1300 Henley Court
Pullman, WA 99163
509.334.6306
www.store.digilent.com

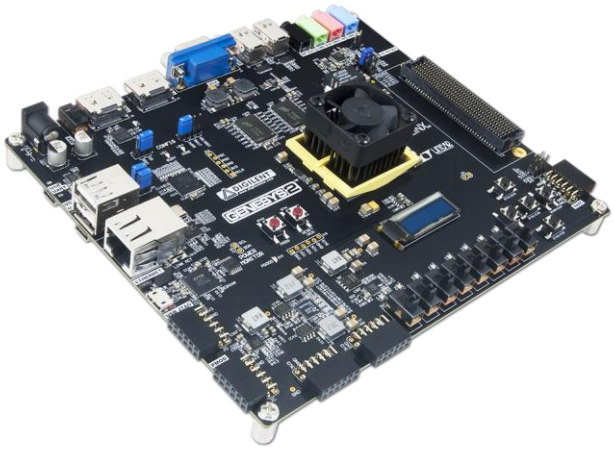
Genesys 2 FPGA Board Reference Manual

Revised March 20, 2017
This manual applies to the Genesys 2 rev. F

Overview

The Digilent Genesys 2 board is an advanced, high-performance, ready-to-use digital circuit development platform based on the latest Kintex-7™ Field Programmable Gate Array (FPGA) from Xilinx. With its high-capacity, high-speed FPGA (Xilinx part number XC7K325T-2FFG900C), fast external memories, high-speed digital video ports, and wide expansions options make the Genesys 2 well suited for data and video processing applications. Several built-in peripherals, including Ethernet, audio and USB 2.0, allow a wide range of other applications. The fully-bonded[1] high-speed FMC HPC connector opens the door to great expansion possibilities.

The Kintex-7 FPGA offers more capacity, higher performance, and more resources than the Virtex-5 from the first-generation Genesys:



The Genesys 2.

- 50,950 logic slices (up 7x), each with four 6-input LUTs and 8 flip-flops
- Close to 16 Mbits of fast block RAM (up 7x)
- Ten clock management tiles, each with phase-locked loop (PLL)
- 840 DSP slices (up 17x)
- Internal clock speeds exceeding 450MHz
- On-chip analog-to-digital converter (XADC)
- Up to 10.3125Gbps gigabit transceivers
- 1800Mbps DDR3 data rate with 32-bit data width
- Commercial -2 speed grade

