

Genesys ZU Reference Manual

TL;DR

The black matte board you are holding in your hand is a prototyping and evaluation board proudly designed by Diligent. At its heart is a Xilinx Zynq UltraScale+ MPSoC ARM-FPGA hybrid, coupled with upgradeable memory, network and multimedia interfaces, and a wide variety of expansion connectors making it a versatile computing platform.

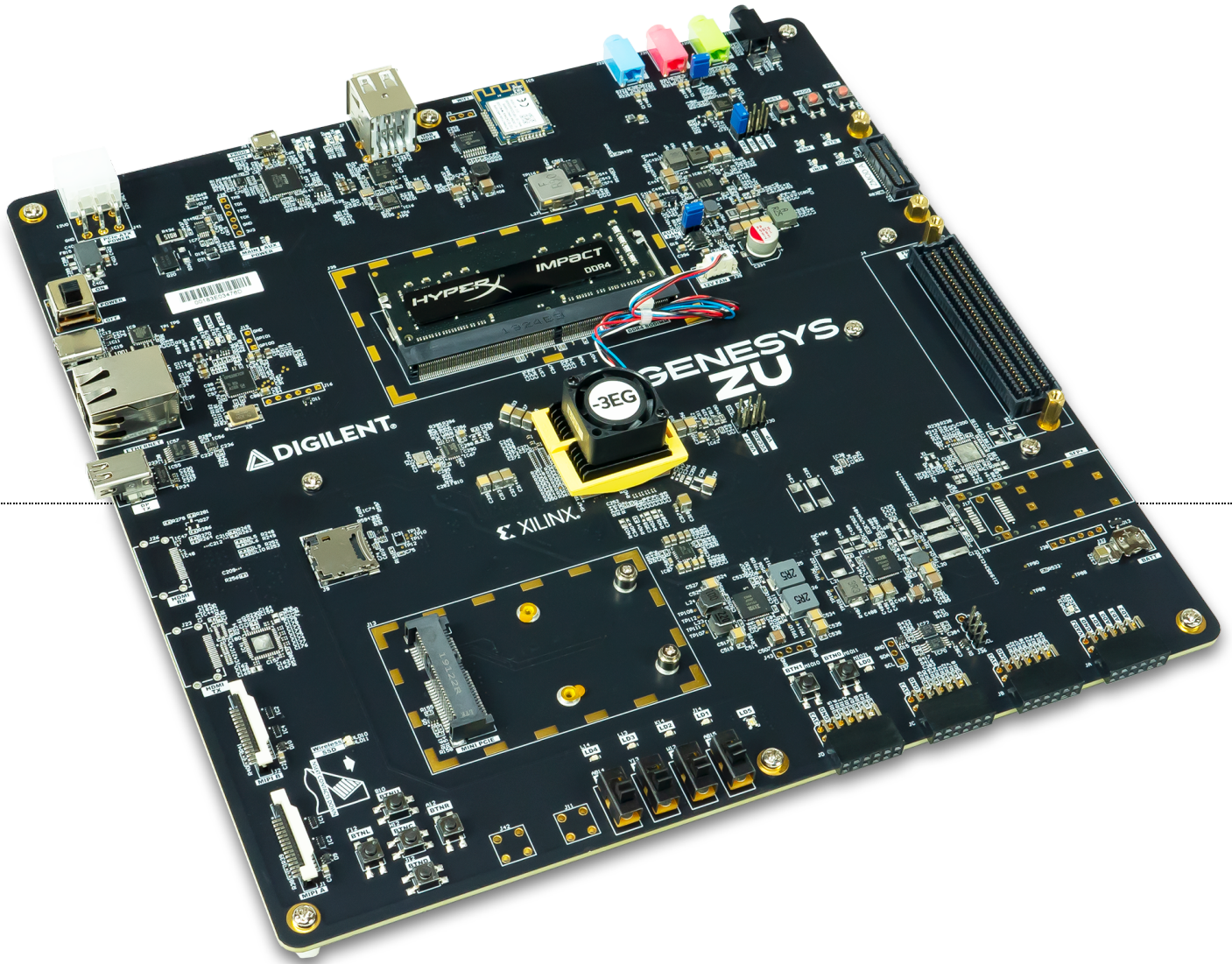
There are two variants of the Genesys ZU mentioned in this Reference Manual: 3EG and 5EV. **(The 5EV is not yet released.)** These two variants are differentiated by the MPSoC model and some peripherals. As compared to the 3EG, with the 5EV you get slightly faster DDR4, more FPGA, a video codec, and GTH transceivers allowing HDMI Source, Sink and SFP+ 10G.

Used stand-alone with a hearty bundle in the box, powered by the 12V power supply, it straight up boots Linux from the microSD card. Connect the USB micro B cable to a PC and open a terminal (115200-8-N-1) to the first COM port out of the two that appear. Login and password are both “root”. The red button labeled “POR” always resets the MPSoC and starts the boot process again.

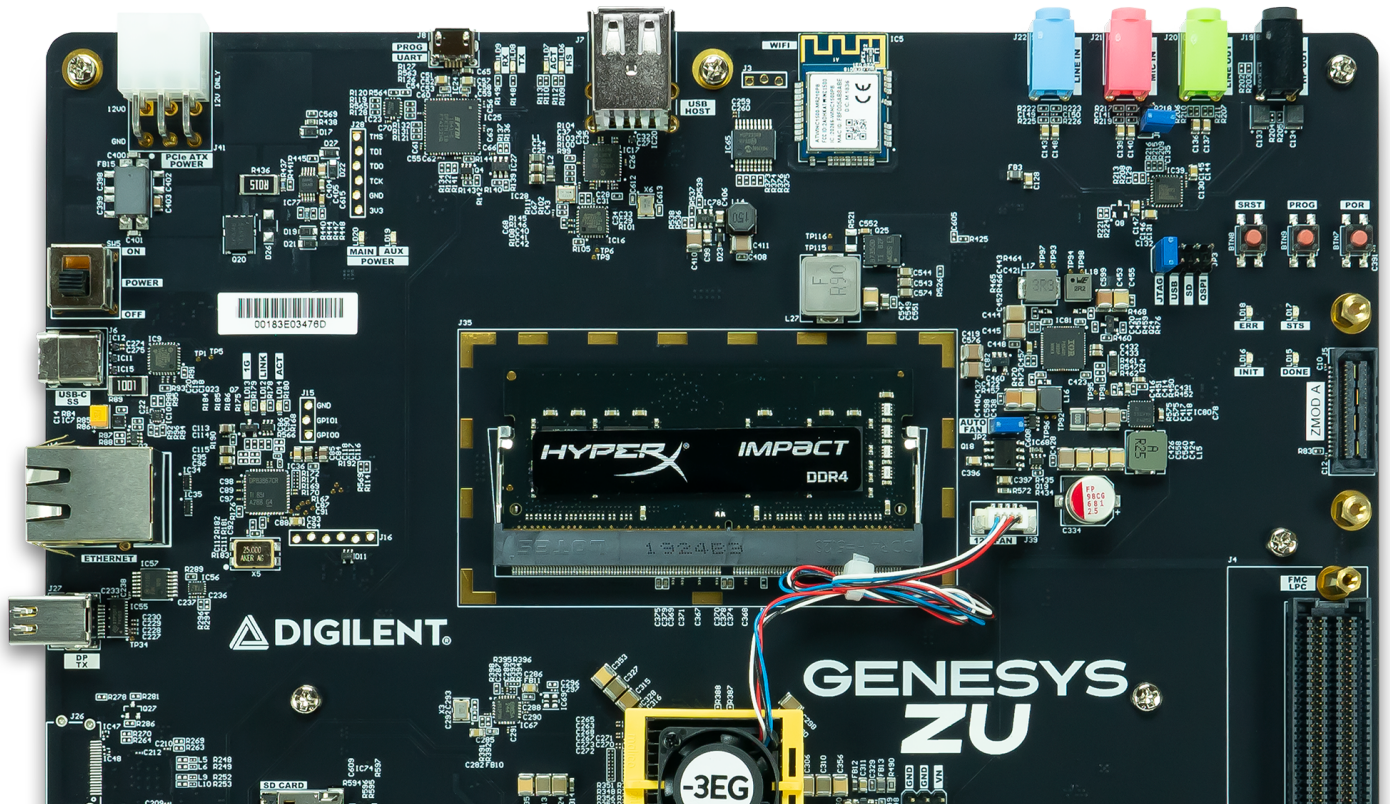
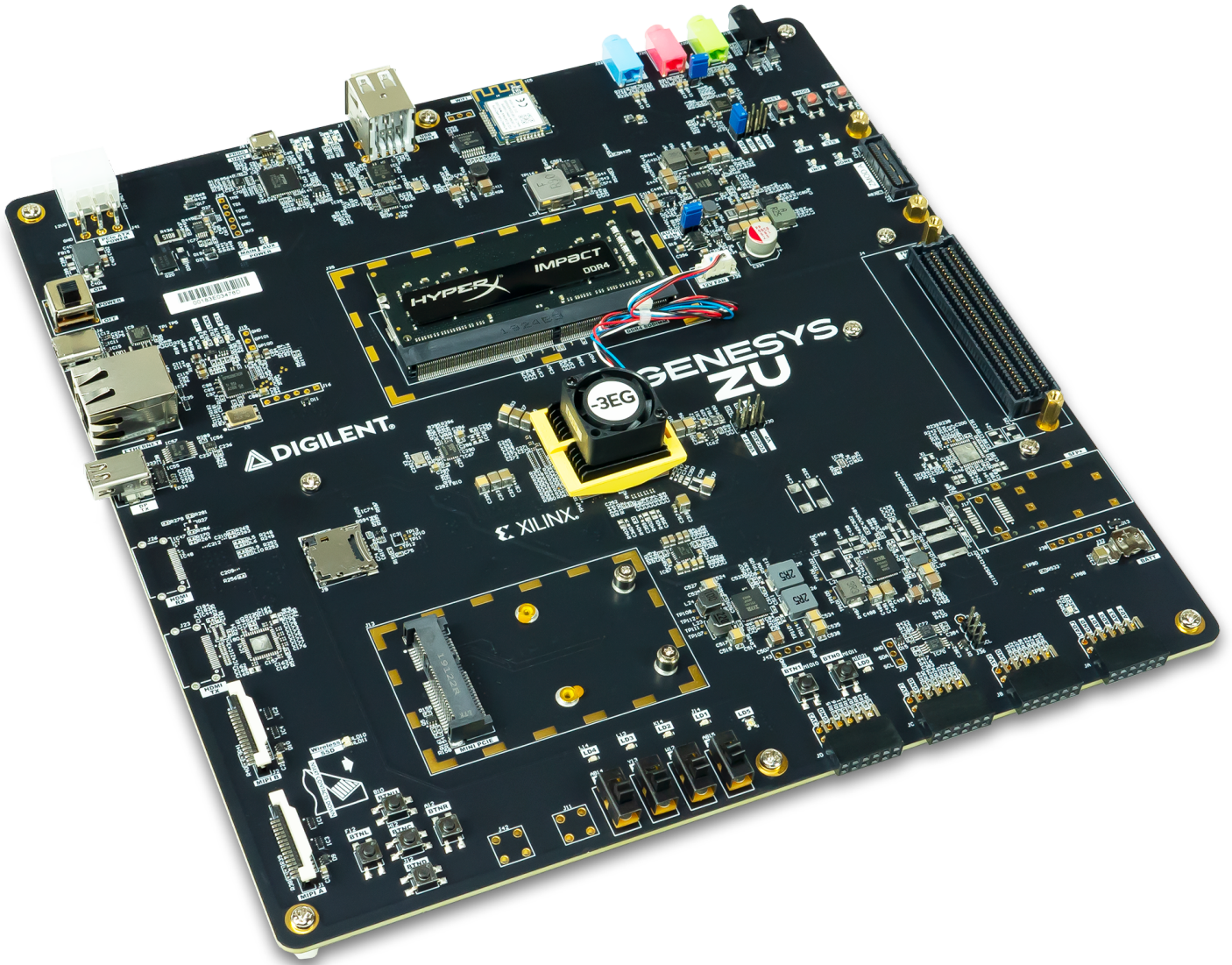
Want to dive deep into development? Head over to our [GitHub page](https://github.com/Diligent) (<https://github.com/Diligent>) and use the repos there as a starting point. Check out our [Getting Started Guide](https://reference.digilentinc.com/reference/programmable-logic/genesys-zu/getting-started) (<https://reference.digilentinc.com/reference/programmable-logic/genesys-zu/getting-started>) for a step-by-step. Build your own boot image on the SD card and boot it like the OOB demo. Not enough? Connecting the JTAG-HS1/HS2 cable to header J28 will allow for on-the-fly programming and debug using Xilinx Vivado and SDK.

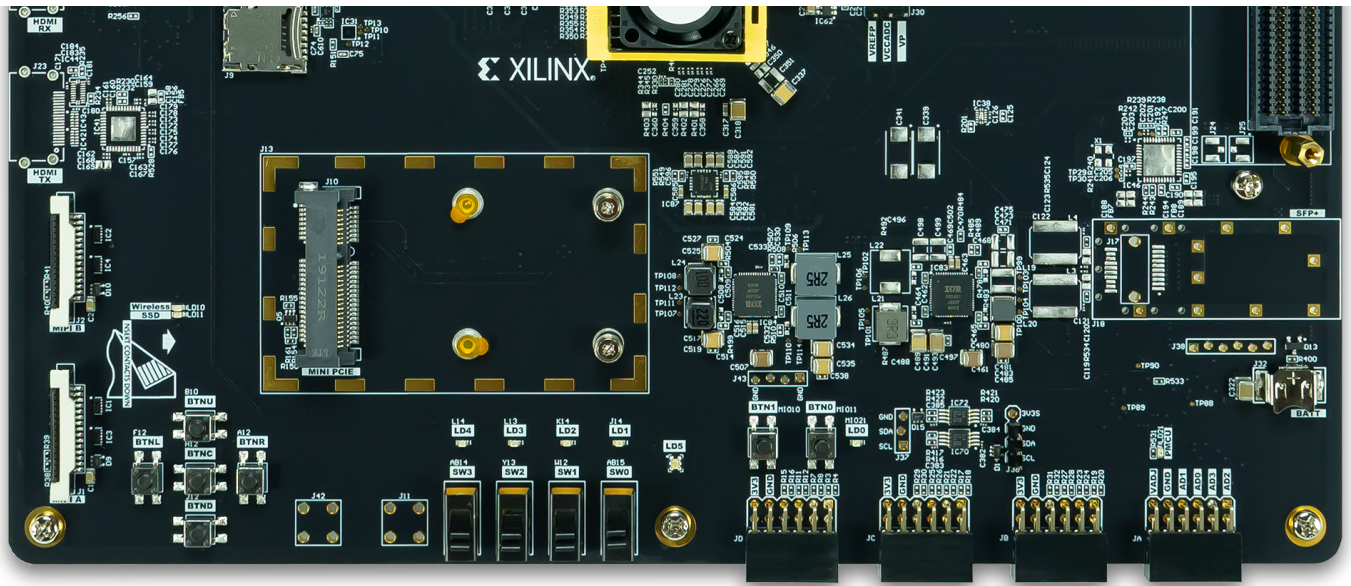
The Diligent Genesys ZU is a stand-alone Zynq UltraScale+ MPSoC prototyping and development board. It is an advanced computing platform with powerful multimedia and network connectivity interfaces. The excellent mix of on-board peripherals, upgrade-friendly DDR4, Mini PCIe and microSD slots, multi-camera and high-speed expansion connectors are bound to support a wide number of use-cases. Furthermore, the Genesys ZU is available in two variants with different MPSoC options and additional features for even more flexibility. Differences are [highlighted*](#) throughout this document.

