Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

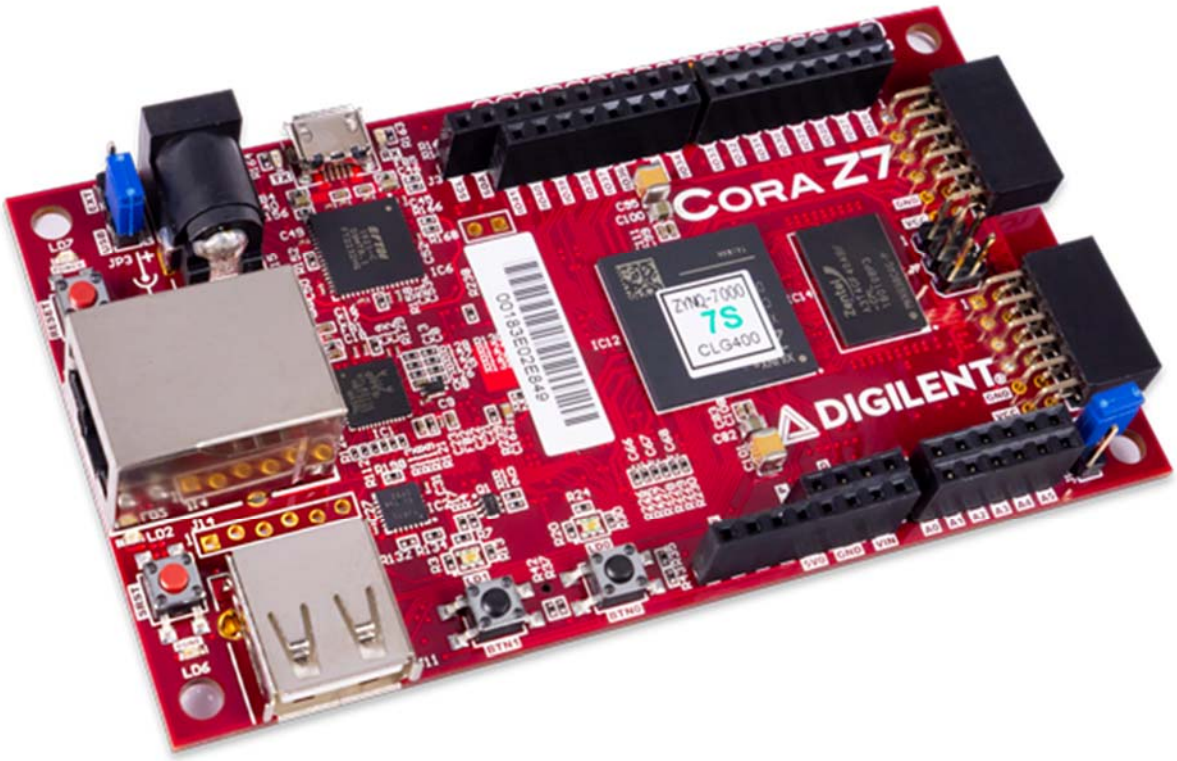
Contact us

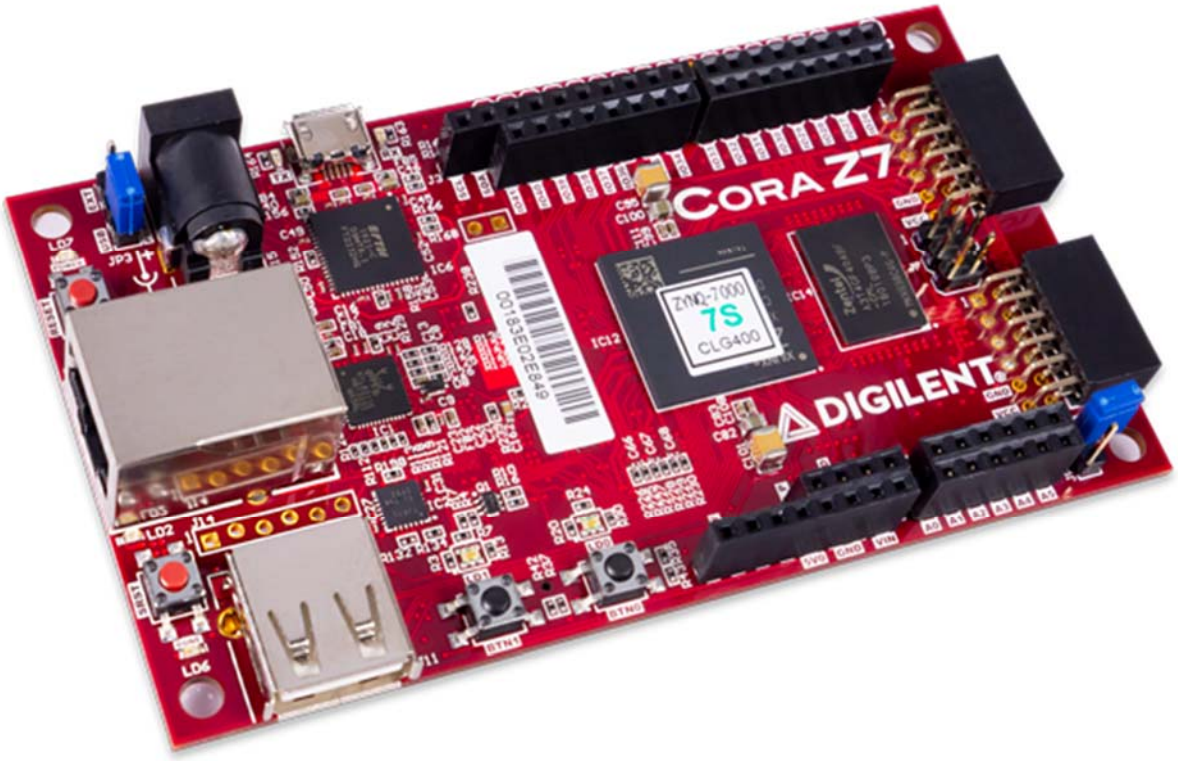
Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

