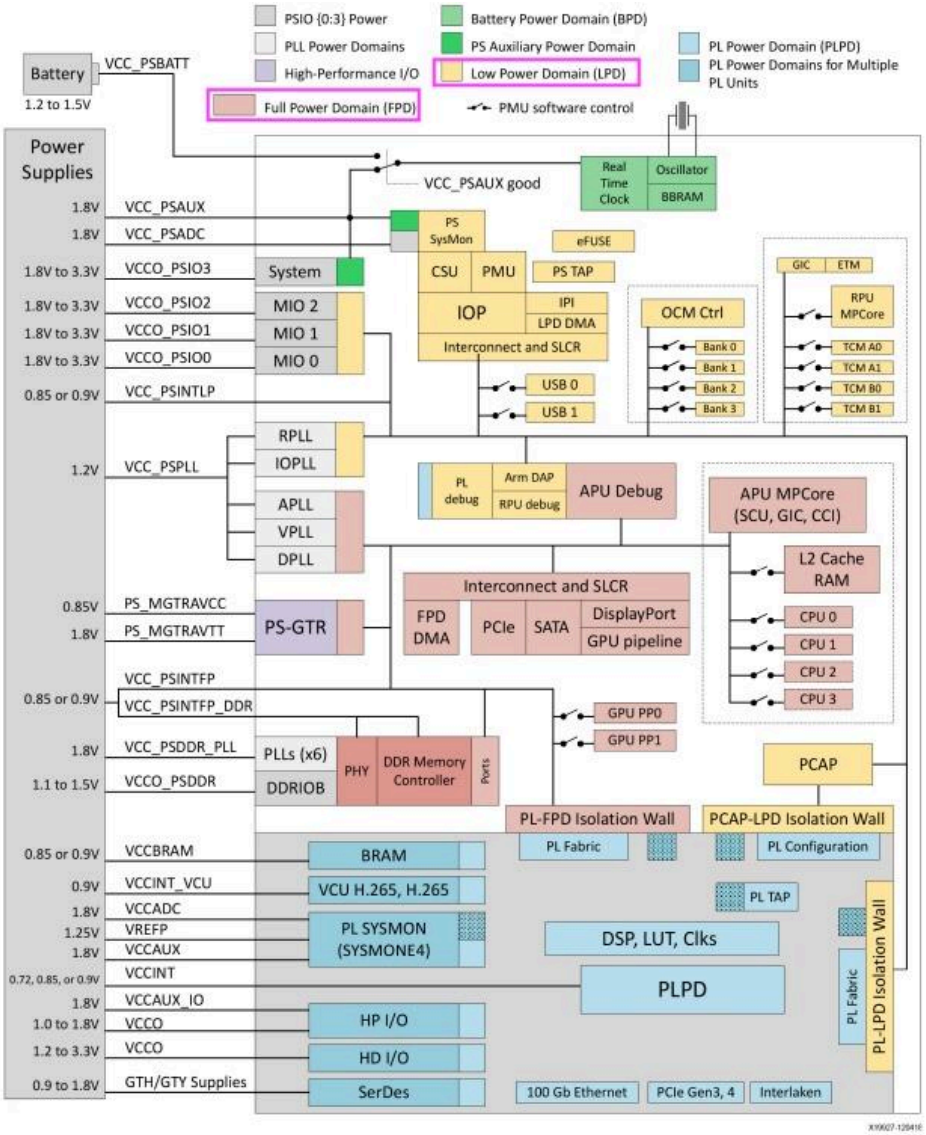
Purple - devices we are **KEEPING**

- These are based on the minimum device list given to us

Yellow - devices we are **REMOVING**

- The logic for removing these comes from research + follow-up notes at the end of the end of this doc by Uzeir, a sponsor.
    - Main logic:
        - Only using some devices in the low-power domain (LPD), so removing those



        - When using third party devices and not Xilinx provided ones, remove the Xilinx provided ones

*A lot of these device names have different labels (ie dma-controller@<address> would be labeled fpd_dma_channel1 or something similar instead) in source files before they are compiled, and those were referenced as well in pursuit of matching the names of devices in the given minimum list to devices in the device tree made for the ZCU106 board

- Cpus - execute code
    - Idle_states - gives ability to sleep CPU for power saving

- Opp-table-cpu - performance level info for power management and dynamic voltage and frequency scaling

- Zynqmp_ipi - zynqmp inter-processor interrupt; allows diff processors within the SoC to interrupt each other

    - mailbox@ff990400 - used for communication between processors; able to use this instead of shared memory

- Dcc - device configuration core; device initialization and configuration
    - ALREADY has status set to disabled