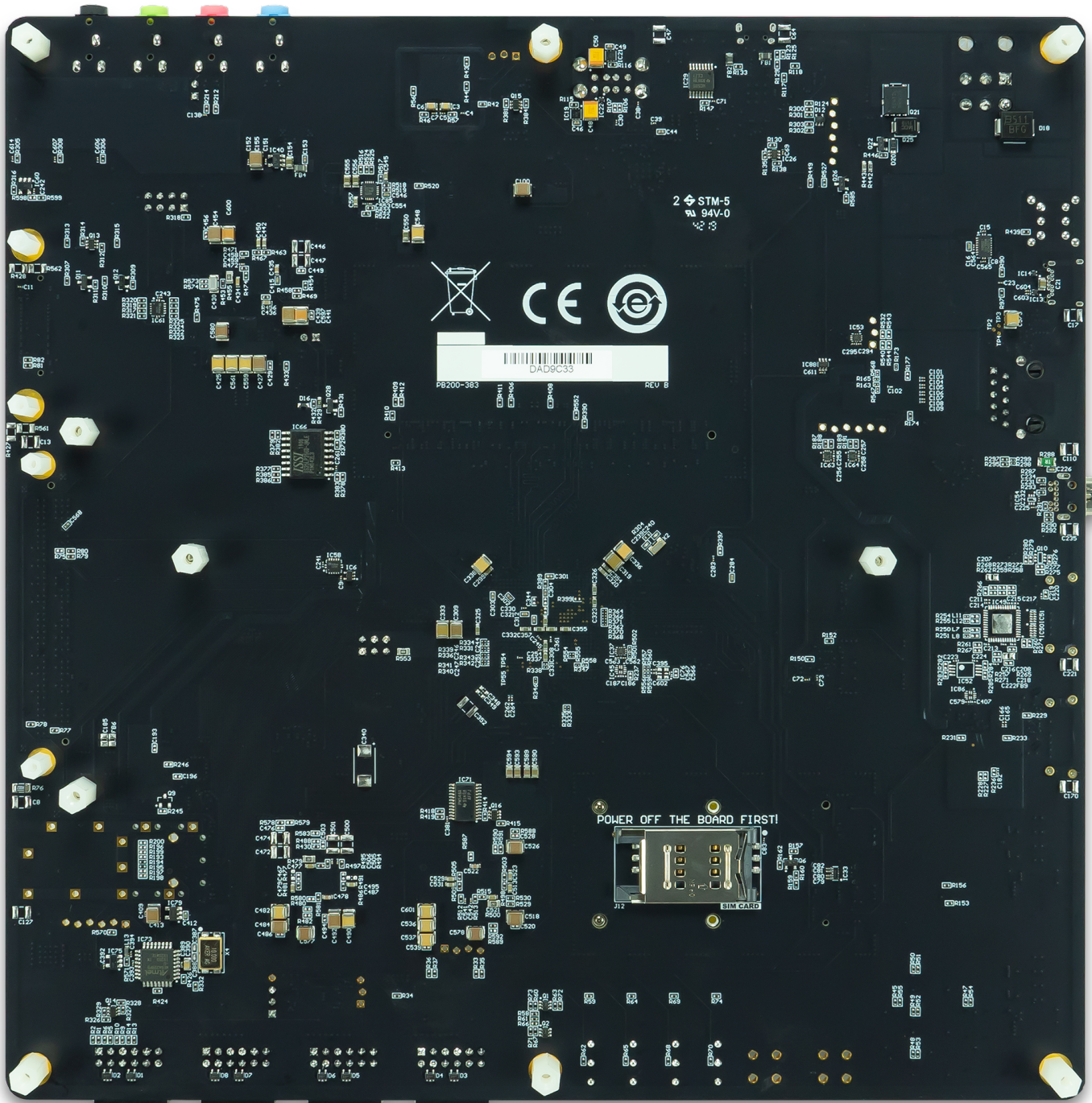
The Xilinx Zynq UltraScale+ MPSoC at the heart of the Genesys ZU is a big leap from the Zynq-7000 series. Faster and more processor cores, upgraded memory interface, integrated gigabit transceivers bring support for DDR4, USB Type-C 3.1, PCIe, SATA, DisplayPort, [SFP+*](#) and [HDMI*](#). The Genesys ZU is primarily targeted towards Linux-based applications that allows easy access to Wi-Fi, cellular radio (WWAN), SSD, USB SuperSpeed and 4K video. The bundled microSD card includes an out-of-box demo that boots a Linux image built in Petalinux and includes some test scripts for some of the peripherals.



https://reference.digilentinc.com/_media/reference/programmable-logic/genesys-zu/genesyszu-obl-2000.png







Features

Feature group	Sub-feature	Genesys ZU -3EG	Genesys ZU -5EV	Zedboard
Processor	APU	Quad A53		Dual A9
	RPU	Dual R5		X
	Main Memory	DDR4, 4GB, 1866 MT/s, upgradeable	DDR4, 4GB, 2133 MT/s, upgradeable	DDR3, 512MB, 1066 MT/s, soldered
	GPU		✓	X

Feature group	Sub-feature	Genesys ZU -3EG	Genesys ZU -5EV	Zedboard
	Video Codec	X	✓	X
Programmable logic	# of logic cells	154K	256K	85K
Peripheral connectivity	USB Type-C 3.1 Gen1 Dual-Role Device	✓		X
	MiniPCIe / mSATA dual slot	Half-/Full-size		X
	USB 2.0 Host	2 x Type-A		1 x OTG
Network connectivity	On-board Wi-Fi	2.4GHz		X
	Ethernet 1G w/ IEEE 1588	✓		X (w/o IEEE 1588)
	WLAN / WWAN / LoRa	option - MiniPCIe		X
	SFP+ 10G Ethernet	X	✓	X
Storage	SD	104 <u>MB./s</u>		25 <u>MB./s</u>
	SSD	option - mSATA		X
	Flash	ISSI 256 Mib SNOR		128 Mib
Multimedia	DisplayPort	1.2a Dual-Lane		X
	Pcam	2 x Dual-Lane		X
	HDMI Source	X	✓	✓
	HDMI Sink	X	✓	X
	Audio Codec	✓		✓
Expansion	Low speed - Pmod	4 x		5 x
	Mid-speed - SYZYGY	✓		X
	Mid-speed - FMC	✓		✓
	High-speed - FMC Gigabit	X	✓	X
User I/O	<u>LED()</u> , Buttons, Switches	✓		✓

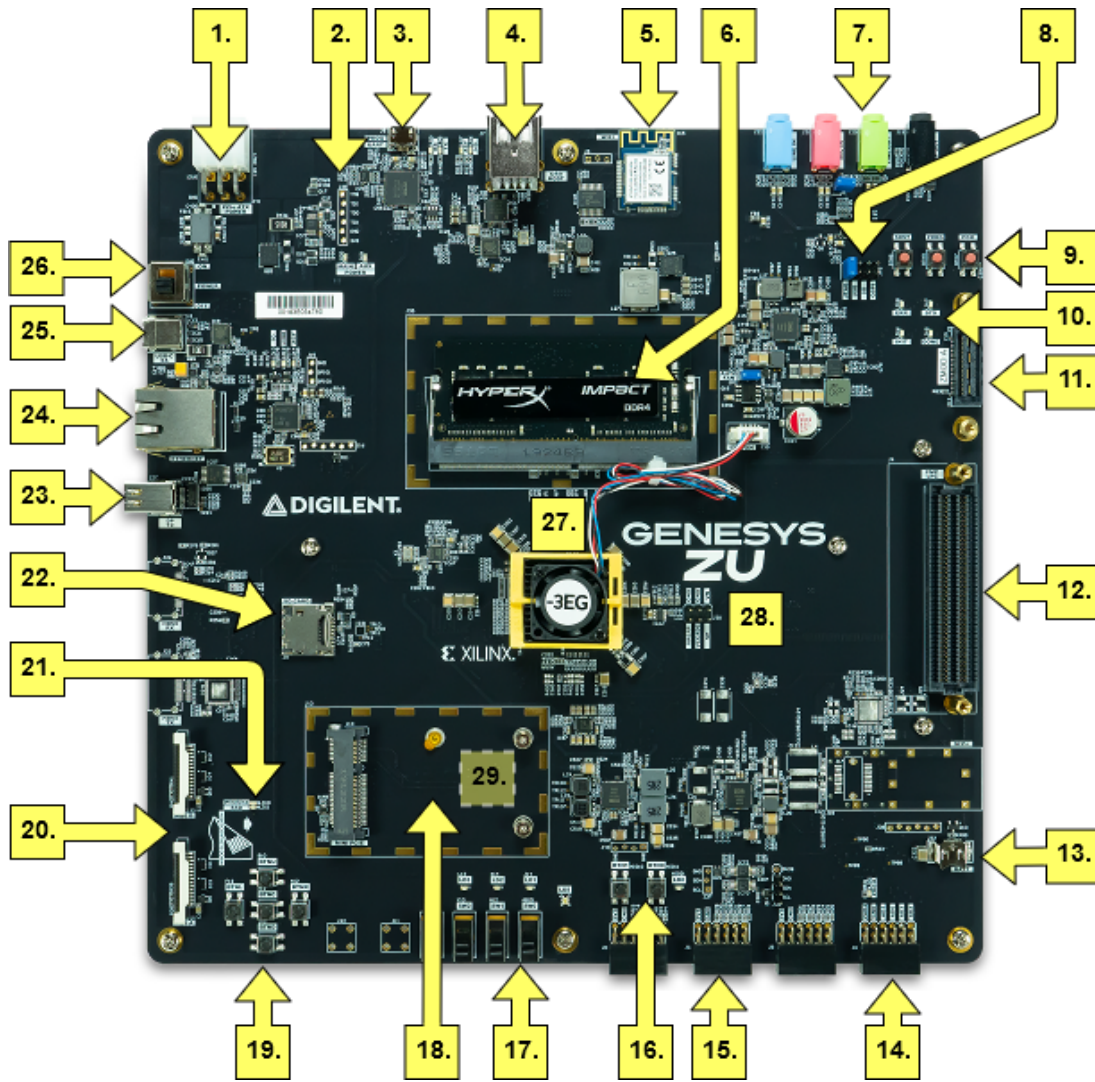


Figure I: Genesys ZU-3EG callout diagram


Callout with Description

Callout #	Description	Callout #	Description	Callout #	Description
1	6-pin PCIe power connector	13	Coin battery retainer	25	Type-C USB 3.1
2	External JTAG port	14	Pmod headers	26	Power switch
3	USB JTAG/UART port	15	Dual digital/analog Pmod	27	Zynq Ultrascale+, heat sink and fan
4	USB 2.0 host connectors	16	User buttons	28	System Monitor header
5	Wi-Fi chip	17	User switches	29	SIM card slot (on the bottom side)
6	4 GiB DDR4 SODIMM module	18	Mini PCIe/mSATA slot		
7	Audio jacks	19	User buttons		
8	Boot mode select jumper	20	MIPI (Pcam) connectors		

Callout #	Description	Callout #	Description	Callout #	Description
9	Reset buttons	21	Wireless and SSD activity LEDs		
10	INIT, DONE, ERR and STS Leds	22	MicroSD Card Slot		
11	Zmod (SYZYGY) connector	23	Mini DisplayPort		
12	FMC LPC connector	24	1G Ethernet port		

Software Support


Zynq UltraScale+ MPSoC platforms are well-suited to be embedded Linux targets, and the Genesys ZU is no exception. Diligent provides a Petalinux project that was used to build the out-of-box image.

The Genesys ZU is fully compatible with Xilinx's high-performance Vivado® Design Suite HL WebPACK™ Edition. This tool set melds FPGA logic design and embedded ARM software development into an easy to use, intuitive design flow. It can be used for designing systems of any complexity, from a complete operating system running multiple server applications, down to a simple bare-metal program that controls some LEDs. It is also possible to treat the MPSoC as a standalone FPGA for those not interested in using the processor in their design. The two MPSoC parts the Genesys ZU is available with, XCZU3EG and XCZU5EV, are  supported (<https://www.xilinx.com/products/design-tools/vivado/vivado-webpack.html#architecture>) under Vivado's free WebPACK™ license, which means the software is completely free to use, including the Logic Analyzer and High-level Synthesis (HLS) features. The Logic Analyzer assists with debugging logic that is running in hardware, and the HLS tool allows C code to be directly compiled into HDL.

In the kit you will also find a free voucher for the Xilinx MIPI CSI-2 IP cores. Follow the instructions on the voucher slip to activate the license. This is a temporary measure until Vivado 2020.1 hits, which should include a license for these IPs free of charge. See the next table for IP-support status for other peripherals.