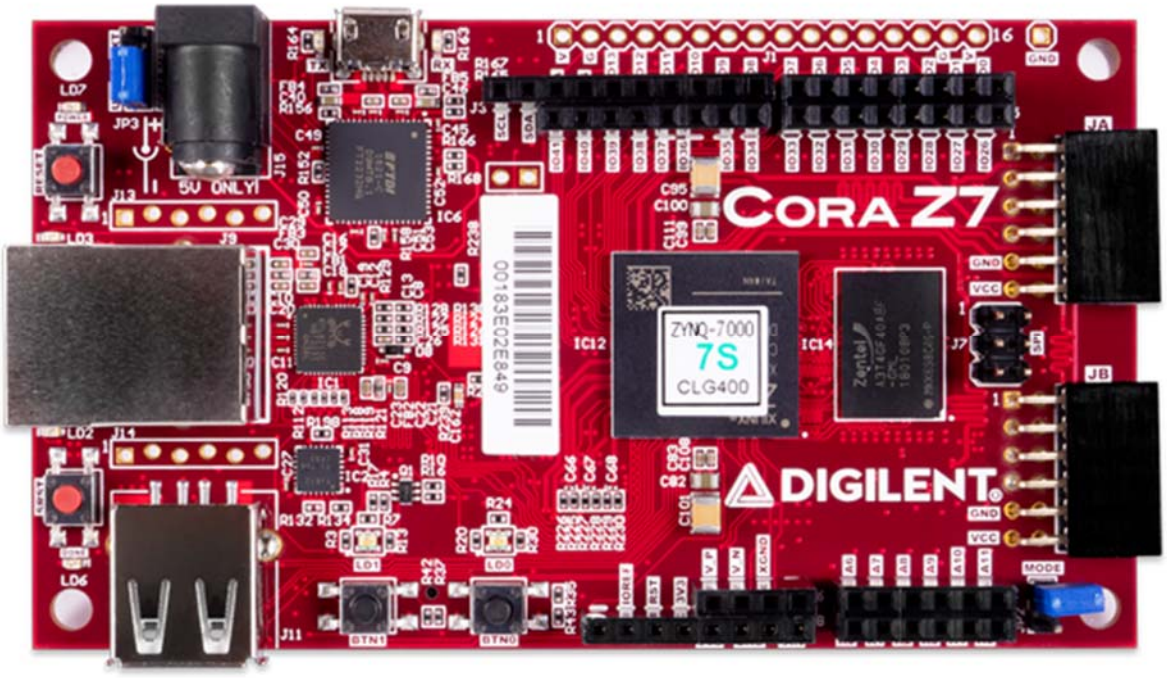
Cora Z7 Reference Manual

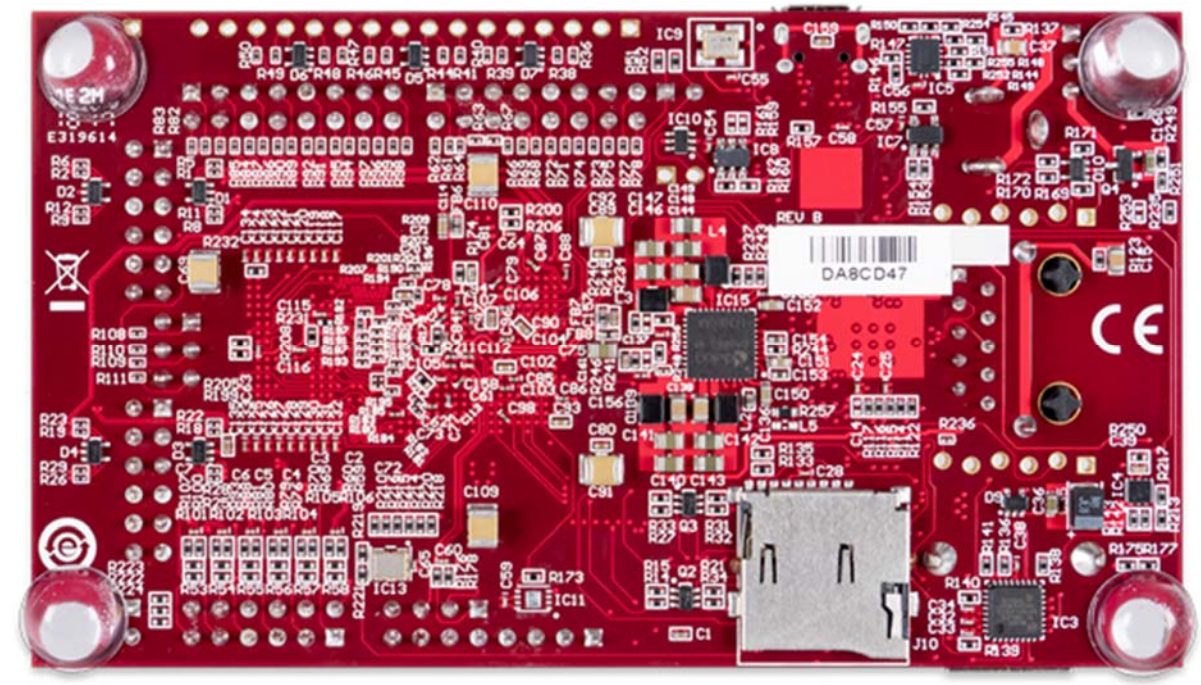
The Digilent Cora Z7 is a ready-to-use, low-cost, and easily embeddable development platform designed around the powerful Zynq-7000 All-Programmable System-on-Chip (APSoC) from Xilinx. The Zynq-7000 architecture tightly integrates a single or dual core 667MHz ARM Cortex-A9 processor with a Xilinx 7-series FPGA. This pairing grants the ability to surround the processor with a unique set of software defined peripherals and controllers, tailored for the target application.

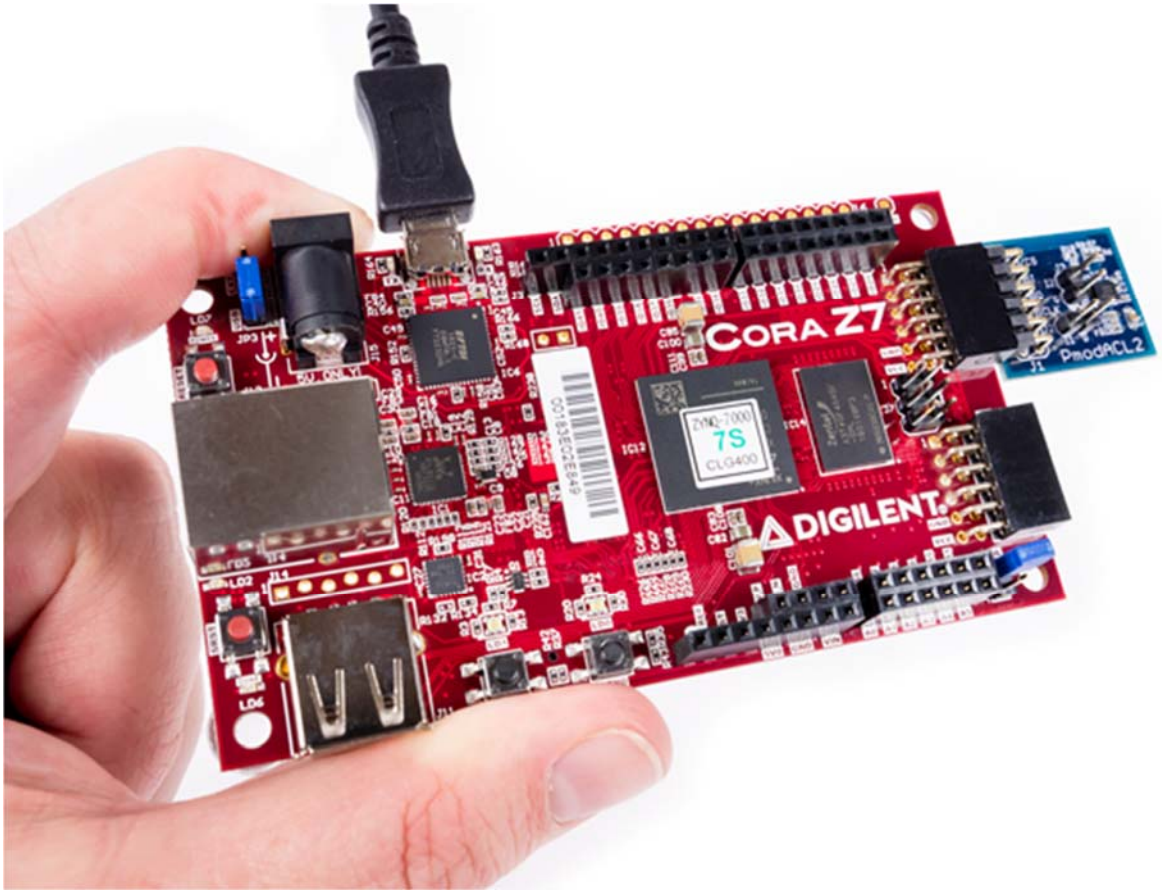
The Cora Z7's wide array of hardware interfaces, from a 1Gbps Ethernet PHY to analog-to-digital converters and general-purpose input/output pins, make it an ideal platform for the development of a vast variety of embedded applications. The small form factor and mounting holes make the Cora Z7 ready to be used as one component of a larger solution. The on-board SD Card slot, Ethernet, and Power solution allow the Cora Z7 to operate independently of a host computer.