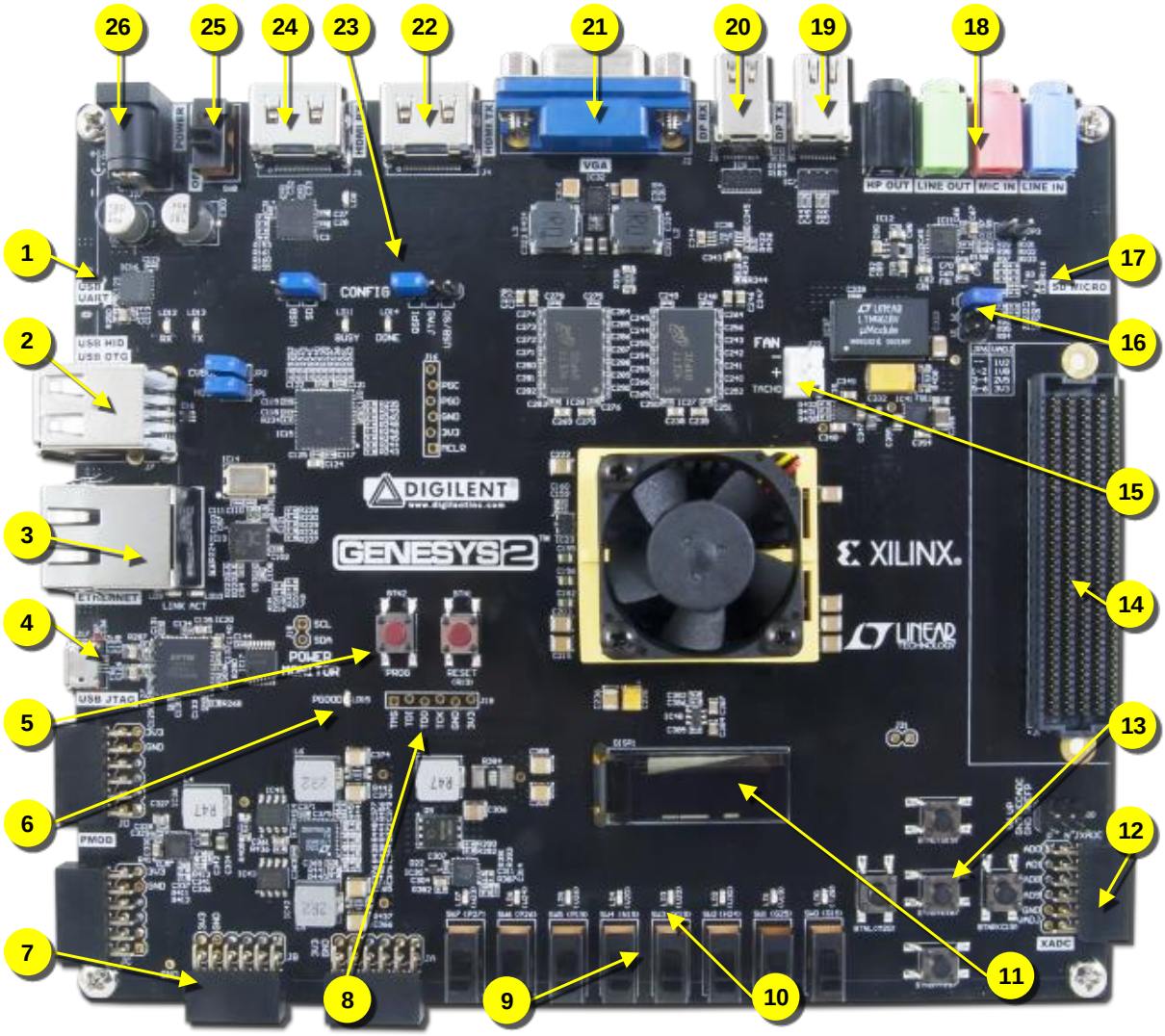
The Genesys 2 also offers an improved collection of ports and peripherals, including:

- Fully bonded[1] 400-pin FMC HPC connector
- USB-UART Bridge
- 8 user switches, 6 buttons
- OLED
- VGA connector
- Pmod for XADC signals
- Two four-lane DisplayPort connectors
- HDMI Sink and HDMI Source
- 10/100/1000 Ethernet PHY
- 1GiB 1800Mt/s on-board DDR3
- USB 2.0 Host/Device/OTG PHY
- Digilent Adept USB port for programming and data
- Ten GTX lanes available in the FMC connector
- MicroSD card connector
- Audio codec w/ four 3.5mm jacks
- Serial Flash
- Five Pmod ports
- USB HID Host for mice, keyboards and USB MSD Host for storage

[1] With the exception of CLK3_BIDIR

The Genesys 2 can be programmed from various sources, like USB thumb drive, microSD, the on-board non-volatile Flash, or the on-board USB-JTAG programmer circuit.

The Genesys 2 is compatible with Xilinx's new high-performance Vivado® Design Suite as well as the ISE toolset. Included in the box is a voucher that unlocks the Design Edition of Vivado that is device-locked to the Genesys 2. This allows designs to be implemented straight out of the box at no additional cost. The Design Edition of Vivado also unlocks the Logic Analyzer tool and still includes the ability to create MicroBlaze™ soft-core processor projects.