Table I: IP support status

Feature/Peripheral	IP support	Version
DDR4 memory controller	PS hard-core, WebPACK built-in	2019.1
MIPI CSI-2/Pcam	PL soft-core,  MIPI CSI Controller Subsystems (https://www.xilinx.com/products/intellectual-property/ef-di-mipi-csi-rx.html), bundled voucher	2019.1
DisplayPort controller	PS hard-core, WebPACK built-in	2019.1
Ethernet 1G	PS hard-core, WebPACK built-in	2019.1
USB 2.0/3.0	PS hard-core, WebPACK built-in	2019.1
PCIe Root/Mini PCIe	PS hard-core, WebPACK built-in	2019.1
SATA/mSATA	PS hard-core, WebPACK built-in	2019.1
On-board Wi-Fi/SPI controller	PS hard-core, WebPACK built-in, open-source Linux  driver (https://github.com/linux4winc/driver)	2019.1
SFP+*	PL soft-core,  10G/25G Ethernet Subsystem (https://www.xilinx.com/products/intellectual-property/ef-di-25gemac.html), license required	2019.1

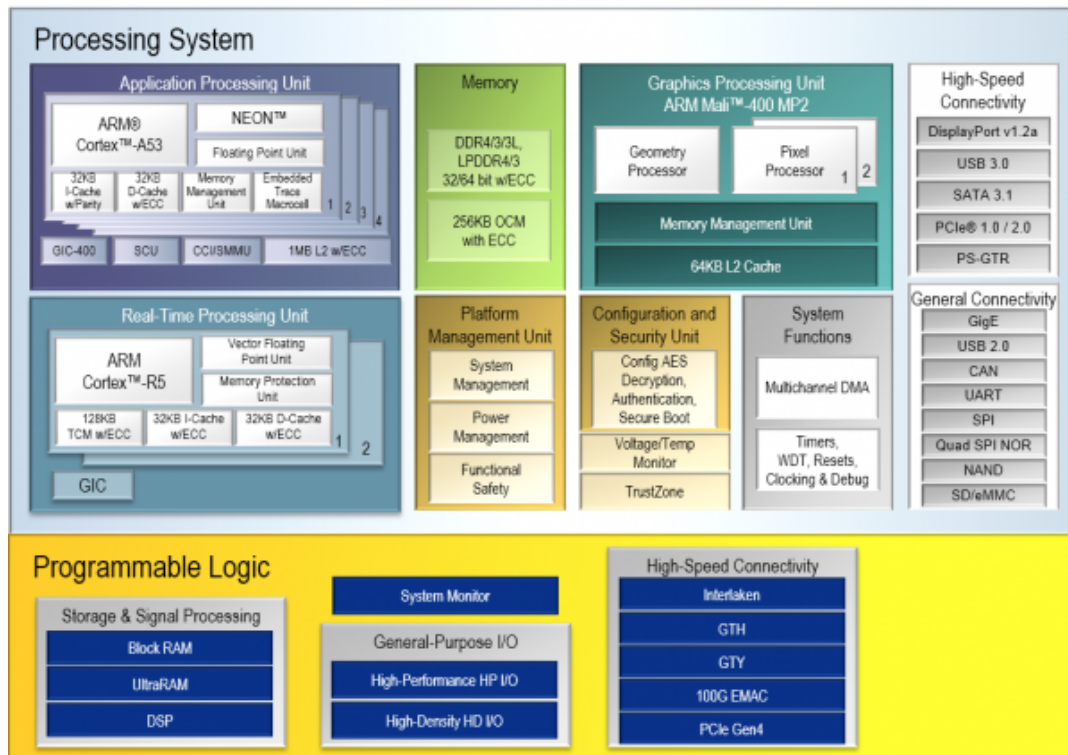
Feature/Peripheral	IP support	Version
HDMI 2.0 Source/Sink*	PL soft-core,  HDMI Subsystem (https://www.xilinx.com/products/intellectual-property/hdmi.html), license required	2019.1
Video PHY Controller*	PL soft-core, WebPACK built-in, requires protocol-implementation like the HDMI Subsystem above, supported by 5EV only	2019.1

The initial Vivado version supported by Diligent for Genesys ZU-related projects is 2019.1. Diligent currently does not provide hardware platforms or examples for Xilinx's Vitis Unified Software Platform, however Vitis support is planned for the near future.

Design resources, example projects, and tutorials are available for download at the [Genesys ZU Resource Center](https://reference.digilentinc.com/reference/programmable-logic/genesys-zu/start) (<https://reference.digilentinc.com/reference/programmable-logic/genesys-zu/start>).

Zynq UltraScale+ MPSoC Architecture

Zynq UltraScale+ MPSoC is the Xilinx second-generation Zynq platform, combining a powerful processing system (PS) and user-programmable logic (PL) into the same device. The Zynq UltraScale+ Processing System core acts as a logic connection between the PS and the Programmable Logic (PL) while assisting you to integrate customized and integrated IP cores with the processing system using the Vivado IP integrator. As you may see in the picture below, the processing system features the Arm flagship Cortex -A53 64-bit quad-core running up to 1.5GHz and Cortex-R5 dual-core real-time processor along with other interfaces such as: DDR Memory Controller, High-Connectivity, General Connectivity, System Functions etc. The Zynq UltraScale+ MPSoC Processing System wrapper instantiates the processing system section of the Zynq UltraScale+ MPSoC for the programmable logic and external board logic. The wrapper includes unaltered connectivity with the ones presented above.



(https://reference.digilentinc.com/_detail/reference/programmable-logic/genesys-zu/3eg_ps_pl.png?id=reference%3Aprogrammable-logic%3Agenesys-zu%3Areference-manual)

Figure II: Zynq UltraScale+ EG

The PS and PL can be coupled with multiple interfaces and other signals to effectively integrate user-created hardware accelerators and other functions in the PL logic that are accessible to the processors. The interfaces between the processing system and programmable logic mainly consist of three main groups: the extended multiplexed I/O (EMIO), programmable logic I/O, and the AXI I/O groups. Besides those, there are up to 78 Multiplexed I/O (MIO) ports available from the processing system. The 78 MIO signals are divided into three banks, and each bank includes 26 device pins. Each bank (500, 501, and 502) has its own power pins for the hardware interface.

MIO 0-25 : Bank 500

MIO 500 3.3 V	Peripherals							
Pin	QSPI	Mini PCIe / SATA	DDR4 SODIMM	MIO Buttons	WI-FI	UART	MIO <u>LED</u> 0	I
0	QSPI_SCLK0OUT							
1	QSPI_D1							
2	QSPI_D2							
3	QSPI_D3							
4	QSPI_D0							
5	QSPI_SS_OUTN							
6(N/C)								
7		PCIE_PERSTN						
8			DDR_SCL					
9			DDR_SDA					
10				BTN1				
11				BTN0				
12					WIFI_PORTEX_SCK			
13								
14								
15					WIFI_SSN			
16					WIFI_PORTEX_MISO 0			
17					WIFI_PORTEX_MOSI 0			
18						UART_TXD_IN		
19						UART_RXD_OUT		
20		PCIE_WAKEN						
21							LD0	
22								I
23								I

24							
25							

Note: The WI-FI signals are shared by the same bus that goes through the I/O expander.

MIO 26-51 : Bank 501

MIO 501 1.8V	Peripherals	
Pin	Ethernet	SD
26	ETH_TX_CLK	
27	ETH_TX_D0	
28	ETH_TX_D1	
29	ETH_TX_D2	
30	ETH_TX_D3	
31	ETH_TX_CTL	
32	ETH_RX_CLK	
33	ETH_RX_D0	
34	ETH_RX_D1	
35	ETH_RX_D2	
36	ETH_RX_D3	
37	ETH_RX_CTL	
38	ETH_INTN_PWDNN	
39		SDIO_SEL
40		SDIO_DIR_CMD
41		SDIO_DIR_DAT0
42		SDIO_DIR_DAT1_3
43		SDIO_POW_EN
44	ETH_RSTN	
45		SDIO_CDN
46		SDIO_R_DAT0
47		SDIO_R_DAT1

48		SDIO_R_DAT2
49		SDIO_R_DAT3
50		SDIO_R_CMD
51		SDIO_R_SCLK_0

MIO 52-77 : Bank 502

MIO 502 1.8V	Peripherals	
Pin	USB 2.0	Ethernet
52	USB20_CLK	
53	USB20_DIR	
54	USB20_DATA2	
55	USB20_NXT	
56	USB20_DATA0	
57	USB20_DATA1	
58	USB20_STP	
59	USB20_DATA3	
60	USB20_DATA4	
61	USB20_DATA5	
62	USB20_DATA6	
63	USB20_DATA7	
64	USB20H_CLK	
65	USB20H_DIR	
66	USB20H_DATA2	
67	USB20H_NXT	
68	USB20H_DATA0	
69	USB20H_DATA1	
70	USB20H_STP	
71	USB20H_DATA3	
72	USB20H_DATA4	

73	USB20H_DATA5	
74	USB20H_DATA6	
75	USB20H_DATA7	
76		ETH_MDC
77		ETH_MDIO

Functional Description

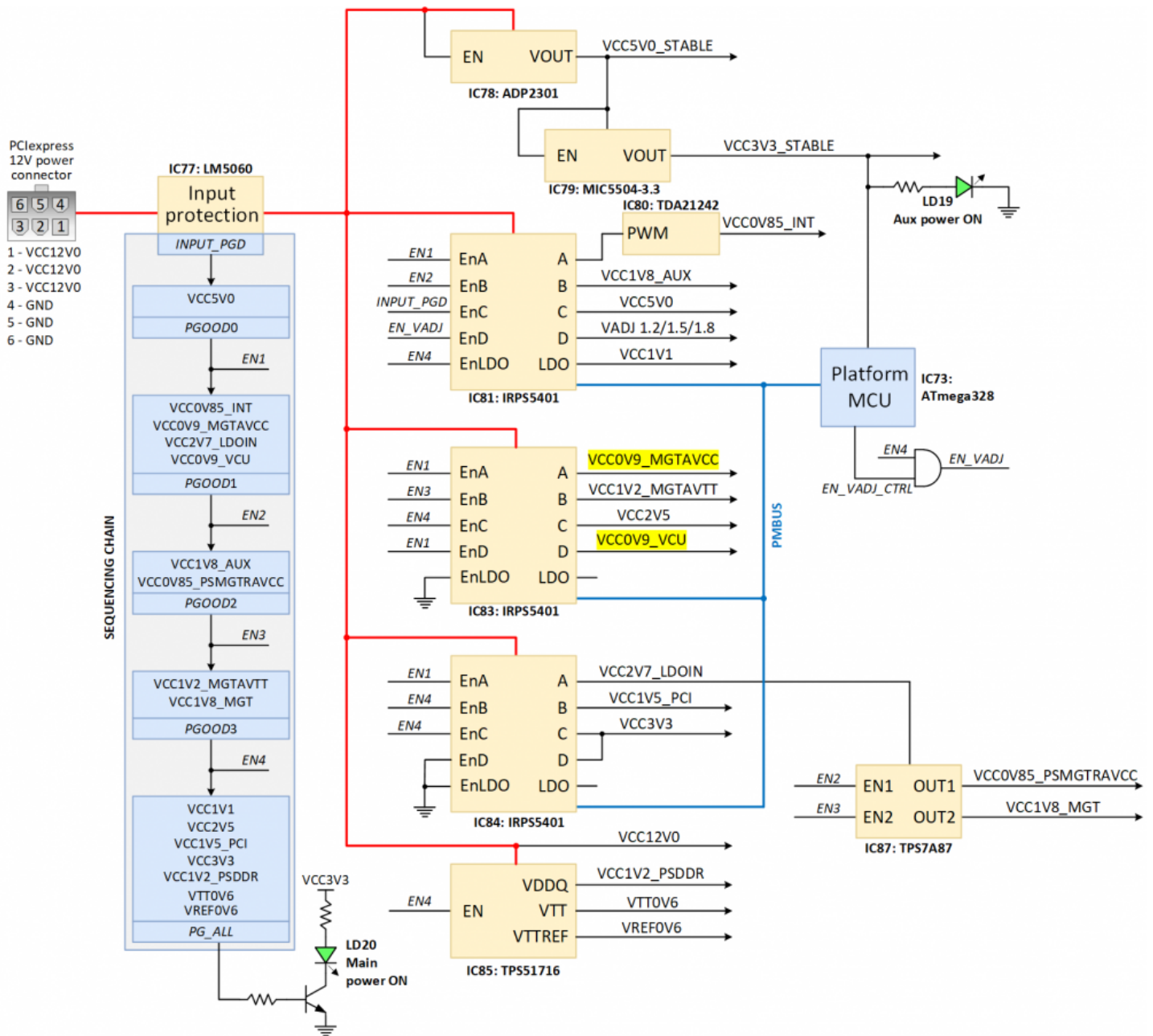
1. Power Supplies

1.1. Power Input

The Genesys ZU power distribution network was designed to meet the specific requirements of Xilinx Zynq UltraScale+ MPSoCs and of the supported peripheral devices. Power to the board is provided via a 2×3 PCIe ATX power connector. Xilinx evaluation boards use a pinout that is not compatible with ATX, therefore mixing power supplies is not possible. The bundled supply is 12V 60W-100W, depending on variant. The board power supplies are turned on or off with the SW5 slide switch.

1.2. Power Specifications

Figure 1.2.1 gives an overview of the Genesys ZU power distribution network.



(https://reference.digilentinc.com/_detail/reference/programmable-logic/genesys-zu/pdn_bloc.png?id=reference%3Aprogrammable-logic%3Agenesys-zu%3Areference-manual)

Figure 1.2.1: Genesys ZU power distribution network and supply sequencing

Note: The `VCC0V9_MGTAVCC*` and `VCC0V9_VCU*` rails are not used on the Genesys ZU-3EG variant.

When the board is turned on, the 12V input voltage provided by the bundled PCIe ATX supply is filtered and fed to the series MOSFETs of the protection circuit. The LM5060 IC monitors the input voltage and the current drawn by the entire board. If an input overvoltage/undervoltage or an overcurrent condition is detected, the LM5060 isolates the VCC12V0 net from the external supply by turning off the series MOSFETs.

During normal operation the MOSFETs provide a low impedance path and the VCC12V0 net is correctly biased with 12V. This voltage serves both as output rail for the FMC port and power input for downstream voltages.

Table 1.2.1 describes the list of power supply rails implemented on Genesys ZU. This table can be used to estimate the available power budget for a given project.