CORA Z7
ZYNQ-7000
7S
CLG400
DIGILENT


CORA Z7
ZYNQ-7000
7S
CLG400
DIGILENT
5V ONLY!
JA
JB

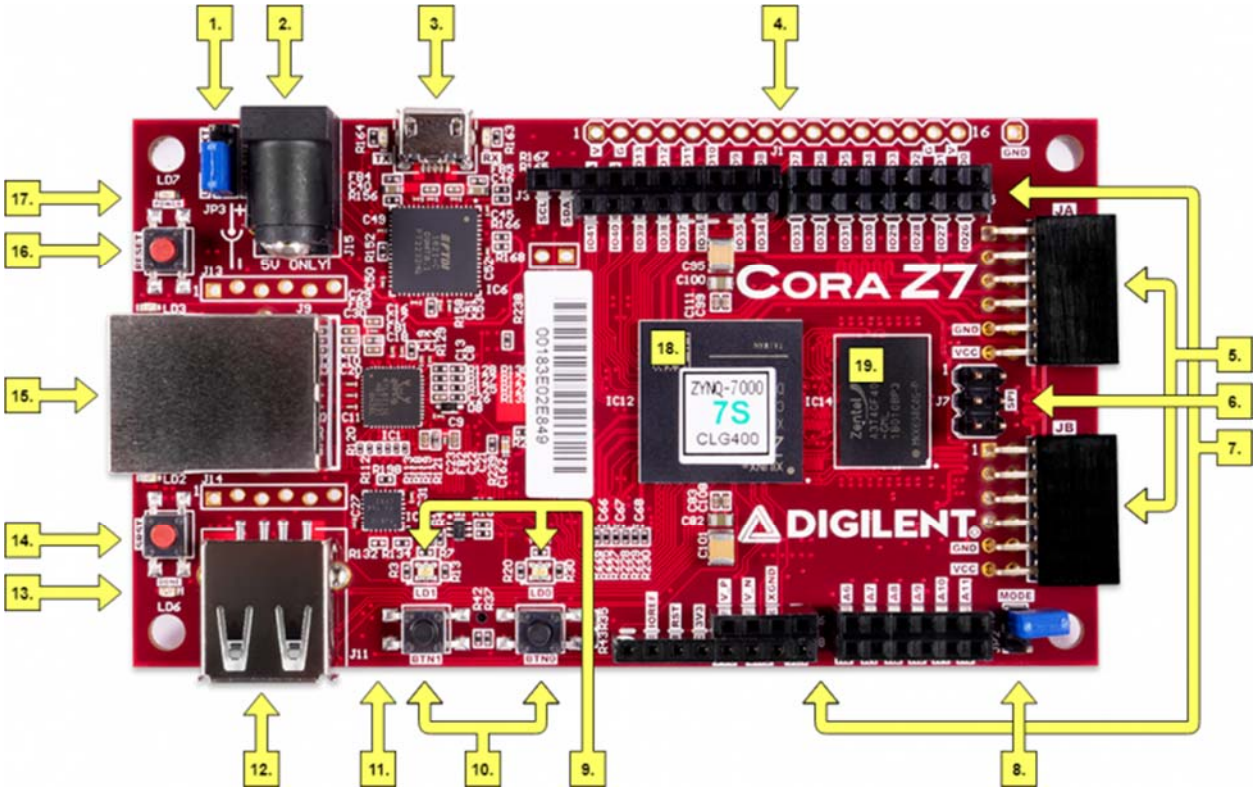

REV. B
DA8CD47
E319614


CORA Z7
DIGILENT
PmodACL2

Features

- **ZYNQ Processor**
  - 667MHz dual-core (*single-core) Cortex-A9 processor
  - FPGA Programmable logic equivalent to Artix-7 FPGA
    - 4,400 Programmable logic slices (*3,600)
    - 80 DSP slices (*60)
    - 270 KB of block RAM (*225 KB)
  - DDR3 memory controller with 8 DMA channels and 4 High Performance AXI3 Slave ports
  - High-bandwidth peripheral controllers: 1G Ethernet, USB 2.0, SDIO
  - Low-bandwidth peripheral controllers: SPI, UART, CAN, I2C
  - Dual-channel, 1 MSPS internal analog-digital converter
  - Programmable from JTAG and microSD card
- **Memory**
  - 512MB DDR3 with 16-bit bus @ 1050Mbps
  - microSD slot
- **Power**
  - Powered from USB or any 4.5V-5.5V external power source
- **USB and Ethernet**
  - Gigabit Ethernet PHY with 48-bit globally unique EUI-48/64™ compatible identifier available on sticker
  - USB-JTAG programming circuitry
  - USB-UART bridge
  - USB OTG PHY (supports host only)
- **Push-buttons and LEDs**
  - Two Push-buttons
  - Two RGB LEDs
- **Expansion Connectors**
  - Two Pmod connectors
    - 16 Total FPGA I/O
  - Arduino/chipKIT Shield connector
    - Up to 49 Total FPGA Digital I/O
    - 6 Single-ended 0-3.3V Analog inputs to XADC
    - 8 Differential 0-1.0V Analog inputs to XADC
  - Unloaded expansion header
    - 12 additional FPGA Digital I/O

(*Z7-07S variant in parentheses where different)



Callout	Description	Callout	Description
1	Power select jumper (Ext. supply / USB)	11	microSD card slot (underside of board)
2	Power jack (for optional ext. supply)	12	USB host port
3	Shared USB JTAG / UART port	13	FPGA programming DONE LED
4	Unloaded expansion header	14	Processor subsystem reset button
5	Pmod connectors	15	Ethernet port
6	SPI header (Arduino/ChipKIT compatible)	16	Power on reset button
7	Arduino/ChipKIT shield connectors	17	Power good LED

Callout	Description	Callout	Description
8	Programming mode jumper (JTAG / microSD)	18	Zynq-7000
9	User tri-color LEDs	19	DDR3L memory
10	User push buttons		

Purchasing Options and Board Variants

The Cora Z7 can be purchased with either a Zynq-7010 or Zynq-7007S loaded. These two Cora Z7 product variants are referred to as the Cora Z7-10 and Cora Z7-07S, respectively. When Digilent documentation describes functionality that is common to both of these variants, they are referred to collectively as the "Cora Z7". When describing something that is only common to a specific variant, the variant will be explicitly called out by its name.

The only difference between the Cora Z7-10 and Cora Z7-07S is the capability of the Zynq part. The Zynq processors both have the same capabilities, but the -10 has about a 1.2 times larger internal FPGA and an additional processor core, as compared to the -07S. The differences between the two variants are summarized below:

Product Variant	**Cora Z7-10**	**Cora Z7-07S**
Zynq Part	XC7Z010-1CLG400C	XC7Z007S-1CLG400C
ARM Processor Cores	2	1
1 MSPS On-chip ADC	Yes	Yes
Look-up Tables (LUTs)	17,600	14,400

Product Variant	Cora Z7-10	Cora Z7-07S
Flip-Flops	35,200	28,800
DSP Slices	80	66
Block RAM	270 KB	225 KB
Clock Management Tiles	2	2

For more information on purchasing, see the Cora Z7 Product Page.

Note: *Due to the sizes of the FPGAs in the Zynq-7010 and Zynq-7007S, they are not very well suited to be used in SDSoC for embedded vision applications. We recommend people purchase the* Arty Z7-20 *if they are interested in these types of applications.*

Software Support

The Cora Z7 is fully compatible with Xilinx's high-performance Vivado Design Suite. This toolset 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 in tandem, down to a simple bare-metal program that controls some LEDs. It is also possible to treat the Zynq AP SoC as a standalone FPGA for those not interested in using the processor in their design. As of Vivado release 2015.4, the Logic Analyzer and High-level Synthesis features of Vivado are free to use for all WebPACK targets, which includes the Cora Z7. The Logic Analyzer assists with debugging logic, and the HLS tool allows you to compile C code directly into HDL.

Master XDC files and Board files for the Cora Z7-10 and Z7-07S are available through the Cora Z7 Resource Center. These files are used to inform Vivado about how the Zynq chip on the Cora is configured and connected to the rest of the Cora.