- Pmu - platform management unit; facilitates isolation of power domains on ZCU106
    - Each power domain can be individually isolated. The platform management unit (PMU) on the LPD facilitates the isolation of each of the power domains. Additionally, the isolation can be turned on automatically when one of the power supplies of the corresponding power domain is accidentally powered down. Since each power domain can be individually isolated, functional isolation (an important aspect of safety and security applications) is possible. See This Figure.
    - https://docs.amd.com/r/en-US/ug1085-zynq-ultrascale-trm/Power-Domains-and-Islands

- Psci - power state coordination interface; defines a set of standaridzed power states a processor can enter

- Firmware

    - Zynqmp-firmware

        - Zynqmp-power; various things related to power management (saving power)

        - Nvmem_firmware - non-volatile memory firmware

            - Soc_revision@0 - keeps track of the version of this SoC (to indicate what version of software the kernel should use for this chip)

            - Efuse_dna@c - eFuse Device DNA; Device DNA is a unique identifier for the device and this keeps track of it

- Efuse_user<multiple of these>@<also multiple addresses> - these can be user-defined storage areas, apparently provide "secure and tamper-evident storage" according to chatGPT, used for storing cryptographic keys, unique IDs, etc.

- Efuse_miscusr@40- same as with efuse_usr

- Efuse_chash@50 - not sure, I assume it stands for cryptographic hash; will assume it performs similar to the above two efuses

- Going to assume all the below are somewhat related to the general efuse security concept and will be used on the board

    - Efuse_pufmisc@54
    - Efuse_sec@58
    - Efuse_spkid@5c
    - Efuse_ppk0hash@a0
    - Efuse_ppk1hash@d0

- Pcap- Programmable Logic Configuration Access (PCAP); helps configures the FPGA

- Zynqmp-aes - Zynqmp- advanced encryption standard; the AES engine used for cryptographic operations like encryption and decryption

- Reset-controller - controls resets of the zynqMP SoC

- Pinctrl - manages configuration and control of GPIO pins for things like UART, SPI, I2C controllers, etc;

    - I2c0-default ***i2c0_dev** - inter-integrated circuit 0- default; is a controller that manages communication between the processor (SoC?) and peripheral devices connected to the I2C bus (I2C is a communication protocol)

        - Mux- pin multiplexing; specifies which pins are associated with the I2C0 controller

        - Conf- responsible for additional pin settings like electrical and functional stuff
        -
    - I2c0- gpio ***i2c0_dev**- i2c0 general purpose input/output; this uses GPIO pins to manually replicate the i2c protocol and is normally used when there is no dedicated i2c pins

        - mux -
        - Conf-
    - i2c1-default ***ic21_dev**
        - Mux
        - Conf
    - I2c1-gpio - same as before
    - Uart0-default ***uart0_dev** (universal asynchronous receiver transmitter) allows for asynchronous communication between devices; UART port connects ZCU106 board to computer to be able to have a terminal that can control it
    - uart1-default ***uart1_dev**
    - usb0-default ***usb0_dev**
    - gem3 ***gem3_dev** (Gigabit Ethernet MAC controllers -- ethernet connectivity)
    - can1-default ***can1_dev**
    - Sdhci1-default- allows communication w/ SD card and its variants
    - Gpio-default - default settings for the GPIO pins
        - Sha384 - cryptographic hash func; used to ensure data hasnt been altered
        - Zynqmp-rsa - zynqMP RSA- hardware accelerated RSA (encryption)
        - Gpio- is the GPIO pin controller, so manages things like reading or writing pin values, etc ***gpio_dev**
        - Clock-controller
- timer
- Edac- error detection and correction; used to detect and correct errors that occur during data transmission or storage
- Fpga_full
- Axi- advanced extensible interface; used for high speed interconnects between components in SoC and FPGA-based designs
    - can@ff060000 -Controller Area Network; used to connect microcontrollers and devices without a host computer ***can0_dev**
    - can@ff070000 ***can1_dev**