Table 1.2.1: Genesys ZU power rails

Voltage Rail	Generated from	Min/Typ/Max Voltage	Max current	Used for
VCC12V0	12V power input	12V+-5%	8.5A	FMC, power input for other rails
VCC5V0_STABLE	VCC12V0	5V+-5%	0.306A	IC81+IC80 , VCC3V3_STABLE
VCC5V0	VCC12V0	5V+-5%	4A	SYZYG, RGB LED(), Audio, USB-s, HDMI Rx AUX
VCC3V3_STABLE	VCC5V0_STABLE	3.16V / 3.3V / 3.465V	0.224A	Platform MCU, Supply I2C pull-ups, FMC, SFP, HDMI Clock, PCIe REFCLK
VCC3V3	VCC12V0	3.154V / 3.3V / 3.4V	7.64A	DDR4 VDDSPD (I2C interface), PCAMs, LEDs, SFP, HDMI, FMC, FPGA Internals, SYZYG, PMODs, PCIe, SD Card, WI-FI, USB 3.0, USB 2.0 Hub, USB Config, Display Port, Flash
VADJ	VCC12V0	Selectable - 1.2V / 1.5V / 1.8V +- 5%	2.4A	FMC, SYZYG, FPGA
VCC2V5	VCC12V0	2.375V / 2.5V / 2.625V	2.1A	DDR4 VPP, MGT Oscillators, Ethernet
VCC1V8_AUX	VCC12V0	1.746V / 1.8V / 1.854V	1.59A	FPGA Internals, Ethernet, DisplayPort, Audio, USB 3.0, USB 2.0, USB Prog, SD Card
VCC1V8_MGT	VCC2V7_LDOIN	1.746V / 1.8V / 1.854V	0.09A	FPGA Internals
VCC1V5_PCI	VCC12V0	1.5V=-5%	0.5A	PCIe port
VREF1V25	VCC3V3	1.25V+-5%	0.01A	XADC reference
VCC1V2_MGTAVTT	VCC12V0	1.164V / 1.2V / 1.236V	0.77A	FPGA Internals
VCC1V2_PSDDR	VCC12V0	1.14V / 1.2V / 1.26V	6.87A	FPGA Internals, DDR4
VCC1V1	VCC1V8_AUX	1V / 1.1V / 1.155V	0.49A	Ethernet, HDMI
VCC0V9_MGTAVCC*	VCC12V0	0.873V / 0.9V / 0.927V	0.55	FPGA Internals (5EV variant only)
VCC0V9_VCU*	VCC12V0	0.873V / 0.9V / 0.927V	2A	FPGA Internals (5EV variant only)
VCC0V85_PSMGTRAVCC	VCC2V7_LDOIN	0.825V / 0.85V / 0.875V	0.3A	FPGA Internals
VCC0V85_INT()	VCC12V0	0.825V / 0.85V / 0.875V	11A	FPGA Internals
VTT0V6	VCC1V2_PSDDR	0.6V (0.49xVCC1V2_PSDDR-0.02V... .. 0.51xVCC1V2_PSDDR+0.02V)	1A	DDR4 VTT

VREF0V6	VCC1V2_PSDDR	0.6V (0.49xVCC1V2_PSDDR... .. 0.51xVCC1V2_PSDDR)	0.01A	DDR4 reference voltage
---------	--------------	--	-------	------------------------

1.3. Power Sequencing

The board is powered up by sliding the SW5 switch to the ON position. The voltage supplies start-up sequence is defined by implementing a power good daisy chain that selectively enables groups of voltages that should start together. All supplies use a soft start mechanism to reduce the surge currents during turn on. The start-up sequence is suggested in Figure 1.2.1 and can be described in the following steps:

1. When the ramping VCC12V0 exceeds the turn-ON threshold of IC78, the VCC5V0_STABLE rail starts up. This triggers the VCC3V3_STABLE supply and powers the IC81 internal logic. The VCC3V3_STABLE rail powers the platform MCU. Its valid state is marked by the green **Aux power ON LED** (LD19).
2. When the protection circuit detects a valid 12V input, it asserts the *INPUT_PGD* signal that triggers the VCC5V0 startup. This voltage powers the internal logic of IC83 and IC84.
3. If VCC5V0 reaches its power-good threshold, the *PGOOD0=En1* signal is asserted. This enables the VCC0V85_INT(), VCC0V9_MGTAVCC*, VCC2V7_LDOIN and VCC0V9_VCU* voltages. The power-good signals of all these supplies are joined in a wired AND configuration and activate *PGOOD1* when all rails have reached their nominal voltages.
4. When all voltages from the first starting group have crossed their power-good thresholds, the *PGOOD1=En2* signal is asserted. This enables the VCC1V8_AUX and VCC0V85_PSMGTRAVCC voltages of the second group.
5. A similar trigger mechanism applies to the third (*PGOOD2=En3*) and the fourth (*PGOOD3=En4*) voltage groups as illustrated in Figure 1.2.1. The fourth group also includes the dedicated DDR4 power supply with all its output voltages.
6. If all voltages from the fourth group have successfully reached their designed values, their power-good (*PG_ALL*) lights up the green **Main Power ON LED** (LD20). At this point the board is fully functional.

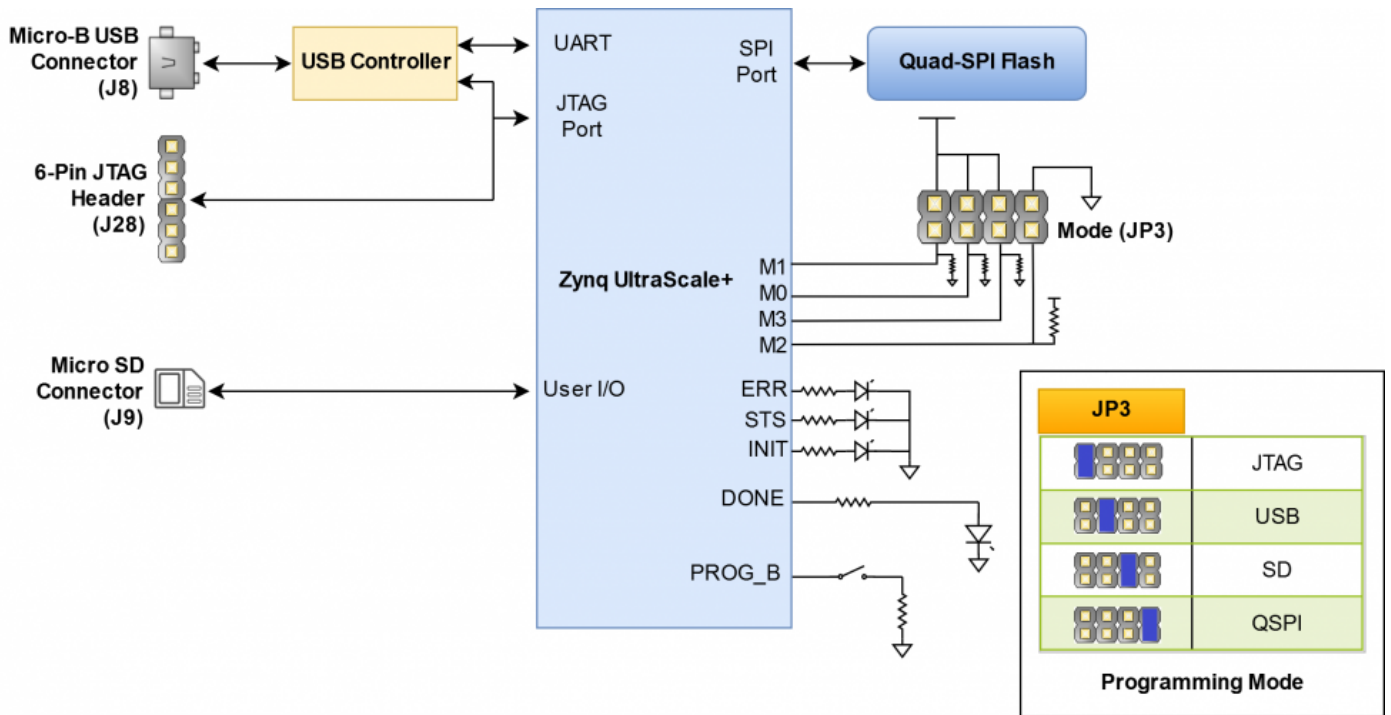
Note: the VADJ rail is controlled separately by the platform MCU that must first set VADJ depending on the peripherals using those voltage. VADJ is in the fourth start-up group but it is conditioned by a valid *EN_VADJ_CTRL* signal generated by the platform MCU.

Sliding the SW5 switch to the OFF position disables the power supplies by pulling *INPUT_PGD* to ground. The capacitor C405 connected to the *EN* terminal of the LM5060 monitoring IC delays the VCC12V0 turn off with approximately 200ms until all power supplies have been safely powered off.

1.4. Earthing/Grounding System

The bundled power supply has a 3-pin AC power cord and internally shorts the DC negative terminal to the earth/ground circuit of the power socket. Therefore, the Genesys ZU's ground circuits are at earth potential. In lieu of a metallic case the Genesys ZU has an internal shield ring at the edge of the board encircling it. The metallic pads of screws, feet and stand-offs (except the top right corner and Zmod stand-offs), but also the shell of connectors are wired to the shield ring. The shield ring ultimately connects to the Genesys ZU's ground circuit through a 2 mOhm resistor. There are also several capacitor pads (not loaded) around the board between the shield ring and the ground circuit. The screws on the edge of the board (except the top-right corner) can be used to wire an additional earth/ground connection to the Genesys ZU.

2. MPSoC Boot Process



(https://reference.digilentinc.com/_detail/reference/programmable-logic/genesys-zu/gzu-config.png?id=reference%3Aprogrammable-logic%3Agenesys-zu%3Areference-manual)

Figure 2.1: Genesys ZU Boot Diagram

2.1. JTAG Boot Mode

JTAG is the most important component of the debug features for software and PL development. The JTAG architecture has three Test Access Port (TAP) controllers:

1. PS TAP (main PS controller with IDCODE)
2. PL TAP (used for PL configuration and boundary scan)
3. DAP (used for ARM debugging, Real time processing unit (RPU) and Application Processing Unit (APU))

Taking into account this architecture, when placed in JTAG boot mode, the processor (APU) will wait until software is loaded by a host computer using the Xilinx tools. After software has been loaded, it is possible to either let the software begin executing, or step through it line by line using Xilinx SDK.

It is also possible to directly configure the PL over JTAG, independent of the processor. This can be done using the Vivado Hardware Server.

The Genesys ZU is configured to boot in Cascaded JTAG mode, which allows the PS to be accessed via the same JTAG port as the PL.

You need a JTAG programmer to connect into the JTAG chain of the Genesys ZU. There is an on-board USB-JTAG controller for which Vivado support is expected in version 2020.1. Until built-in Vivado support is available a Digilent JTAG-HS1 or JTAG-HS2 programming cable will be bundled with the kit. This cable connects to the 6-pin header J28 and is already supported by Vivado.

Connecting both the on-board programmer and the bundled programming cable to the PC might cause conflict in Vivado Hardware Server with not the right cable being opened and no targets being found. The solution is launching Hardware Server manually before any connection attempt is made, or after killing any automatically launched instances of hw_server.exe. Launching Hardware Server manually can be done from the Vivado Tcl Shell using the command below.

```
Vivado% hw_server -e "set jtag-port-filter 210205,210249"
WARNING: [Common 17-259] Unknown Tcl command 'hw_server -e set jtag-port-filter 210205,210249' sending command

***** Xilinx hw_server v2019.1
**** Build date : May 24 2019 at 15:13:31
** Copyright 1986-2019 Xilinx, Inc. All Rights Reserved.

INFO: hw_server application started
INFO: Use Ctrl-C to exit hw_server application

INFO: To connect to this hw_server instance use url: TCP:<hostname>:3121
```

Vivado Hardware Server is launched in the shell and will be listening until the shell is closed. All other Xilinx tools will automatically connect to this instance of the Hardware Server.

For more details about JTAG see “JTAG Functional Description” section in [Zynq UltraScale+ Device Technical Reference Manual \(UG1085\)](https://www.xilinx.com/support/documentation/user_guides/ug1085-zynq-ultrascale-trm.pdf) (https://www.xilinx.com/support/documentation/user_guides/ug1085-zynq-ultrascale-trm.pdf).

2.2. microSD Boot Mode

The Genesys ZU supports booting from a microSD card inserted into connector J9. The SD supported version is 3.0. This boot mode support FAT 16/32 file systems for reading the boot images. Image search for multi-boot is supported. For SD boot mode, the boot image file should be at the root of first partition of the SD card (not inside any directory). The following procedure will allow you to boot the Zynq UltraScale+ from microSD with a standard Zynq UltraScale+ Boot Image created with the Xilinx tools:

1. Format the microSD card with a FAT32 file system.
2. Copy the Zynq UltraScale+ Boot Image created with Xilinx SDK to the microSD card.
3. Rename the Zynq UltraScale+ Boot Image on the microSD card to BOOT.bin.
4. Eject the microSD card from your computer and insert it into connector J9 on the Genesys ZU.
5. Attach a power source to the Genesys ZU.
6. Place a single jumper on JP3, shorting the pins labeled “SD”.
7. Turn the board on. The board will now boot the image on the microSD card.

For more details about SD boot mode see “SD Boot Mode” section in [Zynq UltraScale+ MPSoC Software Developer Guide \(UG1137\)](https://www.xilinx.com/content/dam/xilinx/support/documentation/user_guides/ug1137-zynq-ultrascale-mpsoc-swdev.pdf) (https://www.xilinx.com/content/dam/xilinx/support/documentation/user_guides/ug1137-zynq-ultrascale-mpsoc-swdev.pdf).

2.3. Quad SPI Boot Mode

The Genesys ZU has an on-board 256Mbit Quad-SPI Flash from ISSI that the Zynq UltraScale+ can boot from. Documentation available from Xilinx describes how to use Xilinx SDK to program a Zynq UltraScale+ Boot Image into a Flash device attached to the Zynq UltraScale+. Once the Quad SPI Flash has been loaded with a Zynq UltraScale+ Boot Image, the following steps can be followed to boot from it:

1. Attach a power source to the Genesys ZU.
2. Place a single jumper on JP3, shorting the two center pins (labeled “QSPI”).
3. Turn the board on. The board will now boot the image stored in the Quad SPI flash.

For more details about Quad SPI boot mode see “QSPI24 and QSPI32 Boot Modes” section in [Zynq UltraScale+ MPSoC Software Developer Guide \(UG1137\)](https://www.xilinx.com/content/dam/xilinx/support/documentation/user_guides/ug1137-zynq-ultrascale-mpsoc-swdev.pdf) (https://www.xilinx.com/content/dam/xilinx/support/documentation/user_guides/ug1137-zynq-ultrascale-mpsoc-swdev.pdf).

2.4. USB Boot Mode

It is the only boot mode apart from JTAG where the MPSoC takes a slave role. It shows up as a DFU (Device Firmware Upgrade) USB device to the PC, waiting for a configuration. Using this boot mode you will be able to load the newly created image on Zynq UltraScale+ via the USB Port. For more details see the “Boot Sequence for USB Boot Mode” mode in [Zynq UltraScale+ MPSoC: Embedded Design Tutorial \(UG1209\)](https://www.xilinx.com/support/documentation/sw_manuals/xilinx2019_1/ug1209-embedded-design-tutorial.pdf) (https://www.xilinx.com/support/documentation/sw_manuals/xilinx2019_1/ug1209-embedded-design-tutorial.pdf).

2.5. Status LEDs

The Genesys ZU has four status LEDs:

1. ERR : it is asserted for accidental loss of power, a hardware error or an exception in the Platform Management Unit (PMU);
2. STS : it is asserted in secure lockdown state;
3. INIT : indicates the PL is initialized after the power-on reset (POR);
4. DONE : it is asserted when the PL configuration is completed;

3. Main Memory

Main memory is a single-slot populated DDR4 SODIMM, always upgradeable by the user. It is wired to the PS (Processing System) side using the hard-core memory controller. The bundled module is a 4GiB Kingston HyperX HX424S14IB/4. Although the module supports DDR4-2400 CL14-14-14 timing, data rate is limited by the MPSoC and the board. The 5EV board variant supports DDR4-2133*, while the 3EG supports DDR4-1866.