Zynq platforms are well-suited to be embedded Linux targets, and Cora Z7 is no exception. To help you get started, Digilent provides a Petalinux project that will get you up and running with a Linux system quickly. For more information, see the Cora Z7 Resource Center.

Those familiar with the older Xilinx ISE/EDK toolsets from before Vivado was released can also choose to use the Cora Z7 in that toolset. Digilent does not have many materials to support this, but you can always ask for help on the Digilent Forum.

Functional Description

1 Power Supplies

The Cora Z7 requires a 5 Volt power source to operate. This power source can come from the Digilent USB-JTAG port (J12) or it can be derived from a 5 Volt DC power supply connected to the Power Jack (J15). Unlike other Digilent FPGAs, the Cora Z7 cannot be powered through the Shield Header.

A red power-good LED (LD7), driven by the 3.3V output (VCC3V3) of the DA9062 regulator, indicates that the board is receiving power and that the onboard supplies are functioning as expected. If this LED does not illuminate when an acceptable power supply is connected, please contact your distributor or Digilent Support for further help.

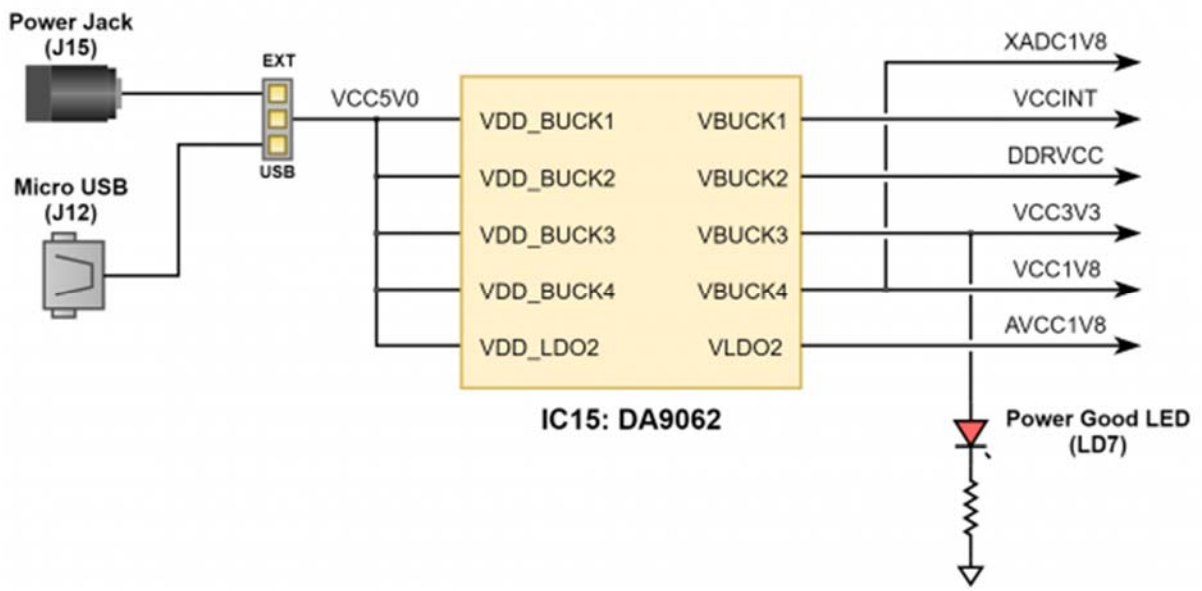


Figure 1.1 Cora Z7 Power Circuit

The USB port can deliver enough power for the vast majority of designs. However, a few demanding applications, including any that drive multiple peripheral boards, might require more power than the USB port can provide. Also, some applications may need to run without being connected to a PC's USB port. In these instances an external power supply can be used by plugging into the Power Jack (J15). The supply must use a coaxial, center-positive 2.1mm (or 2.5mm) internal-diameter plug, and provide a DC voltage of 5 Volts. The supply should provide

a minimum current of 1 amp. Ideally, the supply should be capable of providing 20 Watts of power (5 Volts DC, 4 amps). If the USB port is to be used to deliver power, the Power Select Jumper (JP3) should be set to “USB”. If an external power supply is to be used, JP3 should be set to “EXT” instead.

Voltage regulator circuits from Dialog Semiconductor and ON Semiconductor create the required 3.3V, 1.8V, 1.35V, and 1.00V supplies from the 5V power source. In the event that an external supply or battery pack is used, the on-board Monolithic Power Systems 5V regulator (IC12) provides the 5V source. Table 1.1 provides additional information (typical currents depend strongly on FPGA configuration and the values provided are typical of medium size/speed designs). The 0.675V supply is created by a simple Voltage divider circuit consisting of two 10 KOhm resistors, sourced from the 1.35V rail.

Supply	**Circuits**	**Device**	**Maximum Current**
5.0V	Onboard Regulators, Arduino/chipKit Shield Connector, RGB LEDs	IC4: ON Semiconductor NCP380[1)]	
3.3V	FPGA I/O, USB port, Ethernet	IC15: Dialog Semiconductor DA9062	2A
1.0V	FPGA, Ethernet	IC15: Dialog Semiconductor DA9062	2.5A
1.8V	FPGA Auxiliary, USB port, Ethernet	IC15: Dialog Semiconductor DA9062	1.5A
1.35V	FPGA, DDR3L memory	IC15: Dialog Semiconductor DA9062	2.5A
1.8V	FPGA XADC	IC15: Dialog Semiconductor DA9062	100mA

Table 1.1. Cora Z7 Power Rails.

[1] With JP3 set to “USB”

2 Zynq APSoC Architecture

The Zynq APSoC is divided into two distinct subsystems: The Processing System (PS) and the Programmable Logic (PL). Figure 2.1 shows an overview of the Zynq APSoC architecture, with the PS colored light green and the PL in yellow. Note that the PCIe Gen2 controller and Multi-gigabit transceivers are not available on the Zynq-7010 or Zynq-7007S devices.