Callout	Component Description	Callout	Component Description
1	USB-UART bridge	14	FMC HPC
2	User-USB (bottom) & USB MSD/HID (top)	15	Fan header
3	Ethernet RJ-45 10/100/1000	16	V_{ADJ} jumper
4	USB-JTAG bridge	17	micro SD slot
5	PROG and user reset buttons	18	3.5mm audio jacks
6	Power LED	19	DisplayPort source connector
7	Digital Pmods	20	DisplayPort sink connector

Callout	Component Description	Callout	Component Description
8	JTAG header	21	VGA connector
9	User slide switches	22	HDMI source connector
10	User LEDs	23	FPGA configuration source jumper
11	OLED display	24	HDMI sink connector
12	Dual analog/digital Pmod	25	Power switch
13	User pushbuttons	26	Power jack 12VDC

Table 1. Genesys 2 features and connectors

1 Quick-start

The Genesys 2 comes with an out-of-box demo design that gets loaded from the on-board QSPI flash. It exercises most of the on-board peripherals. Just power the board with the included 12V wall supply, flick the power switch, wait for the design to fully load, and explore the following features:

- Connecting an HDMI/DVI, VGA, or DisplayPort monitor shows a demo image with a mouse pointer.
- Connecting a USB mouse controls the mouse pointer on the display.
- The OLED shows the Digilent logo and various information on several pages. Advance between pages with the BTNC button.
- The internal FPGA temperature, voltage and current readings of various power rails are shown on the OLED.
- Connecting the board to an Ethernet network will acquire link, IP address and become “pingable” at the IPv4 address displayed on the OLED.
- Connecting the USB-UART port to a PC and opening a terminal (115200, 8, N, 1) shows status messages.
- Pushing BTNU records audio off the microphone input for five seconds and plays it back on the headphone output if BTND is pushed, or line-out if BTNL. Similarly, BTNR starts a recording off the line-in jack.
- The LEDs are showing a scanning light bar.
- The fan starts when FPGA internal temperature reaches 60°C and stops when it drops back to 40°C.

To develop new FPGA designs for the Genesys 2, download and install the Xilinx Vivado ® Design Suite[2]. The tools include all the USB drivers for the board. Once installed, the USB JTAG and USB UART ports can be connected to the PC and making the FPGA visible in Vivado Hardware Manager.

Additional resources can be found on the Genesys 2 Resource Center.

2 Power Supplies

The Genesys 2 board can receive power from an external power supply through the center-positive barrel jack (J27). The external supply voltage must be 12 V ±5 %. The Genesys 2 cannot be powered from the USB bus.

[2] http://www.xilinx.com/products/design-tools/vivado.html

All Genesys 2 power supplies can be turned on and off together by a single logic-level power switch (SW8). Power supplies are either enabled/disabled directly by the power switch or indirectly by other supplies upstream. A power-good LED (LD15), driven by the "power good" output of the on-board regulators, indicates that the supplies are turned on and operating normally. An overview of the Genesys 2 power circuit is shown in Figure 1.