Although the bundled module is not ECC-capable, the Genesys ZU is. Just pair it with an ECC module and enable the feature in the Vivado MPSoC PS Configuration Wizard.

3.1. Implementation

There is a single SODIMM slot on the top side of the Genesys ZU just north of the MPSoC. It is wired to the PS-side memory controller and supports any SODIMM module complying with the memory controller's restrictions. These are detailed in the [Zynq UltraScale+ Device Technical Reference Manual \(UG1085\)](https://www.xilinx.com/support/documentation/user_guides/ug1085-zynq-ultrascale-trm.pdf) (https://www.xilinx.com/support/documentation/user_guides/ug1085-zynq-ultrascale-trm.pdf), but common modules of 1R/2R, x8/x16, 64b/72b are supported.

The serial presence detect (SPD) interface is wired to MIO8 (DDR_SCL) and MIO9 (DDR_SDA), accessible through the I2C1 controller.

For better routing some byte swaps were performed detailed in Table 3.1.1. No nibble or bit swaps were needed.

Table 3.1.1: DDR4 interface byte swaps.

System	7	6	5	4	3	2	1	0	8
Slot	0	1	2	3	4	5	6	7	8

ECC byte (lane 8) is equivalent to any of the data bytes from the perspective of the DRAM components. The controller and SODIMM connector have dedicated ECC pins (CBx), which are not used on non-ECC systems. Therefore the ECC lane (CBx) cannot be swapped with other lanes. However, byte and bit swaps in data lanes are transparent to the ECC feature since any swap performed upon write is reversed back upon read.

The Write CRC is a new feature of DDR4 and is complementary and unrelated to the ECC feature. Write CRC protects data in transit, ECC protects data in storage. CRC is calculated both by the controller and the DRAM to avoid data corruption in the the write data burst. It can detect single bit, double bit, odd count and one multi-bit UI vertical column errors. Upon error detection, DRAM will assert the ALERT_n line. The controller should retry the write upon error.

It should be enabled in systems that expect a high amount of signal integrity issues and where high reliability is desired. It trades data rate for reliability. CRC support is optional in SODIMM modules. Even if the module supports it, implementation is not easy. For CRC to work the controller must know what pin swaps were performed on the memory interface. In case of SODIMM modules, there are some restricted pin swaps possible and must be documented in the SPD EEPROM(0). The controller is expected to read these, combine it with pin swaps on the system board and assign the bits to CRC inputs accordingly. According to [AR# 68788](https://www.xilinx.com/support/answers/68788.html) (<https://www.xilinx.com/support/answers/68788.html>) this can be achieved through the DDRC.DQMAP registers, not well documented.

4. Storage

4.1. Quad-SPI Flash

The Genesys ZU features a serial flash memory from ISSI. This memory is used to provide non-volatile code and data storage. It can be used to initialize the PS subsystem as well as configure the PL. The key device attributes are:

- Part number: IS25LP256D-JMLE
- Size: 256Mbit / 32Mbyte
- 1-bit, 2-bit and 4-bit bus widths supported
- 80MHz Normal Read, Up to 166MHz Fast Read
- Up to 664Mb/s in quad-spi mode
- Powered from 3.3V

The Flash is also commonly used to store non-configuration data needed by the application. If doing this from a bare-metal application, the flash memory can be freely accessed using standalone libraries included with a Xilinx SDK BSP project. If doing this from a Petalinux generated embedded Linux system, the Flash can be partitioned as desired and mounted/accessed like a standard MTD block device. See the Petalinux and Xilinx SDK documentation for more information.

The Flash connects to the Quad-SPI Flash controller of the Zynq UltraScale+ via pins in MIO Bank 0/500 (specifically MIO[0:5]).

The memory is divided into uniform 4 KByte sectors or uniform 32/64 Kbyte blocks. A block consists of 8/16 adjacent sectors.

Two globally unique MAC address are programmed in the last sector, sector 8191.

- MAC for Ethernet PHY is stored at address 0x1FFF000
- MAC for SFP+ (5EV only) is stored at address 0x1FFF006

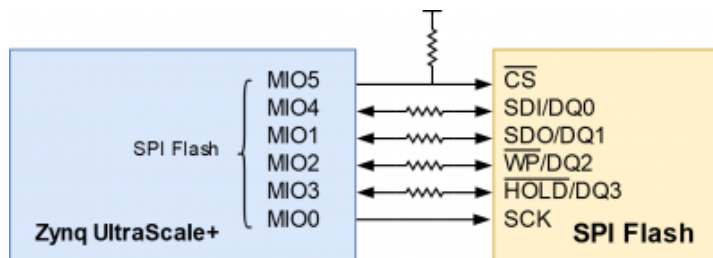
The last sector, used to store the MAC addresses, is protected from write and erase. Any attempt to program or erase the last sector will fail. Consequently, blank check operations (like the one in Vivado Hardware Manager) are also expected to fail, even after a full device erase, if the last sector is not ignored.

The ISSI flash features an Advanced Sector/Block Protection mechanism. Every main flash memory array block/top sector/bottom sector has a non-volatile (PPB) protection bit associated with it. When the bit is 0, the sector is protected from program and erase operations.

There is a TBPARM bit that defines the logical location of the parameter block. The parameter block consists of thirty two 4KB sectors, which replace two 64KB blocks.

When TBPARM is set to 0 the parameter block is in the top of the memory array address space. When TBPARM is set to 1 the parameter block is at the bottom of the array. TBPARM is OTP(One-Time Programmable) and set to 1 when shipped from factory. If TBPARM is programmed to 0, an attempt to change it back to 1 will fail and ignore the program operation. For more details about the Advanced Sector/Block Protection mechanism consult the manufacturer's [datasheet \(http://www.issi.com/WW/pdf/IS25LP\(WP\)256D.pdf\)](http://www.issi.com/WW/pdf/IS25LP(WP)256D.pdf).

To protect the MAC addresses from the last sector, Genesys ZU comes with TBPARM programmed to 0 and the PPB bits for the last sector programmed to 0.



https://reference.digilentinc.com/_detail/reference/programmable-logic/genesys-zu/gzu-flash.png?id=reference%3Aprogrammable-logic%3Agenesys-zu%3Areference-manual

Figure 4.1.1: Genesys ZU Quad SPI Flash

4.2. microSD slot

The microSD connector J9 located on the top side has a hinge-based mechanism. It is compatible with UHS-I allowing 1.8V signalling and speeds up to SDR104, or 104MB/s. To enable UHS-I support and speeds up to SDR104, see the following Answer Records from Xilinx:

<https://www.xilinx.com/support/answers/69978.html> (<https://www.xilinx.com/support/answers/69978.html>) and <https://www.xilinx.com/support/answers/70062.html> (<https://www.xilinx.com/support/answers/70062.html>).

4.3. mSATA slot

The Mini PCIe connector J13 doubles as an mSATA slot allowing fast non-volatile SSD storage. Both half and full-size modules are supported. Since PCIe and SATA share the same GTR lanes, one has to disable PCIe first to enable SATA.

5. Oscillators/Clocks

The PS on Genesys ZU has a clock source of 30MHz on pin R16 (PS_REF_CLK), coming from IC67 (5P49V6965A244NLGI).

The PL on Genesys ZU has a clock source of (by default) 25MHz on pin E12 (SYSCLOCK), coming from the CLK_OUT pin of the Ethernet PHY IC36 (DP83867CRRGZR). This LVCMOS18 I/O standard clock can be used for any purpose in the PL, with certain restrictions:

- It enters the FPGA via HDGC (High-Density I/O Bank Global Clock) pins. These can only directly drive BUFGCE primitives, not MMCM/PLL primitives. See Global Clock Inputs in [ug572 \(https://www.xilinx.com/support/documentation/user_guides/ug572-ultrascale-clocking.pdf\)](https://www.xilinx.com/support/documentation/user_guides/ug572-ultrascale-clocking.pdf). It can still drive MMCM/PLL indirectly, but this will need a `CLOCK_DEDICATED_ROUTE FALSE` constraint, otherwise Vivado DRC will fail.
- Is available whenever the PHY is out of reset and is not specifically configured through its registers to disable this clock. By default it is synchronous to the crystal input of the Ethernet PHY (25 MHz_{crystal}).

Add info on GTR and GTH reference clocks

6. Reset Sources

The Genesys ZU provides several different methods of resetting the Zynq Ultrascale+ device, as described in the following sections.

6.1 Power-on Reset

The Zynq Ultrascale+ PS supports external power-on reset signals. The power-on reset is the master reset of the entire chip. This signal resets every register in the device capable of being reset. The Genesys ZU drives this signal from the PG_ALL signal of the power supplies in order to hold the system in reset until all power supplies are valid. The Genesys ZU also has a red push button, labeled POR, which can toggle the power-on reset of the Zynq Ultrascale+.

6.2 Programmable Logic Reset

A red push button, labeled PROG, toggles the Zynq Ultrascale+'s PS_PROG_B input. This resets the PL and causes DONE to be de-asserted. The PL will remain unconfigured until it is reprogrammed by the processor or via JTAG.

6.3 Processor Subsystem Reset

The external system reset button, labeled SRST, resets the Zynq Ultrascale+ device without disturbing the debug environment. For example, the previous break points set by the user remain valid after system reset. Due to security concerns, system reset erases all memory content within the PS, including the On-Chip-Memory (OCM). The PL is also cleared during a system reset. System reset does not cause the boot mode strapping pins to be re-sampled. After changing boot mode jumpers a power-on reset is needed to act on the new setting.

For more details about configuration pins see “Clock, Reset, and Configuration Pins” section in [Zynq UltraScale+ Device Technical Reference Manual \(UG1085\) \(https://www.xilinx.com/support/documentation/user_guides/ug1085-zynq-ultrascale-trm.pdf\)](https://www.xilinx.com/support/documentation/user_guides/ug1085-zynq-ultrascale-trm.pdf).

7. Network Connectivity

7.1 Wi-Fi

A Microchip ATWINC1500 module provides 2.4GHz IEEE 802.11 b/g/n wireless network connectivity. It interfaces to the MPSoC on the PS-side over SPI, supporting a maximum theoretical data rate of 48Mbps. The ATWINC1500 can be used in bare-metal applications with the full IP stack included in the firmware loaded from flash. However, it is also supported in Linux in the ATWILC1000-compatible mode, where the firmware is loaded on-the-fly upon boot and the `QoS()` IP stack is used.

7.2 1G Ethernet

The Genesys ZU uses a TI DP83867CR PHY to implement a 10/100/1000 Ethernet port for wired connectivity. The PHY connects to MIO Bank 501 (1.8V) and interfaces to the MPSoC via RGMII for data and MDIO for management. The auxiliary interrupt (ETH_INTN_PWDNN) and reset (ETH_RSTN) signals also connect to MIO Bank 501.


After power-up the PHY starts with Auto Negotiation enabled, advertising 10/100/1000 link speeds and full duplex. If there is an Ethernet-capable partner connected, the PHY automatically establishes a link with it, even with the MPSoC not configured.

Three status indicator LEDs are on-board near the RJ-45 connector that indicate speed (LD13), valid link state (LD12), and traffic activity (LD14).

Although the default power-up configuration of the PHY might be enough in most applications, the MDIO bus is available for management. The PHY is assigned the 5-bit address 01111 on the MDIO bus. With simple register read and write commands, status information can be read out or configuration changed. The TI PHY follows industry-standard register map for basic configuration.

The RGMII specification calls for the receive (RXC) and transmit clock (TXC) to be delayed relative to the data signals (RXD[0:3], RXCTL and TXD[0:3], TXCTL). Xilinx PCB guidelines also require this delay to be added. The PHY is configured to insert a delay of 2.0ns between RXD/CTL and RXC, and a delay of 1.5ns between TXD/CTL and TXC.

On an Ethernet network each node needs a unique MAC address. To this end, the last sector of the Quad-SPI flash has been programmed at the factory with a 48-bit globally unique EUI-48/64™ compatible identifier. For more details about MAC address storage see [Quad-SPI Flash](#).

U-Boot has been  patched (<https://github.com/Diligent/Genesys-ZU-OOB-os/commit/7eac6c368efd33042e95eb78abac5c16d87020c4>) to read this MAC address and overwrite the existing node in the device tree binary before handing control over to the Linux kernel.

The identifier is also printed on a sticker found next to the Ethernet jack (J14).

7.3. 10G SFP+

Todo for 5EV.

7.4. WLAN, Bluetooth, WWAN

The Mini PCIe socket allows you to connect any wireless radio module compatible with the PCIe Mini Card standard. With both PCIe x1 and USB 2.0 available in the socket, even dual Wi-Fi/Bluetooth modules can be used. The primary use case is Linux [OS\(\)_\(\)](#), so modules with Linux drivers available in-kernel are recommended. For WWAN radio modules a SIM slot is present on the bottom side of the board.

8. Peripheral Connectivity

8.1. USB Full-Featured Type-C

USB 3.1 Gen1 and USB 2.0 support is handled by the Full-Featured Type-C receptacle J6. The connector has a USB 2.0 pair for backward compatibility, one high-speed transceiver lane (two pairs) for USB 3.1, and configuration channel (CC). Since the plug is reversible, the upper and lower rows double the number of pins for each function, designated by suffixes 1 and 2. Plug orientation is established during the configuration process over the CC1 and CC2 pins. Depending on the orientation, either pins with suffix 1 or pins with suffix 2 carry actual signals. Multiplexing 1 and 2 for the USB 3.1 lane is done by an on-board hardware multiplexer.

Unlike the other USB connector types, Type-C does not inherently establish the relationship of host and device ports. This relationship is determined during the same configuration process.

On the Genesys ZU, data behavior is Dual-Role-Data (DRD), ie. can behave either as a Downstream-Facing Port (DFP) or an Upstream-Facing Port (UFP), depending on the connected partner and MPSoC configuration. Power behavior is Dual-Role-Power (DRP), but even if UFP is negotiated, the board remains self-powered. In DFP role the advertised current capability and limit is 0.9 A with the possibility of increasing it to 1.5 A. USB Power Delivery is not supported.

Management of the Type-C port is handled by a companion chip, the TI TUSB322I. It handles attachment, cable orientation, role detection and current advertisement. It connects to the main I²C bus of the board and can be used to read status and set port roles for Type-C. It responds to address 1000111b on branch 3 of the I²C multiplexer.

The USB 2.0 pair is implemented by a Microchip USB3320 PHY interfacing with the PS-side controller of MPSoC over ULPI. The USB 3.1 lane is implemented using a PS-GTR transceiver lane.

In the box you can find a USB Type-C Legacy Adapter reference as CAR3G1-3 in the Type-C specifications. It has a Full-Featured Type-C plug on one end and a USB 3.1 Standard-A receptacle on the other. Use it to connect non-Type-C USB 2.0 or USB 3.1 devices to the Genesys ZU.