- cci@fd6e0000- control clock interface; designed to control and manage clock signals within the chip ***cci_dev-**
  - pmu@9000
- dma-controller@fd500000- direct memory access; allows data transfer without having to access the CPU ***fpd_dma_chan1**
- dma-controller@fd510000 ***fpd_dma_chan2**
- dma-controller@fd520000 ***fpd_dma_chan3**
- dma-controller@fd530000 ***fpd_dma_chan4**
- dma-controller@fd540000 ***fpd_dma_chan5**
- dma-controller@fd550000 ***fpd_dma_chan6**
- dma-controller@fd560000 ***fpd_dma_chan7**
- dma-controller@fd570000 ***fpd_dma_chan8**
- interrupt-controller@f9010000 ***gic_gic400_dev**
- gpu@fd4b0000 ***gpu_dev**
- dma-controller@ffa80000 - these dmas are in the LPD (low power domain)
- dma-controller@ffa90000
- dma-controller@ffaa0000
- dma-controller@ffab0000
- dma-controller@ffac0000
- dma-controller@ffad0000
- dma-controller@ffae0000
- dma-controller@ffaf0000
LPD and FPD DMA units	Programmable number of outstanding transfers, support for simple and scatter-gather mode, support for read-only and write-only DMA mode, descriptor prefetching, per channel flow control interface.
  - https://docs.amd.com/r/en-US/ug1085-zynq-ultrascale-trm/Functional-Units-and-Peripherals
- memory-controller@fd070000 (mc) **ddrc_dev**
- nand-controller@ff100000 (nand0)
- ethernet@ff0b0000 (gem0)- ethernet interfaces implemented using third party “IP cores” instead of Xilinix’s “GEM IP Core”
- ethernet@ff0c0000 (gem1)
- ethernet@ff0d0000 (gem2)
- ethernet@ff0e0000 ***gem3_dev**
  - ethernet-phy@c
- gpio@ff0a0000 ***gpio_dev**
- i2c@ff020000 ***i2c0_dev**
  - Contains a multitude of sub-devices, including more gpio, i2c’s, etc
- i2c@ff030000 ***i2c1_dev**
  - Same as above
- memory-controller@ff960000 ***ocm_dev**
- perf-monitor@ffa00000 (perf_monitor_**ocm**) ***these are in the LPD so we get rid of them, see below pics**
- perf-monitor@fd0b0000 (perf_monitor_ddr)

- perf-monitor@fd490000 (perf_monitor_cci) ***apm_fpd_dev *This one and below are in the FPD (see pics below)**
- perf-monitor@ffa10000 (perf_monitor_lpd)

Table: APM Units, except the DDR_APM has six slots corresponding to the six DDR memory controller ports.

Table: APM Units

Unit Name	Number of Counters	Power Domain	Clock	Register Set	Location
DDR_APM	10	FPD	TOPSW_LSBUS_CLK	APM_DDR	Six AMD AXI port interface (XPI) data ports on the DDR memory controller.
CCI_APM	8	FPD	TOPSW_LSBUS_CLK	APM_CCI_INTC	AXI channel from the CCI to the main switch.
OCM_APM	8	LPD	LPD_LSBUS_CLK	APM_INTC_OCM	AXI channel from the OCM switch to the OCM memory.
LPD_APM	8	LPD	LPD_LSBUS_CLK	APM_LPD_FPD	AXI channel from the LPD switch to the FPD main switch.

APM_CCI_INTC	APM	0xFD490000	AXI Performance Monitor, Performance monitor
APM_DDR	APMDDR	0xFD0B0000	AXI Performance Monitor, Performance Monitor
APM_INTC_OCM	APM	0xFFA00000	AXI Performance Monitor, Performance Monitor
APM_LPD_FPD	APM	0xFFA10000	

- pcie@fd0e0000 (pcie)* - peripheral component interconnect express; like a bus but has individual lanes between PCIE root and component so theres no contention of the bus ***axipcie_dev**
- spi@ff0f0000- serial peripheral interface; synchronous serial communication ***qspi_dev**
- phy@fd400000- manages physical layer signaling? Deals with electric signals of underlying physical medium, associated with Ethernet, USB, and SATA typically ***serdes_dev *siou_dev**
- rtc@ffa60000- real time clock ***rtc_dev**
- ahci@fd0c0000 (sata)- manages communication between CPU and SATA storage devices like SSDs and HHDs
- mmc@ff160000 (sdhci0)- allows communication w/ multimedia card ***sd0_dev**
- mmc@ff170000 (sdhci1) ***sd1_dev**
- smmu@fd800000 (smmu)- system memory management unit; manages memory access and translation for peripheral devices ***smmu_gpv_dev**
- spi@ff040000 (spi0) - less good version of QPSI- assuming QSPI replaces this?
- spi@ff050000 (spi1)
- timer@ff110000
- timer@ff120000
- timer@ff130000
- timer@ff140000
- serial@ff000000- used for serial communication ***uart0_dev**
- serial@ff010000 ***uart1_dev**

    - usb0@ff9d0000 ***usb0_dev**
        - usb@fe200000
    - usb1@ff9e0000 (usb1)
        - usb@fe300000
    - watchdog@fd4d0000- monitors operation of a system and triggers a reset if it fails to respond in a specific amount of time ***watchdog0_dev**
    - watchdog@ff150000 (lpd_watchdog)
    - ams@ffa50000- analog management system; manages analog functions like power management, etc ***ams_dev**
        - ams_ps@ffa50800
        - ams_pl@ffa50c00