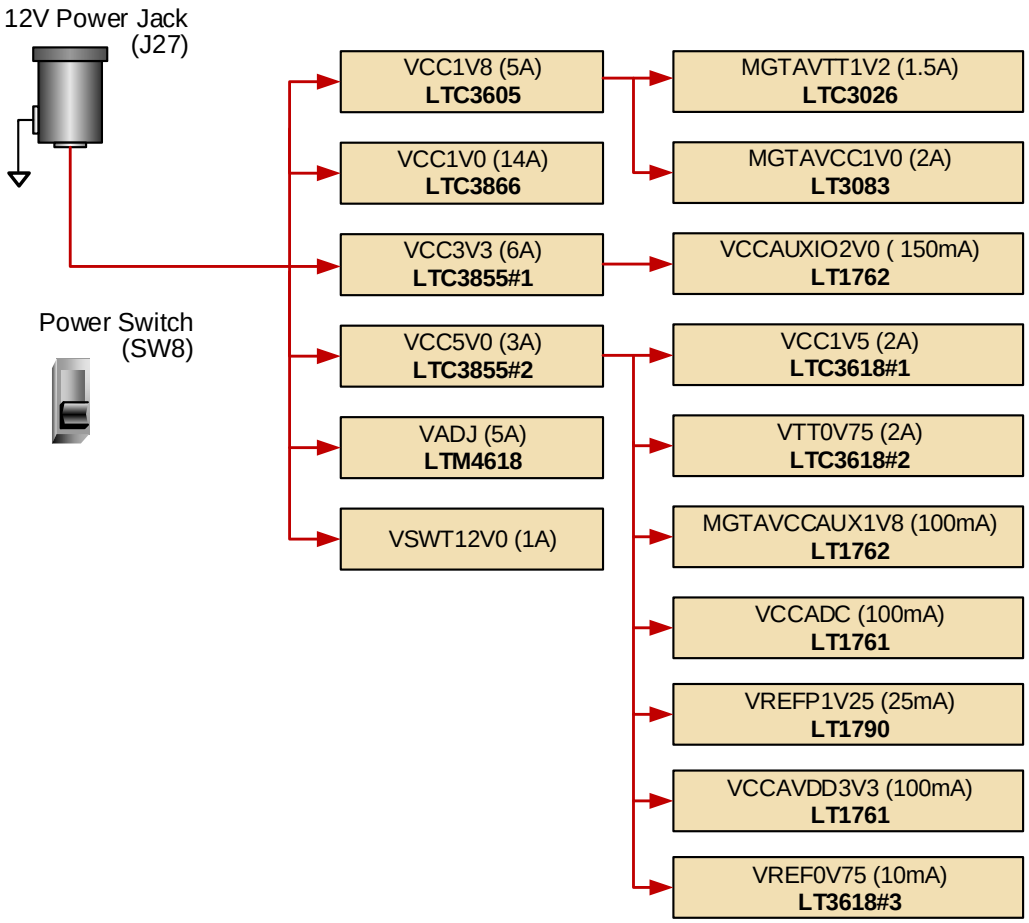


Figure 1. Genesys 2 power circuit

An external power supply can be used by plugging it into the power jack (J27). The supply must use a coax, center-positive 2.1mm internal-diameter plug, and deliver 12VDC ±5 %. The minimum current rating of the supply depends on the actual design implemented in the FPGA, but at least 3A (i.e., at least 36W) is recommended. For high-power FMC applications, a 60W supply is recommended.

Voltage regulator circuits from Linear Technology create the different voltages required by the FPGA and on-board peripherals from the main power input. Some regulators use the outputs of another regulator as input, depending on design considerations. In some cases, this chaining helps in creating the proper power-on sequence for circuits. In other cases, the chaining of power supply enables achieves the same purpose.

Table 2 provides information on maximum and typical currents for each power rail. The typical currents strongly depend on FPGA configuration and the values provided are the current consumption of the OOB demo.

Supply	Circuits	Device	Current (max/typical)
3.3 V	FPGA I/O, USB, FMC, Clocks, Pmod, Ethernet, SD slot, Flash, DisplayPort, HDMI	IC42: LTC3855#1	6 A / 0.8 A
1.0 V	FPGA Core	IC30: LTC3866	14 A / 1.2 A
1.8 V	FPGA Auxiliary	IC36: LTC3605	5 A / 1.6 A
1.5 V	DDR3 and FPGA I/O	IC32: LTC3618	2 A / 0.7 A
0.75V	DDR3 termination, reference	IC32: LTC3618	2 A
2.0 V	FPGA Auxiliary I/O for memory high data rates[3]	IC38: LT1762	150 mA
V_{ADJ} (1.2-3.3 V)	User I/O, FMC and FPGA I/O	IC37: LTM4618	5A
3.3 V	Audio analog supply	IC12: LT1761	100 mA
5.0 V	USB Host, HDMI	IC42: LTC3855#2	3 A / 0.3 A
MGT 1.0 V	Gigabit Transceivers VCC	IC41: LT3083	2 A
MGT 1.2 V	Gigabit Transceivers VTT	IC39: LTC3026	1.5 A
MGT 1.8 V	Gigabit Transceivers AUX	IC40: LT1762	150 mA
XADC 1.8 V	XADC supply	IC47: LT1761	100 mA
XADC 1.25 V	XADC reference	IC48: LT1790	5 mA

Table 2. Voltage rail power ratings.