8.2. USB 2.0 Host

Host-only USB 2.0 functionality is implemented by a Microchip USB3320 PHY and a Microchip USB2513B hub. The PHY is wired to the PS-side controller of MPSoC over ULPI. The hub has three Downstream-Facing Ports. Two of these connect to a dual-stacked Type-A connector, providing 0.5A current per port. The third port is connected to the MiniPCIe slot, an embedded USB port. This allows interfacing with Bluetooth modules, for example.

8.3. USB 2.0 - JTAG/Serial Bridge

The micro Type-B connector J8 connects to an FTDI FT4232HQ USB bridge. It provides a JTAG interface for programming and debugging, one UART interface connected to the MPSoC and one UART interface connected to the Platform MCU. The UART interfaces are exposed as standard COM ports. Their exact designator is determined upon enumeration, but the first one will always connect to the

MPSoC and the second one to the Platform MCU. The MPSoC UART interface is wired to the PS-side MIO Bank 500: UART_TXD_IN to MIO18 and UART_RXD_OUT to MIO19. Signal names that imply direction are from the point-of-view of the DTE (Data Terminal Equipment), in this case the PC (e.g. UART_TXD_IN is the TXD signal of the DTE, meaning it is an output of the DTE and an input of the DCE). The Diligent USB-JTAG function and the USB-UART functions behave independent of one another. Support for USB-JTAG in Vivado is expected in version 2020.1. Read more in section JTAG Boot Mode (https://reference.digilentinc.com/reference/programmable-logic/genesys-zu/reference-manual#jtag_boot_mode).

9. Multimedia

9.1. DisplayPort Source

The dual-lane mini DisplayPort connector J27 is wired to a PS-side DisplayPort Controller via two PS-GTR transceiver lanes. Resolutions up to 4Kx2K@30fps are supported at a maximum 5.4Gbps line rate.

The DisplayPort Auxiliary interface is wired to PL-side pins, therefore routed through EMIO. The EMIO port `dp_aux_data_oe_n` needs to be inverted to fit the active-high polarity of the `DP_AUX_DOE` signal on the Genesys ZU. This can easily be done in PL logic and our <https://github.com/Diligent/Genesys-ZU-OOB-hw> (<https://github.com/Diligent/Genesys-ZU-OOB-hw>) repo shows how to do it.

9.2. HDMI Source

TODO for 5EV

9.3. HDMI Sink

TODO for 5EV

9.4. Audio Codec

The Genesys ZU board includes an Analog Devices ADAU1761 SigmaDSP audio codec (IC39) complementing its multimedia features. Four 1/8" (3.5mm) audio jacks are available for line-out (J20-green), headphone-out (J19-black), line-in (J22-blue), and microphone-in (J21-pink). Each jack carries two channels of analog audio (stereo), with the exception of the microphone input, which is mono.

To record or play back audio the audio data needs to be converted. The audio codec bridges the gap between the analog jacks and the digital FPGA pins. It connects to the PL side of the MPSoC. Analog-to-digital and digital-to-analog conversion is done at up to 24 bits and 96 kHz sampling rate. Digital audio data is carried to/from the FPGA on a serial, full-duplex interface, which supports several different formats, the default being I2S. This interface is clocked by the FPGA through BCLK by default, but the codec can be configured to provide the clock itself.

Configuring the audio codec can be done over I²C. It responds to slave address 0b0111011 on a dedicated I²C bus, followed by a 16-bit register address and one or more data bytes. These registers control every functional aspect of the codec.

The codec is clocked from the FPGA through the Master Clock (MCLK) pin. A clock must be provided for the codec to function, including the I²C port. The exact frequency depends on the desired sample rate and whether PLL will be used, but 12 MHz is a good start.

For proper use, the concept of audio paths needs to be understood. Internal to the codec there are two signal paths: Playback and Record. Both are highly configurable analog paths with mixers and amplifiers that route audio signals through the chip. The Playback path is the output path that routes audio from different sources like the digital-to-analog converter or input mixers towards the headphone and line out jacks. On the other hand, the record path routes audio from the line-in and microphone-in towards the analog-to-digital converters. Having routing elements at every step enables signal mixing between channels, amplification, muting and bypass. However, it also means that each element has to be properly configured along the path.

Keep in mind that audio jack designations might differ from codec analog frontend designators. For example, the line-in jack connects to the AUX port of the codec. The microphone jack is wired to the IN port. Also, notice that although some ports offer differential amplifiers and signaling, they are not used on the Genesys ZU. For example, the OUT port is differential, comprising 4 pins: LOU_{TP}, LOU_{TN}, ROU_{TP}, and ROU_{TN}. However, the N-side of the differential pairs is left floating, while the P-side connects to the jack.

At the very least an audio-aware FPGA design should do the following:

1. Provide MCLK for the audio codec.

2. Use an I²C master controller to configure the core clocking, sample rates, serial interface format and audio path.
3. Send or receive audio samples over the serial audio data channel for playback or record.

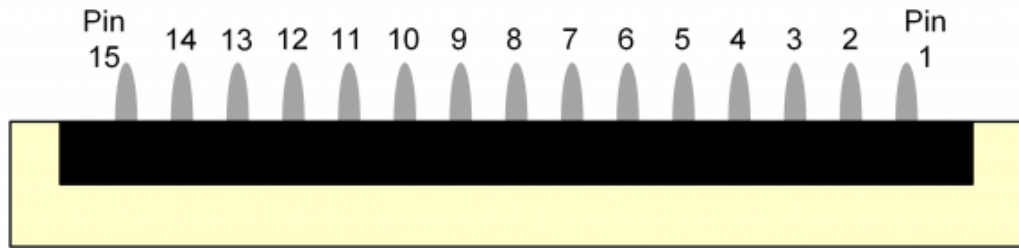
More advanced users might want to try additional features of the ADAU1761. For example, the on-chip SigmaDSP core can be programmed to do user-defined digital signal processing.

All relevant information can be found in the [ADAU1761 datasheet \(http://www.analog.com/media/en/technical-documentation/data-sheets/ADAU1761.pdf\)](http://www.analog.com/media/en/technical-documentation/data-sheets/ADAU1761.pdf).

9.5. MIPI/Pcam Ports

The two MIPI/Pcam ports included on the Genesys ZU are 15-pin, 1 mm pitch, zero insertion force (ZIF) connectors designed specifically for attaching camera sensor modules to host systems. It builds on the Pcam connector standard introduced on the Digilent Zybo Z7, but allows for bi-directional D-PHY lanes thanks to direct I/O support in the UltraScale+ architecture. Therefore, it supports MIPI DSI applications too, while remaining backward compatible with MIPI CSI-2 Pcam modules, like the Digilent Pcam 5C.

The Pcam connector pin-out is rigidly defined and includes a two lane MIPI CSI-2 bus for camera data, an I²C bus for camera configuration, two additional general purpose signals, and 3.3 V for powering the camera module, as depicted in Figure 9.5.1 and Table 9.5.1. Digilent is developing a catalog of Pcam peripheral camera modules with various different types of sensors that all conform to this pin-out. The pin-out was also chosen so that many camera modules designed to work with the Raspberry Pi will also work when connected to the Pcam port.



**Pin 1 will be indicated with a small white circle on the PCB*

(https://reference.digilentinc.com/_detail/reference/programmable-logic/zybo-z7/zybo-z7-pcam-pins.png?id=reference%3Aprogrammable-logic%3Agenesys-zu%3Areference-manual)

Figure 9.5.1: Pcam Pin-out

Table 9.5.1: Pcam Pin-out

Pin Number	Function	Genesys ZU Implementation
1	<u>GND.()</u>	<u>GND.()</u>
2	MIPI D-PHY Lane 0 (-)	Connected to a 1.2V VCCO HP bank
3	MIPI D-PHY Lane 0 (+)	Connected to a 1.2V VCCO HP bank
4	<u>GND.()</u>	<u>GND.()</u>
5	MIPI D-PHY Lane 1 (-)	Connected to a 1.2V VCCO HP bank
6	MIPI D-PHY Lane 1 (+)	Connected to a 1.2V VCCO HP bank
7	<u>GND.()</u>	<u>GND.()</u>
8	MIPI D-PHY Clock (-)	Connected to a 1.2V VCCO HP bank
9	MIPI D-PHY Clock (+)	Connected to a 1.2V VCCO HP bank
10	<u>GND.()</u>	<u>GND.()</u>
11	<u>GPIO.()</u> /Power enable	Connected to a 1.2V VCCO HP bank via a 3.3V level-translator

Pin Number	Function	Genesys ZU Implementation
12	<u>GPIO_0</u> /Clock feedback	N/C
13	SCL	Connects to branch 0 (MIPI A) and branch 1 (MIPI B) of the main I ² C bus multiplexer
14	SDA	Connects to branch 0 (MIPI A) and branch 1 (MIPI B) of the main I ² C bus multiplexer
15	3V3	3.3 V Power rail

Pcam modules are connected to the Pcam host port using a flexible flat cable (FFC). To connect the cable to the Genesys ZU follow these instructions:

1. Locate the Pcam connector.
2. Pull directly up on the white colored tab to open the connector.
3. Insert the FFC with the contacts facing the left edge, away from the center of the Genesys ZU.
4. Ensure the FFC is fully inserted.
5. Gently press down on both sides of the white colored tab to latch the FFC into the connector.
6. The FFC is now connected properly.

10. Expansion Ports

10.1. Mini PCIe / mSATA

J13 socket implements a versatile expansion option for adding SSD, WLAN, Bluetooth or WWAN modules to the Genesys ZU. It is compatible with PCI Express Mini card types F1/F2 (Full-Mini) and H1/H2 (Half-Mini) and mSATA card types Mini and Full size. Mechanical compatibility is assured by the relocatable stand-offs included with the board. Electrically, the SATA lane and the PCIe x1 lane share the same PS-GTR transceiver lane. Therefore, it is up to the MPSoC configuration to enable either the SATA or the PCIe Root controller and map it to the GTR lane. Mini PCIe modules can also make use of the embedded USB 2.0 port wired to the on-board USB hub and the MPSoC USB1 controller up the chain.

10.2. Low-Pin Count FMC Connector

The Genesys ZU includes an FPGA Mezzanine Card (FMC) Standard-conforming carrier card connector that enables connecting mezzanine modules compliant with the same standard. Genesys ZU-based designs can now be easily extended with custom or off-the-shelf high-performance modules.

The actual connector used is a 160-pin Samtec ASP-134603-01, the low-pin count, 10mm stacking height variant of the standard. All user defined signals are bonded to the PL-side of the MPSoC to HP banks 64 and 65. On the 5EV variant the multi-gigabit transceiver lane is also wired to the PL-side GTH transceiver, sharing the lane with the SFP+ slot.* The 34 differential pairs are powered by the Genesys ZU VADJ rail adjustable in the 1.2 V - 1.8 V range. There is also a 12 V rail wired to the FMC connector, which can supply up to 1 A to the mezzanine card.

FMC mezzanine cards are NOT hot-swappable. Connecting or disconnecting a card from the Genesys ZU while the board is powered on may cause damage to the mezzanine card and/or the board, and is to be avoided.

The UltraScale+ HP banks support the highest data rates available in the non-GT I/O architecture over the FMC connector.

The pin-out of the FMC connector can be found in the XDC constraints file available on reference.digilentinc.com. The schematic also shows the mapping between FMC connector pins and FPGA pins. Keep in mind that pin designators for the connector are not the same as pin designators for the FPGA specified in the XDC constraints file. For example, the connector pin with designator H28 and named LA24_P is wired via net FMC_LA24_P to the FPGA pin with designator AF7 and named IO_L11P_T1U_N8_GC_64. In the constraints file FMC_LA24_P will need to be location constrained to AF7.

For above-gigabit speed rates on the 5EV variant, the gigabit transceiver lane is wired to a multiplexer. The SFP+ slot and the FMC share a single GTH lane through the multiplexer. Therefore, only one of them can be used at any time. The FMC user defined pins are not affected by this limitation. Each transceiver lane includes a receive pair and a transmit pair. Lane DP0 is wired through the mux to quad 224. The

reference clock GBTCLK0 goes to a jitter filter and can be sent to MGTREFCLK0 pins of the same quad. Table 10.2.1 shows how the FMC gigabit signals are mapped to pins and GTH primitives. Refer to the UltraScale Architecture GTH Transceivers User Guide (📄 ug576 (https://www.xilinx.com/support/documentation/user_guides/ug576-ultrascale-gth-transceivers.pdf)) for more information.

Table 10.2.1: FMC Gigabit Signals Mapping

Quad	Primitive		Pin type	Pin	FMC signal
224	GTHE4_CHANNEL	X0Y7	MGTHTXP/N3	N4/N3	DP0_C2M_P/N
			MGTHRXP/N3	P2/P1	DP0_M2C_P/N

For FMC designs which use FMC_LA_07_P/_N and/or FMC_LA15_P/_N lines with an I/O standard which needs DCI, please make sure you add the DCIRESET primitive to the respective designs. Refer to the UltraScale Architecture SelectIO Resources User Guide (📄 ug571 (https://www.xilinx.com/support/documentation/user_guides/ug571-ultrascale-selectio.pdf)) chapter “Special DCI Requirements in Some Banks” for more information.

Table 10.2.2: Maximum length mismatch including MPSoC package delay.

Signal Group	Length matching	
	Intra-pair	Inter-pair
LA[00-33], CLK[0-1]_M2C	1 mm	10 mm
DP0*, GBTCLK0*	0.14 mm	100 mm

Add impedance, net length, length matching and S-parameter reports

10.3. Zmod

The Zmod port uses the SYZYGY Standard interface to communicate with installed SYZYGY pods. The port is compatible with version 1.1 of the SYZYGY specification from Opal Kelly.

Attachment detection is implemented by the Platform MCU and pod presence is communicated to the MPSoC over the SYZYGY_DETECTED signal. For now, it is up to the MPSoC to read the port's SYZYGY DNA and implement SmartVIO functionality by requesting a compatible voltage on the VADJ rail. SYZYGY DNA is accessible on branch 5 of the I²C multiplexer on the main I²C bus at address 0110000b.

SYZYGY pods are NOT hot-swappable. Connecting or disconnecting a pod from the Genesys ZU while the board is powered on may cause damage to the pod and/or the board, and is to be avoided.