    - dma-controller@fd4c0000 ***dpdma_dev**
    - dp_aud@fd4ac000 (zynqmp_dpaud_setting)- digital audio processing
    - display@fd4a0000 (zynqmp_dpsub)- manages any displays connected to the device ***dport_dev**
- Fclk1- fixed clock; doesn't require "dynamic clock scaling"
- Fclk2
- Fclk3
- Pass_ref_clk- clock for specific component on system, same for all below
- Video_clk
- Pass_alt_ref_clk
- Gt_crx_ref_clk
- Aux_ref_clock
- dp_aclk
- Gpio-keys- generally used to implement interrupts for buttons on a keyboard/keypad
    - Sw19
- leds
    - Heartbeat-led
- Chosen- gives some configuration info to the OS, including things like which device to boot from, kernel CLI params, etc.
- ina - all ina nodes seemed to be associated with i2c bus
- ref48M (ref48)- 48 MHz reference clock
- refhdmi (refhdmi)- HDMI reference signal
- Memory

Follow-up notes from Uzeir clarifying some devices on the minimum device required list:

I have a few questions about finalizing the devices to be removed from the device. From the minimum devices list provided to us, there were a few devices that seemingly aren't in the device tree:

apm_fpd_dev: It seems like, based on this https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/18842046/APM , that there should be something like apm@<address corresponding to DDR or CCI for the FPD> in the device tree, but there isn't. There is an entry for this in the file device-tree.mss in the Petalinux project, but that's it. Does the presence of perf monitor and the absence of the APM imply that the board design we're working on is missing the APM, so it replaces it with perf monitor for more generic performance monitoring, or does the entry in the .mss imply its present?

UK: APB is an acronym for AXI Performance Monitor so performance monitor nodes are the PS APMs. There are 4 of those. The driver is drivers/uio/uio_xilinx_apm.c. If you look at the memory map they correspond to PS APMs you are asking about:
i.e.
APM_CCI @0xFD49_0000 is perf-monitor@0xFD49_0000
APM_DDR @0xFD0B_0000 is perf-monitor@0xFD0B_0000
APM_OCM @0xFFA0_0000 is perf-monitor@0xFFA0_0000
APM_LPD_FPD @0xFFA1_0000 is perf-monitor@0xFFA1_0000

There are also PL based APMs which we aren't using in our hardware design since we aren't using FPGA.

crf_apb_dev, crl_apb_dev: assuming these are control, reset, and frequency for advanced peripheral bus and control, reset, and low-power management for APB. These also only appear in the device-tree.mss file. I couldn't really find any information on what these are/do besides this link: https://docs.xilinx.com/r/en-US/ug1085-zynq-ultrascale-trm/CRF-APB-Registers. APB appears as clk_apb in the clock-name property for some nodes, but that is it. Not sure where/if/how this is already present in the device tree we have.

UK: I think most of these register are only used when psu_init is performed or are used by the BSP drivers. I don't see any drivers in Linux that use them. They may be appearing in *.mss because they are used by low level BSP drivers. I wouldn't worry about these. If you want to see how they are used in BSP grep for various registers or look into psu_init files. Below link will show the registers supported by CRF/CRL APB modules.
https://docs.xilinx.com/r/en-US/ug1087-zynq-ultrascale-registers/CRF_APB-Module
https://docs.xilinx.com/r/en-US/ug1087-zynq-ultrascale-registers/CRL_APB-Module

dport_dev: there was no exact match for this one, but there are dp_aud and display nodes; are these considered dport devices or is that a specific standalone device?
UK: This should be node display@0xfd4a0000
https://docs.xilinx.com/r/en-US/ug1087-zynq-ultrascale-registers/DISPLAY_PORT-Module

pcie_attrib/pcie_high: neither of these appear directly in the directory where the .dts and other .dtsi are, but do appear in some qemu related directories. We have identified the axi pcie device, so I'm not sure if these are covered by that or not.
UK:

pcie_attrib: memory region is accessed in the device driver under pcie node (pcie@fd0e0000)
pcie_high: I don't see a node for this but it should be referring to the high memory region of zynqmp devices, 0x8_0000_0000

sd0/sd1: Just confirming that sdhci0 and sdhci1 correspond to these?
UK: Yes

smmu_gpv_dev and smmu_reg_dev: There is a smmu node in the tree, like the pcie devs above, but these two specific strings given to us only appear in the device-tree.mss as well. Pretty much the same question as for the others like this.
UK:
smmu_gpv_dev: It should be smmu@fd800000
smmu_reg_dev: Not in the current device tree but it should be pointing to address location 0xFD5F_0000
https://docs.xilinx.com/r/en-US/ug1087-zynq-ultrascale-registers/SMMU_REG-Module