The V_{ADJ} power rail requires special attention. It is an adjustable rail that powers the FMC mezzanine connector, user push-buttons, switches, XADC Pmod connector and the FPGA banks connected to these peripherals (banks 15, 16, 17). The feedback pin of the V_{ADJ} regulator is connected to a resistor network modifiable by jumper JP6. Changing its position changes the resistor divider in the feedback loop, thereby changing the voltage on the regulator's output. The possible voltages are listed in Figure 2. If JP6 is not set, the V_{ADJ} voltage defaults to 1.2 V. This feature enables setting the V_{ADJ} voltage to suit a certain FMC mezzanine card or application. It is recommended to only change the JP6 position with the power switch in the OFF position.

Please note that for proper voltage levels in digital signals connected to V_{ADJ}-powered FPGA banks (ex. user push-buttons), the correct I/O standard still needs to be set in the design user constraints (XDC or UCF file). See the schematic and/or the constraints file to determine which signals are in V_{ADJ}-powered banks. The provided master UCF and XDC files assume the default V_{ADJ} voltage of 1.2V, declaring LVCMOS12 as the I/O standard for these signals.

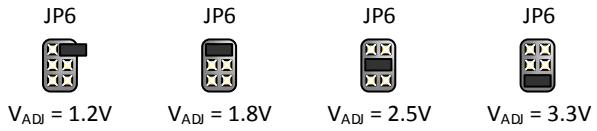


Figure 2. V_{ADJ} programmable voltages.

[3] See the 7 Series FPGAs SelectIO Resources User Guide (ug471) for details.

3 Power monitoring

I²C-interfaced monitoring circuits, INA219 from Texas Instruments, are available on the main power rails. These allow real-time voltage, current, and power readings in the FPGA. Six such circuits share the same I²C bus with different slave addresses. These are summarized in

Table 3, along with recommended configuration values.

Supply rail	I²C device address	Shunt resistor	Configuration register	Calibration register	Designator
VCC1V0	b1000101	5 mΩ	0x0867	0x4000	IC29
VCC1V5	b1001100				IC33
VCC1V8	b1001000				IC35
VADJ	b1000001				IC34
VCC3V3	b1000100				IC44
VCC5V0	b1000000				IC46

Table 3. Power monitoring circuit parameters.

The configuration and calibration registers are volatile, so they need to be initialized after power-up. After initialization is done voltage, current, and power values can be read from dedicated registers. It is recommended to add glitch filters to the I²C master controller to avoid spurious start or stop conditions occurring on the bus. This is especially important when using an external I²C master connected to the J18 header (not mounted by default).

For more information on the INA219, see its datasheet[4].

The principle of operation is measuring bus and shunt voltages using a programmable-gain differential amplifier and an analog-to-digital converter. The schematic for one such circuit is shown in Figure 3. The two analog inputs are connected across a shunt resistor placed in series between the power supply and the load. Current consumed by the load produce a voltage drop across the shunt resistor. This voltage is measured by the INA219 and is used to calculate the current. In addition, the bus voltage is measured on V- with respect to GND and is the voltage on the respective power rail. The voltage and current measurements are used to calculate power consumption. If the INA219 is configured, it will correctly calculate all three parameters. In its default, power-up configuration it provides bus and shunt voltage only, which can be used to calculate current and power in the FPGA.