Each SYZYGY Standard interface contains 14 single-ended I/O pins (2 of which I²C), 8 differential I/O pairs (which can alternatively be used as 16 additional single-ended I/O pins), and two dedicated differential clocks - one for input and one for output. The Zmod port is wired to PL-side MPSoC banks powered by the VADJ rail, sharing them with FMC signals. Therefore if both an FMC mezzanine card and a Zmod are connected to the Genesys ZU, a common voltage supported by both needs to be chosen for VADJ. The differential pairs were prioritized and wired to HP banks, allowing the maximum data rates supported by the SelectI/O architecture. However, the single-ended pins are wired to an HD bank, limiting the data rate to 250 Mb/s according to the Zynq UltraScale+ MPSoC Data Sheet: DC and AC Switching Characteristics (📄 ds925) (https://www.xilinx.com/support/documentation/data_sheets/ds925-zynq-ultrascale-plus.pdf). Template constraints for the Zmod port can be found in the Genesys ZU's Master XDC file, available through Digilent's 📄 digilent-xdc (<https://github.com/Digilent/digilent-xdc>) repository on Github.

The two standoff locations on the sides of the Zmod port, on the top side of the Genesys ZU board, are hexagonal, 5 mm tall with M2.5 x 0.45 internal threaded holes. For mounting Zmods on Genesys ZU, two screws with M2.5 x 0.45 thread and 3mm thread length would be needed.

For more information on the SYZYGY standard, see 📄 syzygyfpga.io (<https://syzygyfpga.io/>).

10.3.1. SYZYGY Pod Compatibility

The Genesys ZU Zmod port is compatible with a variety of different SYZYGY pods. Information required to determine if the Genesys ZU is compatible with a certain pod is summarized in Table 10.3.1.1.

Table 10.3.1.1: SYZYGY Compatibility

Parameter	Port A (STD)
Port Type	Standard
	Single-Width
Total 5V Supply Current	1 A
Total 3.3V Supply Current	3.5 A (shared with FMC)
VIO Supply Voltage Range	1.2V to 1.8V
Total VIO Supply Current	2.1 A (shared with FMC)
Port Groups	Group 1: A
I/O Count	28 total (8 DP)
Length Matching	10 mm inter-pair, 1mm intra-pair

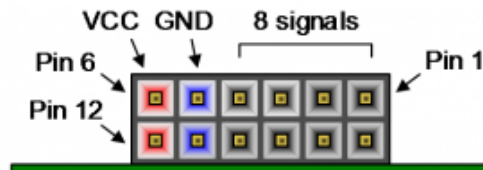
10.3.2

Fix Me! *Is there any information on this not encapsulated by the syzygy spec.()?*

Add S-parameter reports

10.4. Pmod

Pmod ports are 2×6, right-angle, 100-mil spaced female connectors that mate with standard 2×6 pin headers. Each 12-pin Pmod port provides two 3.3V V_{CC} signals (pins 6 and 12), two Ground signals (pins 5 and 11), and eight logic signals, as shown in Figure 10.4.1 below. For more information regarding the power supply specifications of the Pmod ports, refer to the [Digilent Pmod™ Interface Specification \(https://www.digilentinc.com/Pmods/Digilent-Pmod_%20Interface_Specification.pdf\)](https://www.digilentinc.com/Pmods/Digilent-Pmod_%20Interface_Specification.pdf), “Power Supply” section.



(https://reference.digilentinc.com/_detail/basys3-pmod_connector.png?id=reference%3Aprogrammable-logic%3Agenesys-zu%3Areference-manual)

Figure 10.4.1: Pmod port

Diligent produces a large collection of Pmod accessory boards that can attach to the Pmod ports to add ready-made functions like A/D's, D/A's, motor drivers, sensors, and other functions. See [www.digilentinc.com \(http://www.digilentinc.com\)](http://www.digilentinc.com) for more information. The vivado-library repository on the [Digilent Github \(https://github.com/Digilent/\)](https://github.com/Digilent/) contains pre-made IP cores for many of these Pmods that greatly reduce the work of integrating them into your project. This repository's `hierarchies` branch contains additional scripts and sources that can be used to speed up the process of integrating these cores. See the Pmod-related tutorials on the Genesys ZU Resource Center for help using them.

PMODS JA, JB, JC, JD

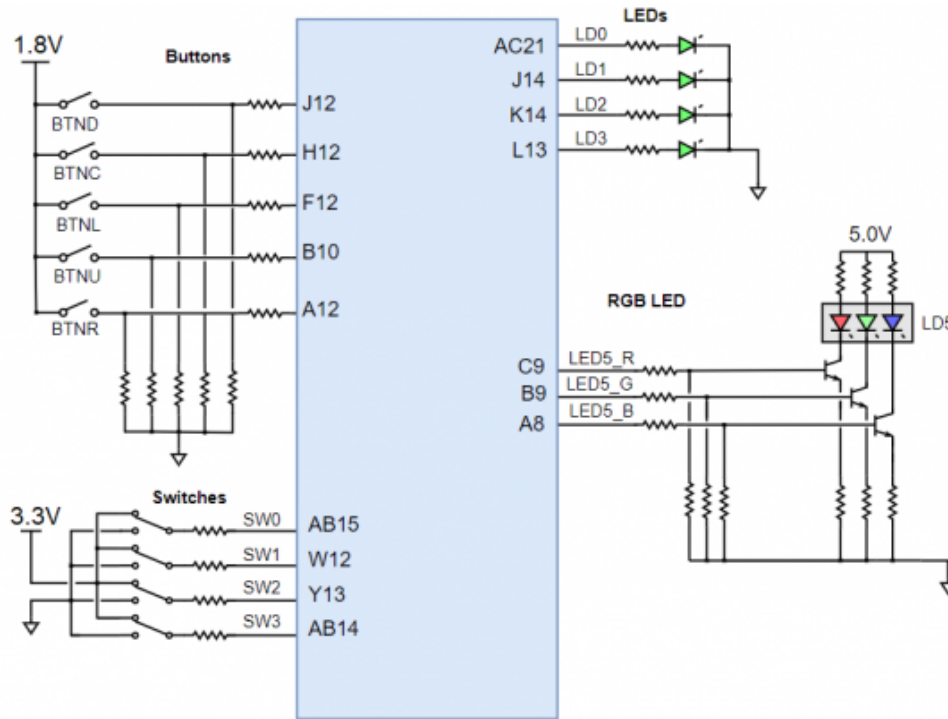
Dual Digital/Analog PMOD	PMOD JB	PMOD JC	PMOD JD
JA1_P:W10	JB1:AE13	JC1:E13	JD1:E15
JA2_P:AA11	JB2:AG14	JC2:G13	JD2:A14
JA3_P:AB10	JB3:AH14	JC3:B13	JD3:B15
JA4_P:Y9	JB4:AG13	JC4:D14	JD4:F15
JA1_N:Y10	JB7:AE14	JC7:F13	JD7:E14
JA2_N:AA10	JB8:AF13	JC8:C13	JD8:B14
JA3_N:AB9	JB9:AE15	JC9:C14	JD9:D15
JA4_N:AA8	JB10:AH13	JC10:A13	JD10:A15

10.5. Dual Digital/Analog Pmod

On the Genesys ZU, one of the Pmod connectors is not like the others. Designated by JA, the Pmod connector is wired to pins that can serve as auxiliary inputs to the system monitor `ADC()` inside the MPSoC. These pins are in a VADJ-powered bank, so the `VCC()` pins are not powered from the 3.3 V rail, like on regular Pmods, but from VADJ. VADJ on the Genesys ZU is in the 1.2 V - 1.8 V range. Pins 1-7, 2-8, 3-9, 4-10 are paired and routed differentially. Although these pins can be used in digital mode, the particularities of this connector must be taken into account when connecting Pmod modules to it.

11. Basic I/O

The Genesys ZU includes five push-buttons, four slide switches, one tri-color `LED()` and four green LEDs connected to the Zynq PL, as shown in Figure 11.1 below. These I/Os are connected to the Zynq via series resistors to prevent damage from inadvertent short circuits (a short circuit could occur if a pin assigned to a push-button was inadvertently defined as an output). The five push-buttons are arranged in a plus-sign configuration (center, left, right, up and down buttons, respectively).



(https://reference.digilentinc.com/_detail/reference/programmable-logic/genesys-zu/basic_io_genesys_zu.png?id=reference%3Aprogrammable-logic%3Agenesys-zu%3Areference-manual)

Figure 11.1: Genesys ZU PL Basic I/O

Genesys ZU also has two push-buttons and one green `LED_0` connected to the Zynq PS: push-buttons `BTN0` and `BTN1` are connected to `MIO11` and `MIO10`, respectively, and the green `LED_0` is connected to `MIO21`.

11.1. Push-Buttons

The push-buttons are “momentary” switches that normally generate a low output when they are at rest, and a high output only when they are pressed.

11.2. Slide Switches

The slide switches generate constant high or low inputs depending on their position: when the slide is in a low position (i.e. close to the lower board edge), the generated input is low; when the slide is in a high position (i.e. close to the center of the board), the generated input is high.

11.3. Tri-Color LED

The tri-color `LED_0` has three input signals that drive the cathodes of three smaller internal LEDs: one red, one blue, and one green. Driving the input signal corresponding to one of these colors low will illuminate the internal `LED_0`. The input signals are driven by the Zynq PL through a transistor, which inverts the signals. Therefore, to light up the tri-color `LED_0`, the corresponding PL pins need to be driven high. The tri-color `LED_0` will emit a color dependent on the combination of internal LEDs that are currently being illuminated. For example, if the red and blue signals are driven high and green is driven low, the tri-color `LED_0` will emit a purple color.

Note: Digilent strongly recommends the use of pulse-width modulation (PWM) when driving the tri-color LEDs. Driving any of the signals to a steady logic '1' will result in the `LED_0` being illuminated at an uncomfortably bright level. This can be avoided by ensuring that none of the tri-color signals are driven with more than a 50% duty cycle. Using PWM also greatly expands the potential color palette of the tri-color `LED_0`. Individually adjusting the duty cycle of each color between 0% and 50% causes the different colors to be illuminated at different intensities, allowing virtually any color to be displayed.

11.4. Green LEDs

The individual high-efficiency LEDs are anode-connected to the Zynq Ultrascale+ via 330-ohm resistors, so they will turn on when a logic high voltage is applied to their corresponding I/O pin.

12. Platform Management

Tying all the features of the Genesys ZU together into a computing platform requires an embedded controller independent of the MPSoC. We call it Platform MCU. Part of the platform is the coin battery, the fan, a temperature sensor inside the MPSoC and Power Management Units (PMU). Management is done through dedicated signals or over the main I²C bus.

12.1. Main I²C bus

Almost all I²C-capable peripherals are accessible through the main I²C bus. Multiple masters have access:

- Platform MCU in the Auxiliary 3.3 V domain through MUX_SCL and MUX_SDA,
- 3-pin header J36 in the Auxiliary 3.3 V domain through MUX_SCL and MUX_SDA,
- MPSoC PS-side in the Main 3.3 V domain through MUX_SCL_LS and MUX_SDA_LS in Bank 500,
- MPSoC PL-side in the Main 3.3 V domain through MUX_SCL_LS and MUX_SDA_LS in Bank 46*/26.

It follows that any I²C master controller implementation in the MPSoC must be multi-master tolerant and must support arbitration.

The only slave device that can be accessed after power-on is an 8-channel I²C multiplexer, a TI TCA9548A, responding to address 1110000b. The rest of the slaves are distributed on the eight channels numbered from 0 to 7. To access a device on a particular channel, address the multiplexer first and write a single byte to it with the bit corresponding to the desired channel set to 1. After the STOP condition, the multiplexer will unite all the enabled channels and the main bus. Now a slave on the enabled channels can be accessed by its respective address. Make sure that there are no address conflicts on enabled channels. For example, having a Pcam 5C connected to each of the two MIPI/Pcam ports, and enabling both channel 0 and 1 simultaneously will cause a conflict. This might be desired, allowing writing the same data to both Pcams, but reading is problematic and arbitration will happen. The recommended approach is having just one of the channels enabled at any time.

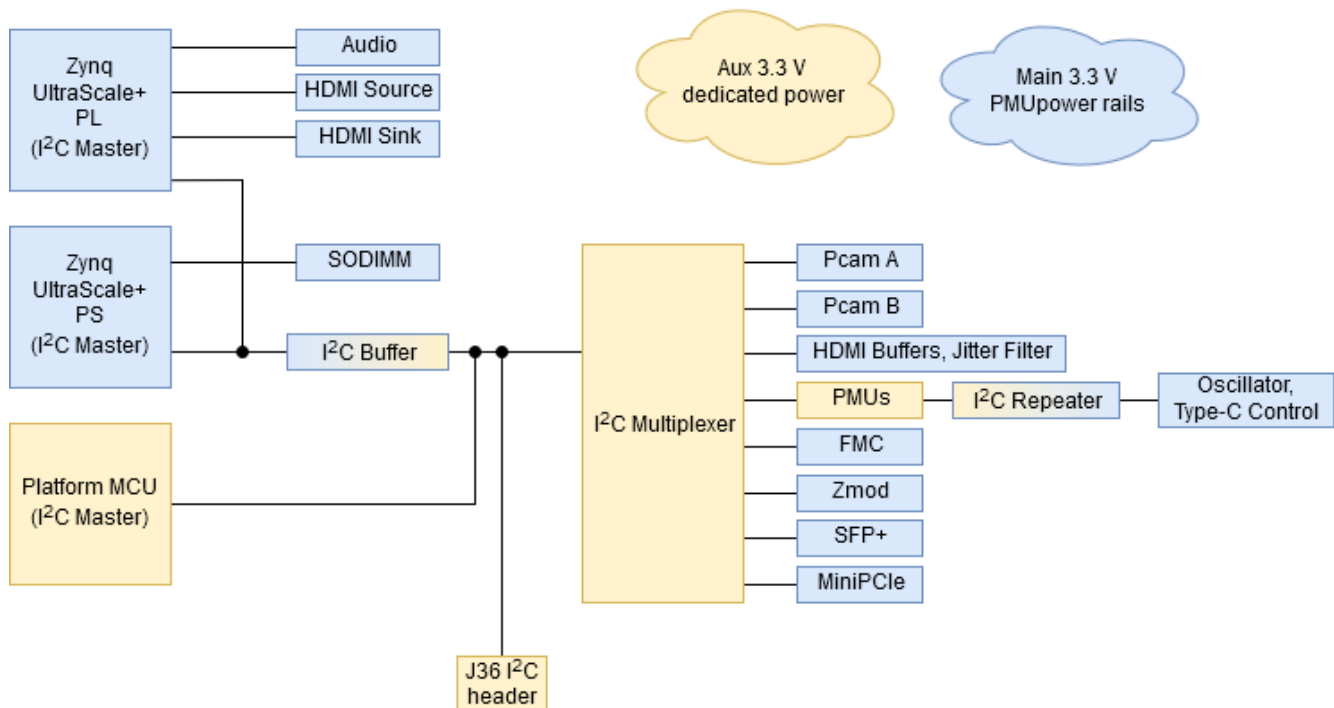


Figure 12.1.1: Genesys ZU I²C topology

12.2. Platform MCU

The Platform MCU is implemented by a Microchip ATmega328PB. It is on the auxiliary 3.3V power domain, immediately available after power-up. This power domain is independent of the PMUs which provide main power, giving the Platform MCU control over main power. It also shares the main I²C bus with the MPSoC, giving it access to all the critical peripherals. Other features include MPSoC temperature sensing, fan speed control, and VADJ voltage setting.