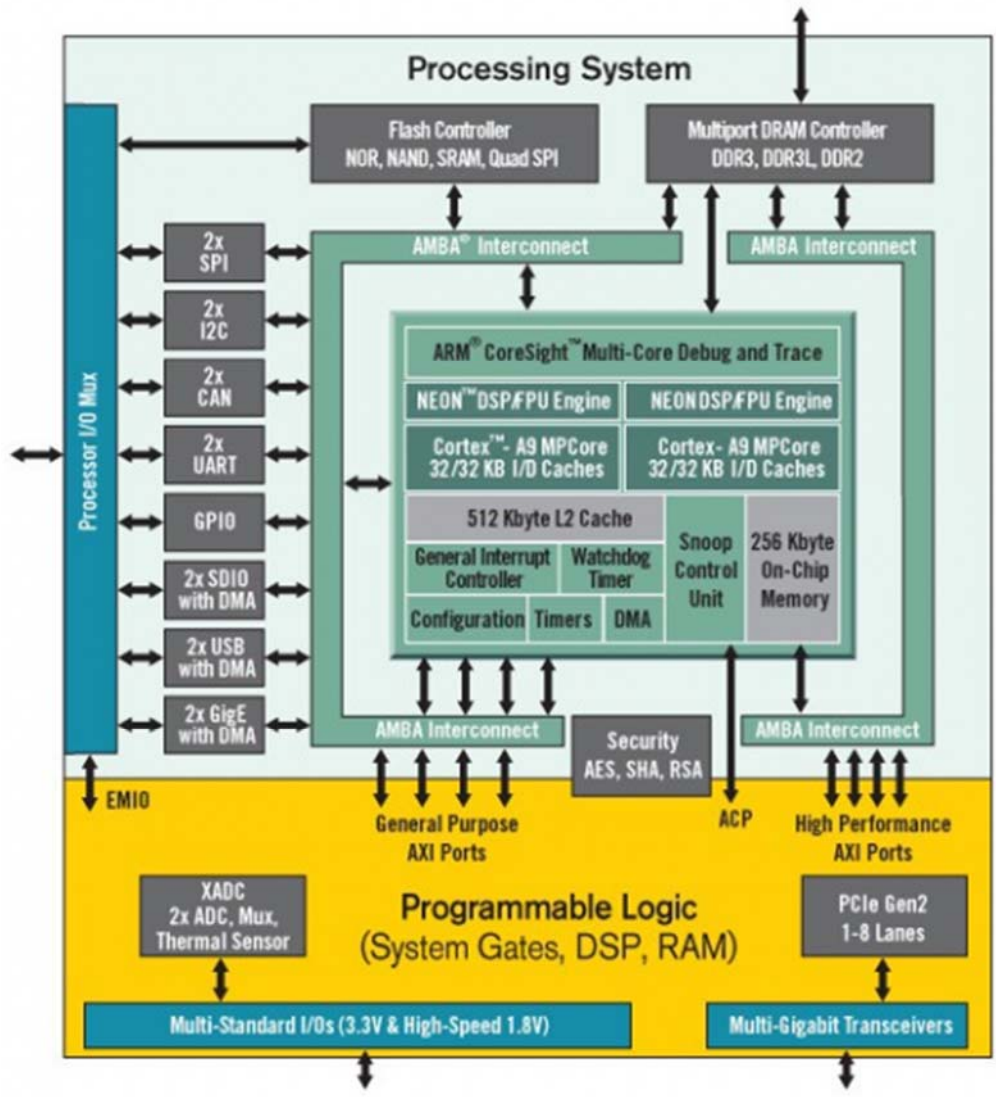


Figure 2.1 Zynq APSoC architecture

The PL is nearly identical to a Xilinx 7-series Artix FPGA, except that it contains several dedicated ports and buses that tightly couple it to the PS. The PL also does not contain the same configuration hardware as a typical 7-series FPGA, and it must be configured either directly by the processor or via the JTAG port.

The PS consists of many components, including the Application Processing Unit (APU), Advanced Microcontroller Bus Architecture (AMBA) Interconnect, DDR3 Memory controller, and various peripheral controllers with their inputs and outputs multiplexed to 54 dedicated pins (called Multiplexed I/O, or MIO pins). The Zynq-7010 APU contains two Cortex-A9 processors, while the Zynq-7007S APU only contains one. Peripheral controllers that do not have their inputs and outputs connected to MIO pins can instead route their I/O through the PL, via the Extended-MIO (EMIO) interface. The peripheral controllers are connected to the processors as slaves via the AMBA interconnect, and contain readable/writable control registers that are addressable in the processors' memory space. The programmable logic is also connected to the interconnect as a slave, and designs can implement multiple cores in the FPGA fabric that each also contain addressable control registers. Furthermore, cores implemented in the PL can trigger interrupts to the processors (connections not shown in Figure 2.1) and perform Direct Memory Access (DMA) transfers to and from DDR3 memory.

There are many aspects of the Zynq APSoC architecture that are beyond the scope of this document. For a complete and thorough description, refer to the Zynq Technical Reference Manual.

Table 2.1 depicts the external components connected to the MIO pins of the Cora Z7. The Zynq Presets File found on the Cora Z7 Resource Centercan be imported into EDK and Vivado Designs to properly configure the PS to work with these peripherals.

MIO 500 (3.3 V)	**Peripherals**				
Pin	**Configuration Mode**	**ENET 0**	**USB 0**	**Shield**	**UART 0**
0 (N/C)					
1 (N/C)					
2	MODE0				
3	MODE1				
4	MODE2				

5	MODE3				
6	MODE4				
7	VCFG0				
8	VCFG1				
9		Ethernet Reset			
10		Ethernet Interrupt			
11			USB Over Current		
12				Shield Reset	
13 (N/C)					
14					UART Input
15					UART Output

MIO 501 (1.0V)	**Peripherals**		
Pin	**ETH 0**	**USB 0**	**SDIO 0**

16	TXCK		
17	TXD0		
18	TXD1		
19	TXD2		
20	TXD3		
21	TXCTL		
22	RXCK		
23	RXD0		
24	RXD1		
25	RXD2		
26	RXD3		
27	RXCTL		
28		DATA4	
29		DIR	

30		STP	
31		NXT	
32		DATA0	
33		DATA1	
34		DATA2	
35		DATA3	
36		CLK	
37		DATA5	
38		DATA6	
39		DATA7	
40			CCLK
41			CMD
42			D0
43			D1

44			D2
45			D3
46		RESETN	
47			CD
48 (N/C)			
49 (N/C)			
50 (N/C)			
51 (N/C)			
52	MDC		
53	MDIO		

Table 2.1. MIO Pinout

---

3 Zynq Configuration

Unlike Xilinx FPGA devices, APSoC devices such as the Zynq-7007S are designed around the processor, which acts as a master to the programmable logic fabric and all other on-chip peripherals in the processing system. This causes the Zynq boot process to be more similar to that of a microcontroller than an FPGA. This process involves the processor loading and executing a Zynq Boot Image, which includes a First Stage Bootloader (FSBL), a bitstream for configuring the programmable logic (optional), and a user application. The boot process is broken into three stages:

Stage 0

After the Cora Z7 is powered on, or the Zynq is reset (in software or by pressing SRST), the processor (CPU0 for the Cora Z7-10) begins executing an internal piece of read-only code called the BootROM. If and only if the Zynq was just powered on, the BootROM will first latch the state of the mode pins into the mode register (the mode pins are attached to JP2 on the Cora Z7). If the BootROM is being executed due to a reset event, then the mode pins are not latched, and the previous state of the mode register is used. This means that the Cora Z7 needs a power cycle to register any change in the programming mode jumper (JP2). Next, the BootROM copies an FSBL from the form of non-volatile memory specified by the mode register to the 256 KB of internal RAM within the APU (called On-Chip Memory, or OCM). The FSBL must be wrapped up in a Zynq Boot Image in order for the BootROM to properly copy it. The last thing the BootROM does is hand off execution to the FSBL in OCM.

Stage 1

During this stage, the FSBL first finishes configuring the PS components, such as the DDR memory controller. Then, if a bitstream is present in the Zynq Boot Image, it is read and used to configure the PL. Finally, the user application is loaded into memory from the Zynq Boot Image, and execution is handed off to it.

Stage 2

The last stage is the execution of the user application that was loaded by the FSBL. This can be any sort of program, from a simple “Hello World” design, to a Second Stage Boot loader used to boot an operating system like Linux. For a more thorough explanation of the boot process, refer to Chapter 6 of the Zynq Technical Reference manual.

The Zynq Boot Image is created using Vivado and Xilinx Software Development Kit (Xilinx SDK). For information on creating this image please refer to the available Xilinx documentation for these tools.

The Cora Z7 supports two different boot modes: microSD and JTAG. The boot mode is selected using the Mode jumper (JP2), which affects the state of the Zynq configuration pins after power-on. Figure 3.1 depicts how the Zynq configuration pins are connected on the Cora Z7.