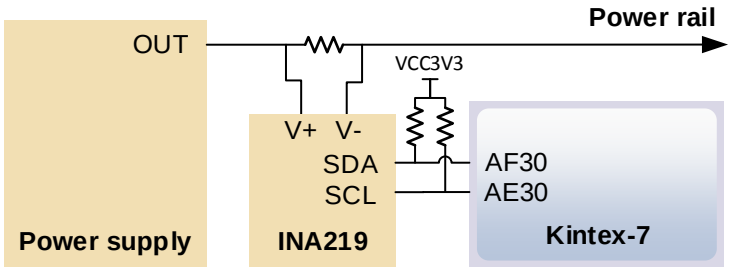


Figure 3. Power monitoring circuit

[4] http://www.ti.com/lit/gpn/ina219

4 Fan

The Genesys 2 comes with a fan and a secondary heat sink pre-installed on the FPGA package heat sink. The fan is powered from the external 12V DC supply rail and controlled by the FPGA. Control is done by the “FAN_EN” signal. Pulling the signal high from the FPGA opens the transistor driving the fan. This pin is pulled high by default. Feedback is obtained on the “FAN_TACH” signal. This generates a pulse with a frequency proportional to the rotation speed of the fan. Each rotation generates four pulses on “FAN_TACH”. The period of these pulses shortens with higher rotation speed and lengthens at slower speeds.

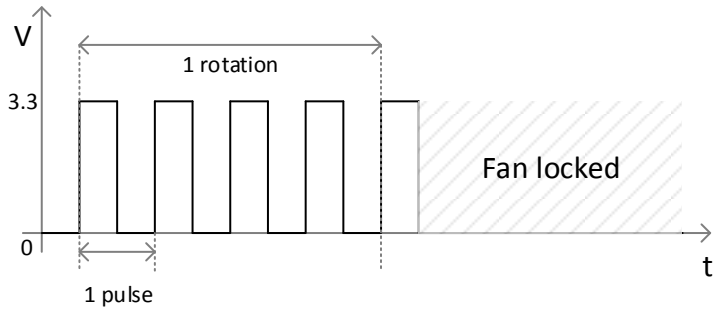


Figure 4. “FAN_TACH” fan speed feedback signal.

The fan uses a 3-pin header for power, ground and feedback. It is recommended leaving the fan connected at all times. Depending on FPGA design complexity and actual usage the fan might not be needed at all. In this case the enable signal can be used to stop the fan, and start it when the FPGA internal temperature (as read by the XADC) gets above a certain limit.

5 FPGA Configuration

After power-on, the Kintex-7 FPGA must be configured (or programmed) before it can perform any functions. You can configure the FPGA in one of four ways:

1. A PC can use the Digilent USB-JTAG circuitry (port J17, labeled “USB JTAG”) to program the FPGA any time the power is on.
2. A file stored in the nonvolatile serial (SPI) flash device can be transferred to the FPGA.
3. A programming file can be transferred to the FPGA from a micro SD card.
4. A programming file can be transferred from a USB mass-storage device (ex. pen drive) attached to the USB HOST port.

Figure 5 shows the different options available for configuring the FPGA. An on-board “mode” jumper (JP5) and a media selection jumper (JP4) select between the programming modes.