The Platform MCU program memory has two sections:

- Application where the firmware resides

- Bootloader where the bootloader resides.

12.2.1. Application Section

The Platform MCU has the following interfaces on Genesys ZU:

- FPGA temperature sensing and fan control with feedback
- PMU interfaces
- SYZYGY connector interface
- FMC connector interface
- UART interface to the PC

The Platform MCU monitors the FPGA temperature and adjust the fan speed accordingly. Also, it checks that the actual fan speed is close to the set value and reports a fault to the PC otherwise. FPGA temperature monitoring is performed using the temperature diode inside the FPGA.

At board power-up, Platform MCU detects if a SYZYGY peripheral board (“Pod”) is connected to Genesys ZU and signals this to the FPGA. If a SYZYGY Pod was detected, the SYZYGY_DETECTEDN pin (H11) is driven LOW. Otherwise the pin is driven HIGH.

If VADJ_AUTO pin (G10) from FPGA is LOW, the Platform MCU establishes the correct VADJ voltage value based on VADJ_LEVEL1 (AC13) and VADJ_LEVEL0 (AC14) input pins, regardless of whether a syzygy pod and/or FMC mezzanine module is connected. Table 12.2.1.1 presents the VADJ level encoding.

Table 12.2.1.1: VADJ levels encoding

VADJ_LEVEL1	VADJ_LEVEL0	VADJ level
0	0	VADJ disabled
0	1	1.2V
1	0	1.5V
1	1	1.8V

Although neither FMC mezzanine modules nor SYZYGY pods are designed to be plug-and-play and their detection needs to be done only at board power-up, the VADJ voltage value can still change during board operation. Due to this, VADJ_AUTO signal value can change during board operation. The Platform MCU will detect the pin change and will adjust the VADJ voltage level according to the actual VADJ_LEVEL1 and VADJ_LEVEL0 pins state. The voltage rail will reach its power good threshold in maximum 60 ms after the falling edge of VADJ_AUTO. The power good threshold is set to 100 mV less than the nominal voltage.

To set the desired VADJ level you have to:

1. Drive the VADJ_LEVEL1 and VADJ_LEVEL0 to encode the desired VADJ level.
2. Generate a falling edge condition on VADJ_AUTO.

In the current implementation, the FPGA must detect the correct VADJ level required by SYZYGY and FMC modules and must set the VADJ_LEVEL1 and VADJ_LEVEL0 signals accordingly. The VADJ_LEVEL1 and VADJ_LEVEL0 signals will be taken into account by the Platform MCU only if the VADJ_AUTO signal is driven LOW by the FPGA. If VADJ_AUTO is HIGH, the VADJ power rail is disabled.

On Genesys ZU there is a LED_() labeled with PMCU. This is the status led that is used by the Platform MCU to display the system fault that has the highest priority. The blinking pattern for each fault is presented in Table 12.2.1.2.

After Platform MCU startup, if no issues were encountered, this LED_() should blink in a pattern Long Blink – Short Pause - Long Blink – Long Pause then it should turn off.

- A “long blink” and a “long pause” last for approximately 1 second each;
- A “short blink” and a “short pause” last for about 200ms each;

Table 12.2.1.2: Fault blink patterns

Blink Pattern (Repeated)	Issue	Priority	Comments
--------------------------	-------	----------	----------

Blink Pattern (Repeated)	Issue	Priority	Comments
Long Blink – Long Pause	Fan Speed Fault	0	Register 0x04 Bit 0

The Platform MCU exposes to the PC a register interface, accessible via UART. The full register map is shown in Table 12.2.1.3.

Table 12.2.1.3: Register map

Offset	Register Name	Size [bits]	R/W	Description
0x00	ID Register	8	R	It contains a fixed value, used for determining if the Platform MCU is alive and responding over UART (0x00).
0x01	Firmware Version	8	R	It contains the firmware major version on bits 7-4 and the minor version on bits 3-0
0x02	Scratch Register	8	R/W	Read/write register used to test both the write and the read register interfaces.
0x03	Measured FPGA Core Temperature	8	R	FPGA Temperature value, as computed by the Platform MCU based on the thermal diode measurement.
0x04	Measured Fan Speed	16	R	FPGA Fan Speed value in rpm, as computed based on the fan feedback pin. Upon read, the low byte is received first.
0x06	Fan Speed Fault	8	8	Bits 7-1: Unused Bit 0: '1' if fan speed is outside the expected range; '0' otherwise
0x07-0x0E	Reserved for future functionality			
0x0F	Platform MCU Control	8	R/W	Bits 7-2: Unused Bit 1: Fault Status: - '1' = Clear all faults - [Default] '0' = Do nothing Once written with '1', this bit returns by itself to '0' after all the faults have been cleared. Bit 0: I2C Communication to PMUs Control: - '1' = I2C Communication to PMUs Disabled - [Default] '0' = I2C Communication to PMUs Enabled
0x10-0x70	Reserved for future functionality			

When the PC wants to access a Platform MCU register, it needs to respect the following protocol:

- It needs to send at first the address byte. Bits 7 to 1 of this byte contain the register start address (found in the first column from Table 12.2.1.3). Bit 0 of this byte is logic '1' for read transactions, and logic '0' for write transactions.
- It then needs to send a second byte, containing the number of bytes to read/write.
- For write transactions, it then needs to send the actual values to be written in the Platform MCU registers. The number of bytes sent in this phase needs to match the value from the previous byte ("the number of bytes to read/write"), otherwise communication with the Platform MCU will hang.
- For read transactions, the Platform MCU will then start to send the requested register values.
- For write transactions, the maximum length is 256 bytes. After such a transaction, wait for a minimum of 100ms before sending a new transaction (read or write) to the Platform MCU.
- For consecutive write transactions, the total maximum length is 256 bytes. After such a transaction, wait for a minimum of 100ms before sending a new transaction (read or write) to the Platform MCU.

- All bytes to be sent to the PMCU need to have their nibbles converted to `ASCII()` characters prior to being sent; e.g.: 0x0C in hex needs to be converted to “0C” in `ASCII()`.
- All characters received from the PMCU need to be converted from `ASCII()` to hex nibbles after being received; e.g. “0C” received in `ASCII()` format needs to be converted to 0x0C in hex.

Write transaction example. Let’s say the PC wants to clear all system faults. For this, it will send to the Platform MCU:

- Byte 0: 1E (i.e. Bits 7-1 = 0x0F which is the address of Platform MCU register; Bit 0 = ‘0’ meaning write transaction)
- Byte 1: 01 (i.e. the PC want to write 1 byte)
- Byte 2: 02 (value to write to register at address 0x0F - Clear all faults)

Read transaction example. Let’s say the PC wants to read Fan Speed Faults register from the Platform MCU. For this, it will send to the Platform MCU:

- Byte 0: 0D (i.e. Bits 7-1 = 0x06 which is the address of Fan Speed Fault register; Bit 0 = ‘1’ meaning read transaction)
- Byte 1: 01 (i.e. the PC wants to read 1 byte)

The Platform MCU will then send 1 byte back to the PC, containing the value of the Fan Speed Fault register.

To read a 16-bit register like Measured Fan Speed the PC will send to the Platform MCU:

- Byte 0: 09 (i.e. Bits 7-1 = 0x04 which is the address of Measured Fan Speed register; Bit 0 = ‘1’ meaning read transaction)
- Byte 1: 02 (i.e. the PC wants to read 2 bytes)

The Platform MCU will then send 2 bytes back to the PC. The first byte that is sent is the least significant one. If one receives 150C one have to swap the first byte with the second one and to convert the value to decimal: 150C → 0x0C15 → 3093 RPM.

The UART baudrate should be set to 115200/8/E/1 (115200 baud, 8 data bits, even parity, 1 stop bit).

12.2.2. Bootloader Section

In the program memory, along with the Firmware Application, there is a bootloader that launches the Application at power-up.

12.3. Fan

Mounted on the MPSoC heatsink, there is a 12 V fan with a 4-pin header. It can automatically be controlled by the Platform MCU based on the MPSoC temperature or set to the fixed full speed. This option is controlled by JP2 and is user-selectable.

Table 12.3.1: Fan jumper positions

Jumper JP2 “AUTO FAN”	Fan Speed
Set	Automatic
Not set	Full

12.4. Coin battery

A Seiko TS621E lithium rechargeable battery provides power to the MPSoC Battery Power Domain (BPD) through the `VCC()_PSBATT` pin. It is connected in parallel with a 100 uF capacitor. The BPD includes the real-time clock with a dedicated crystal oscillator and a `RAM()` available for storing a secure configuration key. The capacitor alone can provide power for approx. 15 minutes after main power is turned off. The battery will provide power after that.

The nominal voltage of the TS621E is 1.5 V and has a nominal capacity of 1.3 mAh. The Zynq UltraScale+ MPSoC Data Sheet: DC and AC Switching Characteristics (ds925) lists the maximum I_{CC_PSBATT} at 3.65 uA. Some leakage current exists through the charging diode, a maximum of 100 nA. Therefore, the capacity of the fully-charged battery is enough for a minimum of: $1.3 \text{ mAh} * 1000 \text{ uA} / 1 \text{ mA} / 3.75 \text{ uA} = 347 \text{ hours} = 14 \text{ days}$ without main power. Whenever main power is turned on the battery will be re-charged.

The battery is removable and can be replaced only by a 1.5 V, 6.8 mm rechargeable lithium coin battery. Do not use non-rechargeable batteries!

However, those prepared to void their warranty can physically remove the charging circuit by de-soldering D13 or R400. In this case any battery, even a non-rechargeable one, that meets MPSoC voltage specs can be used.

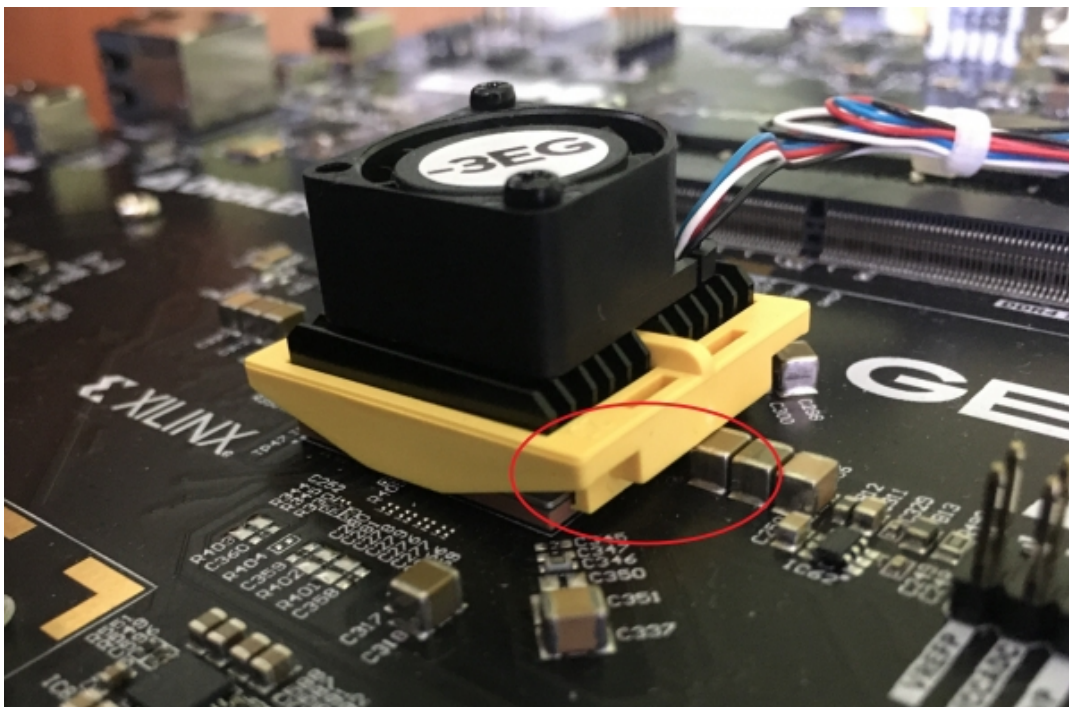
Hardware Errata

Although we strive to provide perfect products, we are not infallible. The Genesys ZU is subject to the limitations below.

Product Name	Variant	Revision	S/N	Problem	Status
Genesys ZU	-3EG	B	DAD9C39-DAD9D32, DADA13A, DADA13B	Fan assembly error resulting in sub-optimal mechanical hold. (https://reference.digilentinc.com/reference/programmable-logic/genesys-zu/reference-manual#fan_assembly_error)	Fixed in rev C.

1. Fan Assembly Error

Due to an assembly error the heatsink clip of the MPSoC cooling fan has been mounted incorrectly, resulting in sub-optimal mechanical hold between the heatsink and the MPSoC package. On the affected production build the two slots on the yellow clip are facing right, where tall components prevent seating the clip.



(https://reference.digilentinc.com/_detail/reference/programmable-logic/genesys-zu/5759d48d-d837-4f1b-97f1-89fd1533973f.jpg?id=reference%3Aprogrammable-logic%3Agenesys-zu%3Areference-manual)

Figure 1. Incorrect clip orientation prevents proper seating of the clip

For optimal mechanical hold the clip's lip should reach underneath the MPSoC package substrate on both the left and right sides. Since the heatsink is secured by double-sided tape too, the sub-optimal mechanical hold of the clip is expected to cause issues only in time with excessive material aging of the tape under high temperature swings and/or mechanical vibration. The correct orientation of the yellow heatsink clip is with the two slots facing left.

Workaround

The fan and the yellow clip are user-removable. Disconnect all cables from the Genesys ZU before starting the operation. To remove the fan use a suitable Philips screwdriver to loosen the two screws fixing the fan to the heatsink. Put the fan aside for a moment. Use a narrow flathead screwdriver in one of the clip slots to carefully disengage the lips of the clip from underneath the MPSoC package. The lips are fragile and without due care they can break off easily. Once the lip on both sides of the clip is above the MPSoC package, the clip can be removed. Rotate the clip 180 degrees so that the slots are on the left and re-install it on the MPSoC package. Hook the lip of the clip under the MPSoC package on one side first and press down the other side to fasten the clip. A step-by-step description of the clip (dis)assembly is described here: http://www.malico.com.tw/index.php?option=com_content&view=article&id=451&Itemid=406&lang=en (http://www.malico.com.tw/index.php?option=com_content&view=article&id=451&Itemid=406&lang=en). After the yellow clip is in place, the fan can be screwed back in the same place as before.