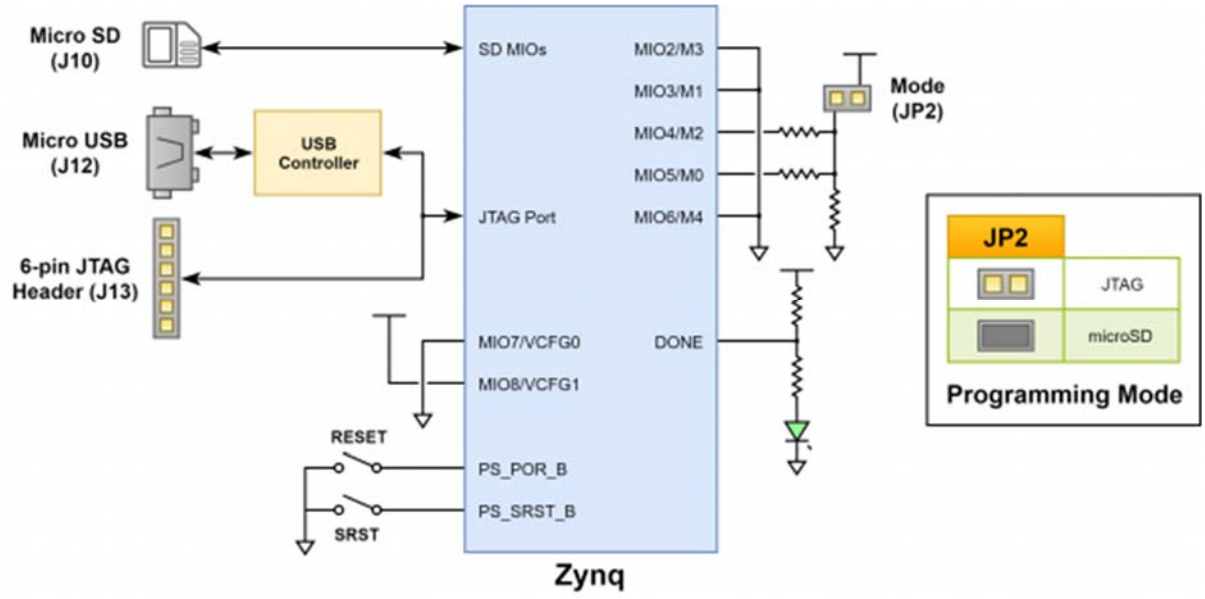


Figure 3.1. Cora Z7 configuration pins.

The two boot modes are described in the following sections.

3.1 microSD Boot Mode

The Cora Z7 supports booting from a microSD card inserted into connector J10. The following procedure will allow the Zynq to boot from microSD with a standard Zynq Boot Image created with the Xilinx tools:

1. Format the microSD card with a FAT32 file system.
2. Copy the Zynq Boot Image created with Xilinx SDK to the microSD card.
3. Rename the Zynq Boot Image on the microSD card to BOOT.bin.
4. Eject the microSD card from the host computer and insert it into connector J10 on the Cora Z7.
5. Attach a power source to the Cora Z7 and select it using JP3.
6. Place a jumper on JP2, shorting the two pins together.
7. Turn the board on. The board will now boot the image on the microSD card.

3.2 JTAG Boot Mode

When placed in JTAG boot mode, the processor 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.

4 DDR3L Memory

The Cora Z7 includes a Micron MT41K256M16HA-125 DDR3L memory component, creating a single rank 16-bit wide interface and a total of 512 MiB (Mebi-byte, or 536,870,912 bytes) of capacity. The DDR3L is connected to the hard memory controller in the Processor Subsystem (PS), as outlined in the Zynq documentation.

The PS incorporates an AXI memory port interface, a DDR controller, the associated PHY, and a dedicated I/O bank. DDR3L memory interface speeds up to 533 MHz/1066 Mbps are supported.

The Cora Z7 was routed with 40 ohm (+/-10%) trace impedance for single-ended signals, and differential clock and strobes set to 80 ohms (+/-10%). A feature called DCI (Digitally Controlled Impedance) is used to match the drive strength and termination impedance of the PS pins to the trace impedance. On the memory side, the DDR3L chip calibrates its on-die termination and drive strength using a 240 ohm resistor on the ZQ pin.

Due to layout reasons, the two data byte groups (DQ[0-7], DQ[8-15]) were swapped. To the same effect, the data bits inside byte groups were swapped as well. These changes are transparent to the user. During the whole design process the Xilinx PCB guidelines were followed.

Both the memory chip and the PS DDR bank are powered from the 1.35V supply. The mid-point reference of 0.675V is created with a simple resistor divider and is available to the Zynq as external reference.

For proper operation it is essential that the PS memory controller is configured properly. Settings range from the actual memory flavor to the board trace delays. For your convenience, the Cora Z7 Vivado board files are available on the Cora Z7 Resource Center and automatically configure the Zynq Processing System IP core with the correct parameters.

For best DDR3L performance, DRAM training is enabled for write leveling, read gate, and read data eye options in the PS Configuration Tool in Xilinx tools. Training is done dynamically by the controller to account for board delays, process variations and thermal drift. Optimum starting values for the training process are the board delays (propagation delays) for certain memory signals.

Board delays are specified for each of the byte groups. These parameters are board-specific and were calculated from the PCB trace length reports. The DQS to CLK Delay and Board Delay values are calculated specific to the Cora Z7 memory interface PCB design.

For more details on memory controller operation, refer to the Xilinx Zynq Technical Reference manual.

---

5 USB UART Bridge (Serial Port)

The Cora Z7 includes an FTDI FT2232HQ USB-UART bridge (attached to connector J12) that lets you use PC applications to communicate with the board using standard COM port commands (or the tty interface in Linux). Drivers are automatically installed in Windows and newer versions of Linux. Serial port data is exchanged with the Zynq using a two-wire serial port (TXD/RXD). After the drivers are installed, I/O commands can be used from the PC directed to the COM port to produce serial data traffic on the Zynq pins. The port is tied to PS (MIO) pins and can be used in combination with the UART 0 controller.

The Zynq presets file (available through the Cora Z7 Resource Center) takes care of mapping the correct MIO pins to the UART 0 controller and uses the following default protocol parameters: 115200 baud rate, 1 stop bit, no parity, 8-bit character length.

Two on-board status LEDs provide visual feedback on traffic flowing through the port: the transmit LED (LD5) and the receive LED (LD4). Signal names that imply direction are from the point-of-view of the DTE (Data Terminal Equipment), in this case the PC.

The FT2232HQ is also used as the controller for the Digilent USB-JTAG circuitry, but the USB-UART and USB-JTAG functions behave entirely independent of one another. Programmers interested in using the UART functionality of the FT2232 within their design do not need to worry about the JTAG circuitry interfering with the UART data transfers, and vice-versa. The combination of these two features into a single device allows the Cora Z7 to be programmed, communicated with via UART, and powered from a computer attached with a single Micro USB cable.

The DTR signal from the UART controller on the FT2232HQ is connected to MIO12 of the Zynq device via JP1. Should the Arduino IDE be ported to work with the Cora Z7, this jumper can be shorted and MIO12 could be used to place the Cora Z7 in a “ready to receive a new sketch” state. This would mimic the behavior of typical Arduino IDE boot-loaders.

---

6 microSD Slot

The Cora Z7 provides a microSD slot (J10) for non-volatile external memory storage as well as for booting the Zynq. The slot is wired to Bank 1/501 MIO[40-47], including the Card Detect signal. On the Zynq PS, peripheral SDIO 0 is mapped out to these pins and controls communication with the SD card. The pinout can be seen in Table 6.1. The peripheral controller supports 1-bit and 4-bit SD transfer modes, but does not support SPI mode. Based on the Zynq Technical Reference manual, SDIO host mode is the only mode supported.

Signal Name	Description	Zynq Pin	SD Slot Pin

SD_D0	Data[0]	MIO42	7
SD_D1	Data[1]	MIO43	8
SD_D2	Data[2]	MIO44	1
SD_D3	Data[3]	MIO45	2
SD_CCLK	Clock	MIO40	5
SD_CMD	Command	MIO41	3
SD_CD	Card Detect	MIO47	9

Table 6.1. microSD pinout

The SD slot is powered from the 3.3V rail, but is connected through MIO Bank 1/501 (1.8V). Therefore, a TI TXS02612 level shifter is used to perform the necessary translation. The TXS02612 is actually a 2-port SDIO port expander, but only its level shifter function is used. The connection diagram can be seen in Figure 6.1. Mapping out the correct pins and configuring the interface is handled by the Cora Z7 Zynq presets file, available through the Cora Z7 Resource Center.

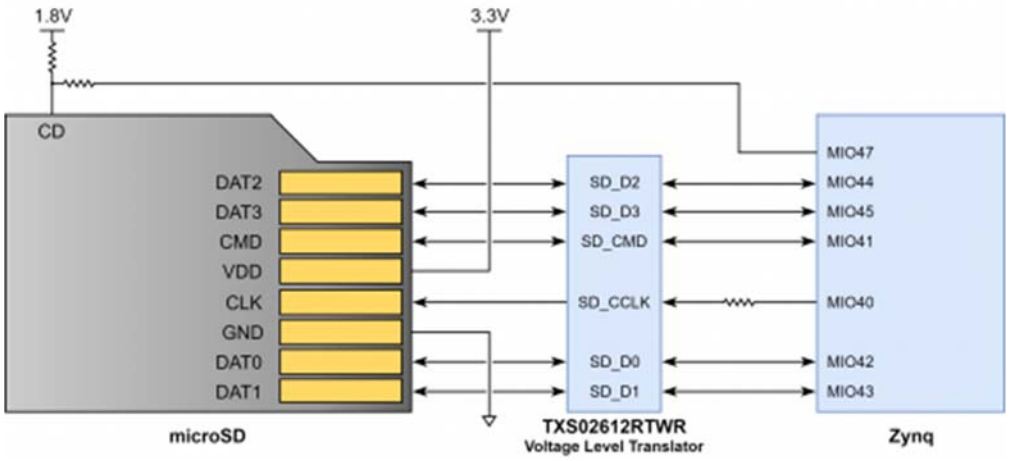


Figure 6.1. microSD slot signals

Both low speed and high speed cards are supported, as the maximum clock frequency is 50 MHz. A Class 4 card or better is recommended.

Refer to section 3.1, microSD Boot Mode, for information on how to boot the Cora Z7 from an SD card. For more information, consult the Zynq Technical Reference manual.

---

7 USB Host

The Cora Z7 implements one of the two available PS USB OTG interfaces on the Zynq device. A Microchip USB3320 USB 2.0 Transceiver Chip with an 8-bit ULPI interface is used as the PHY. The PHY features a complete HS-USB Physical Front-End supporting speeds of up to 480Mbs. The PHY is connected to MIO Bank 1/501, which is powered at 1.8V. The USB0 peripheral is used on the PS, connected through MIO[28-39]. The USB OTG interface is configured to act as an embedded host. USB OTG and USB device modes are not supported.

The Cora Z7 is technically an “embedded host”, because it does not provide the required 150 µF of capacitance on VBUS required to qualify as a general purpose host. It is possible to modify the Cora Z7 so that it complies with the general purpose USB host requirements by loading C35 with a 150 µF capacitor. Only those experienced at soldering small components on PCBs should attempt this rework. Many USB peripheral devices will work just fine without loading C35. Whether the Cora Z7 is configured as an embedded host or a general purpose host, it can provide 1A on the 5V VBUS line.

Note that if your design uses the USB Host port (embedded or general purpose), then the Cora Z7 should be powered via a wall adapter capable of providing more power.

---

8 Ethernet PHY

The Cora Z7 uses a Realtek RTL8211E-VL PHY to implement a 10/100/1000 Ethernet port for network connection. The PHY connects to MIO Bank 501 (1.8V) and interfaces to the Zynq-7000 APSoC via RGMII for data and MDIO for management. The auxiliary interrupt (INTB) and reset (PHYRSTB) signals connect to MIO pins MIO10 and MIO9, respectively.

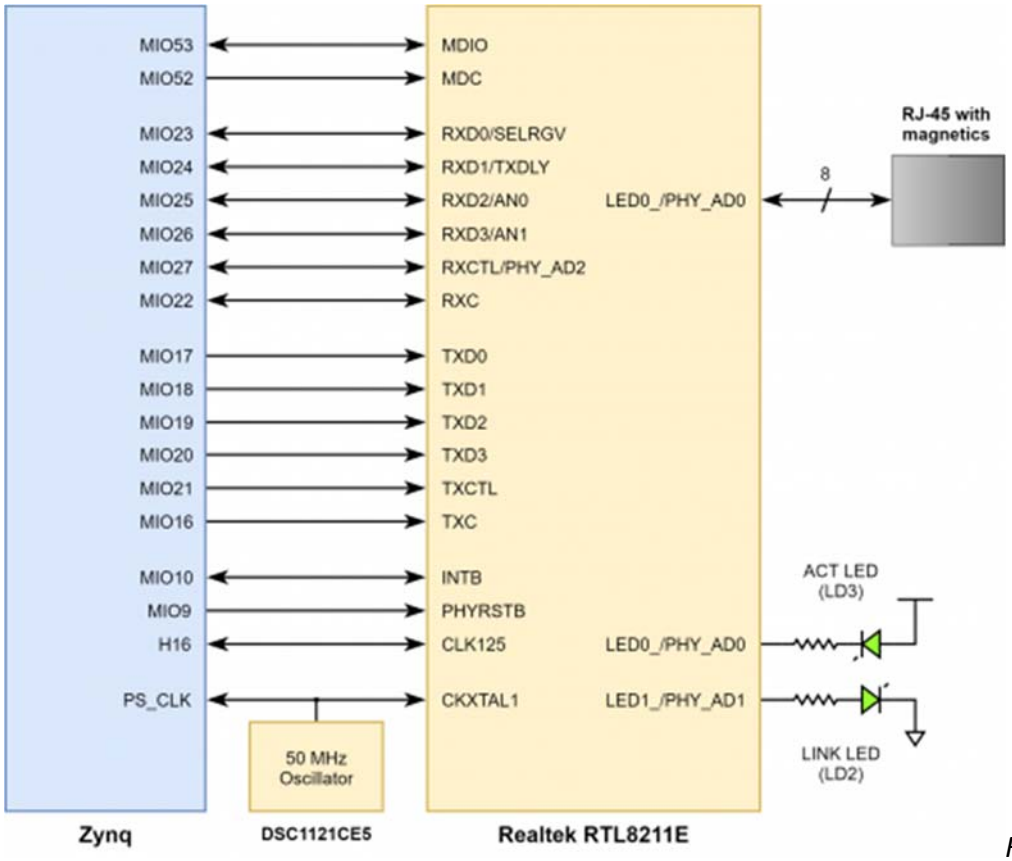


Figure 8.1. Ethernet PHY signals

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 Zynq not configured.

Two status indicator LEDs are on-board near the RJ-45 connector that indicate traffic (LD9) and valid link state (LD8). Table 8.1 shows the default behavior.

Function	Designator	State	Description
LINK	LD8	Steady On	Link 10/100/1000
		Blinking 0.4s ON, 2s OFF	Link, Energy Efficient Ethernet (EEE) mode
ACT	LD9	Blinking	Transmitting or Receiving

Table 8.1. Ethernet status LEDs.

The Zynq incorporates two independent Gigabit Ethernet Controllers. They implement a 10/100/1000 half/full duplex Ethernet MAC. Of these two, GEM 0 can be mapped to the MIO pins where the PHY is connected. Since the MIO bank is powered from 1.8V, the RGMII interface uses 1.8V HSTL Class 1 drivers. For this I/O standard an external reference of 0.9V is provided in bank 501 (PS_MIO_VREF). Mapping out the correct pins and configuring the interface is handled by the Cora Z7 Zynq Presets file, available on the Cora Z7 Resource Center.

Although the default power-up configuration of the PHY might be enough in most applications, the MDIO bus is available for management. The RTL8211E-VL is assigned the 5-bit address 00001 on the MDIO bus. With simple register read and write commands, status information can be read out or configuration changed. The Realtek 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 RTL8211E-VL is capable of inserting a 2ns delay on both the TXC and RXC so that board traces do not need to be made longer.

The PHY is clocked from the same 50 MHz oscillator that clocks the Zynq PS. The parasitic capacitance of the two loads is low enough to be driven from a single source.

On an Ethernet network each node needs a unique MAC address. To this end, a sticker has been applied to the Cora Z7 at the factory, displaying a 48-bit globally unique EUI-48/64™ compatible identifier.

For more information on using the Gigabit Ethernet MAC, refer to the Zynq Technical Reference manual.

---

9 Clock Sources

The Cora Z7 provides a 50 MHz clock to the Zynq PS_CLK input, which is used to generate the clocks for each of the Processing System (PS) subsystems. The 50 MHz input allows the processor to operate at a maximum frequency of 650 MHz and the DDR3 memory controller to operate at a maximum of 525 MHz (1050 Mbps). The Cora Z7 Zynq Preset file available within the Digilent Vivado Board File package (Installation Instructions) can be imported into the Zynq Processing System IP core in a Vivado project to properly configure the Zynq to work with the 50 MHzinput clock.

The PS has a dedicated Phase-Locked Loop (PLL) capable of generating up to four reference clocks, each with settable frequencies, that can be used to clock custom logic implemented in the Programmable Logic (PL). Additionally, the Cora Z7 provides an external 125 MHz reference clock directly to pin H16 of the PL. The external reference clock allows the PL to be used completely independently of the PS, which can be useful for simple applications that do not require the processor.

The PL of the Zynq also includes Mixed-Mode Clock Managers (MMCM) and PLLs that can be used to generate clocks with precise frequencies and phase relationships. Any of the four PS reference clocks or the 125 MHz external reference clock can be used as an input to the MMCMs and PLLs. Both the Cora Z7-07S and Z7-10 include 2 MMCM's and 2 PLL's. For a full description of the capabilities of the Zynq PL clocking resources, refer to the “7 Series FPGAs Clocking Resources User Guide” available from Xilinx.

Figure 9.1 outlines the clocking scheme used on the Cora Z7. Note that the reference clock output from the Ethernet PHY is used as the 125 MHzreference clock to the PL, in order to cut the cost of including a dedicated oscillator for this purpose. Keep in mind that CLK125 will be disabled when the Ethernet PHY (IC1) is held in hardware reset by driving the PHYRSTB signal low.

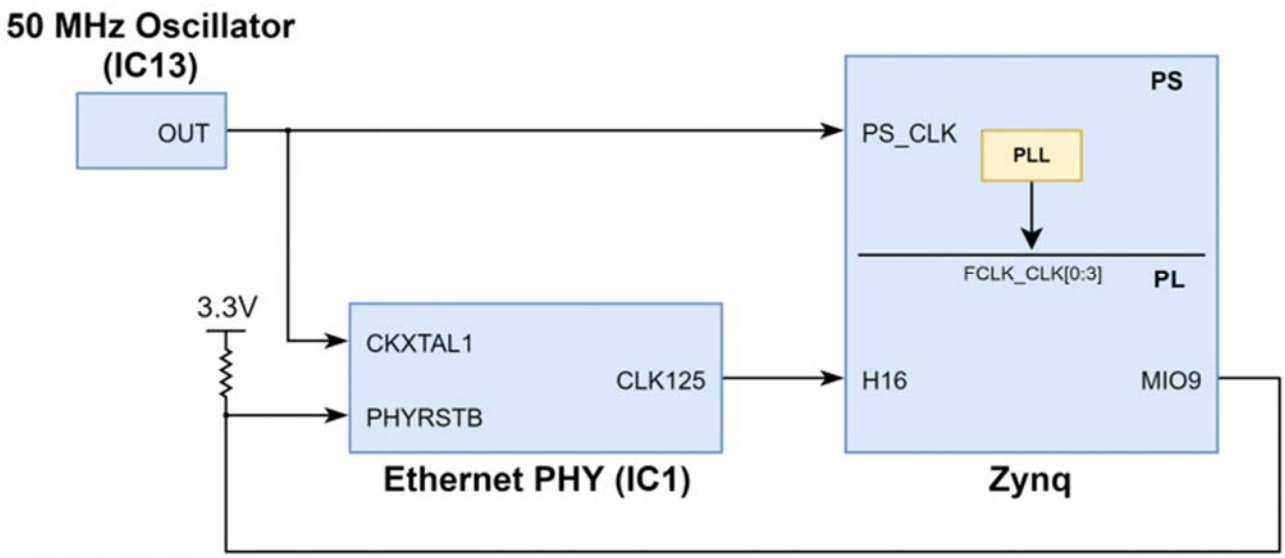


Figure 9.1. Cora Z7 Clocking

---

10 Reset Sources

10.1 Power-on Reset

The Zynq Processing System (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 Cora Z7 drives this signal from the nRESET signal of the DA9062 DC-DC converter system in order to hold the system in reset until all power supplies are valid. A push-button, labeled RESET, can be used to toggle the power-on reset signal.

Note: *The power-on reset will not reset attached shields.*

10.2 Processor Subsystem Reset

The external system reset, labeled SRST, resets the Zynq 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 OCM. The PL is also cleared during a system reset. System reset does not cause the boot mode strapping pins to be re-sampled.

The SRST button also causes the CK_RST signal to toggle and trigger a reset on any attached shields.

---

11 Basic I/O

The Cora Z7 board includes two tri-color LEDs and 2 push buttons as shown in Figure 11.1. The push buttons are connected to the Zynq PL via series resistors to prevent damage from inadvertent short circuits (a short circuit could occur if an FPGA pin assigned to a push button was inadvertently defined as an output). The two 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.

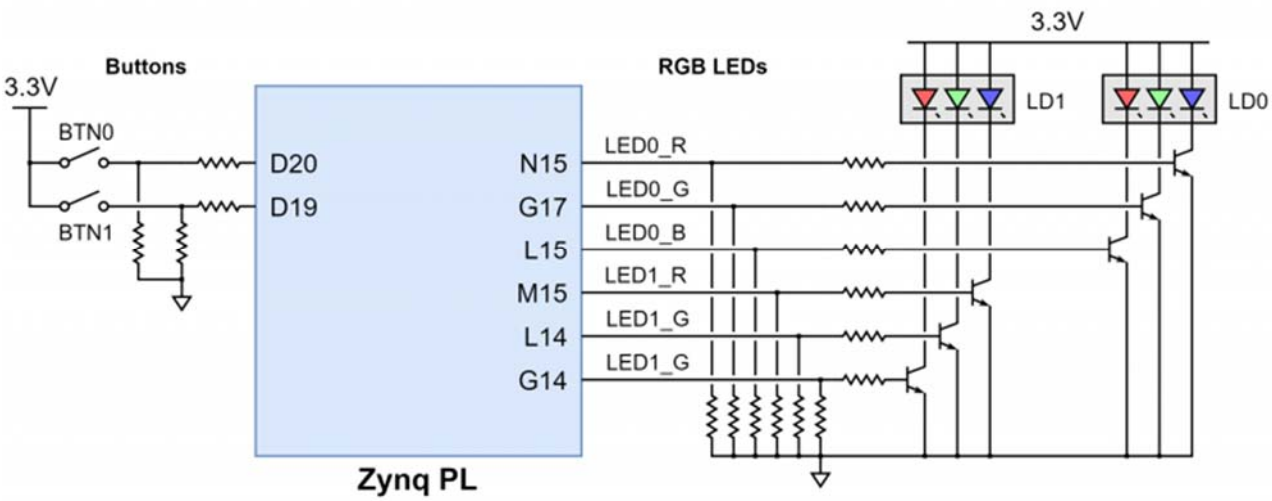


Figure 11.1. Cora Z7 Basic I/O

11.1 Tri-Color LEDs

The Cora Z7 board contains two tri-color LEDs. Each tri-color LED has three input signals that drive the cathodes of three smaller internal LEDs: one red, one blue, and one green. Driving the signal corresponding to one of these colors high will illuminate the internal LED. The input signals are driven by the Zynq PL through a transistor, which inverts the signals. Therefore, to light up the tri-color LED, the corresponding signals need to be driven high. The tri-