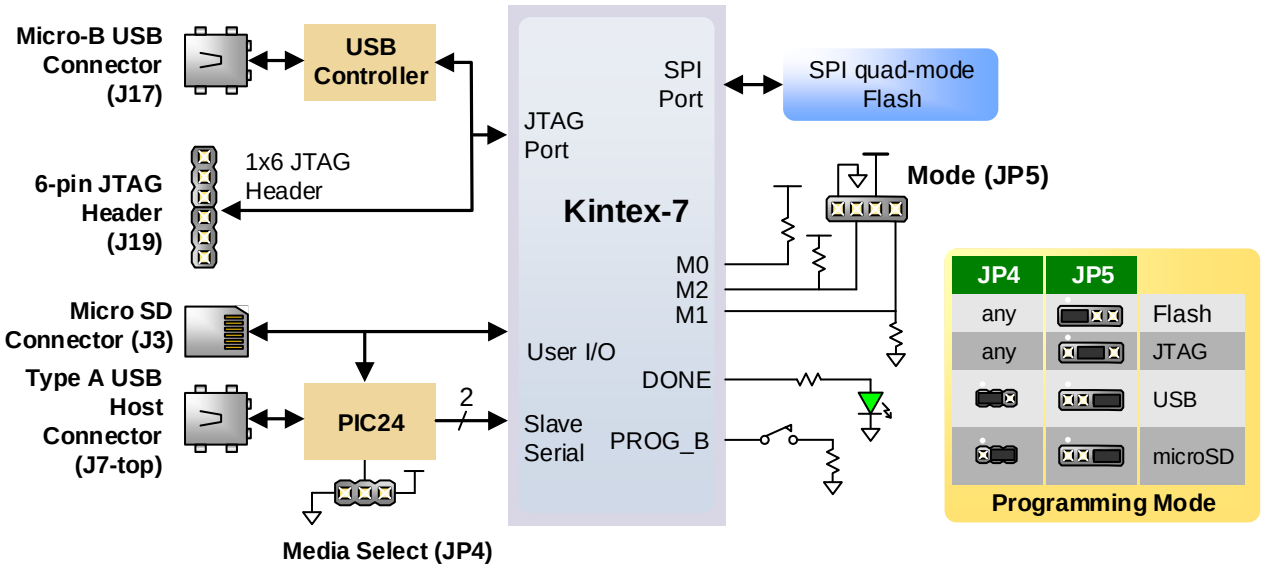


Figure 5. Genesys 2 Configuration Options.

The FPGA configuration data is stored in files called bitstreams that have the .bit file extension. The ISE or Vivado software from Xilinx can create bitstreams from VHDL, Verilog, or schematic-based source files (in the ISE toolset, EDK is used for MicroBlaze embedded processor-based designs).

Bitstreams are stored in volatile SRAM-based memory cells within the FPGA. This data defines the FPGA's logic functions and circuit connections, and it remains valid until it is erased by removing board power, by pressing the reset button attached to the PROG input, or by writing a new configuration file using the JTAG port.

A Kintex-7 325T bitstream is typically 91,548,896 bits long and can take a long time to transfer depending on the programming mode. The time it takes to program the Genesys 2 can be decreased by compressing the bitstream before programming, and then allowing the FPGA to decompress the bitstream itself during configuration. Depending on design complexity, compression ratios of 10x can be achieved. Bitstream compression can be enabled within the Xilinx tools (ISE or Vivado) to occur during generation. For instructions on how to do this, consult the Xilinx documentation for the toolset being used. This option is available for all programming modes.

Mode-specific speed-ups are also available. JTAG clock frequency can be set to the maximum supported by the programming cable in iMPACT/Vivado Hardware Manager. Similarly, the clock frequency for the SPI Flash can be increased in device properties (Vivado) or bitstream generation options (ISE). The micro-SD and USB mass-storage device configuration modes already operate at their maximum possible speed.

After being successfully programmed, the FPGA will cause the "DONE" LED (LD14) to illuminate. Pressing the "PROG" button (BTN2) at any time will reset the configuration memory in the FPGA. After being reset, the FPGA will immediately attempt to reprogram itself from whatever method has been selected by the programming mode jumpers.

The following sections provide greater detail about programming the Genesys 2 using the different methods available.

5.1 JTAG Programming

The Xilinx tools typically communicate with FPGAs using the Test Access Port and Boundary-Scan Architecture, commonly referred to as JTAG. During JTAG programming, a .bit file is transferred from the PC to the FPGA using

the onboard Digilent USB-JTAG circuitry (port J17) or an external JTAG programmer, such as the Digilent JTAG HS2, attached to port J19. You can perform JTAG programming at any time after the Genesys 2 has been powered on, regardless of what the mode jumper (JP5) is set to. If the FPGA is already configured, then the existing configuration is overwritten with the bitstream being transmitted over JTAG. Setting the mode jumper to the JTAG setting is useful to prevent the FPGA from being configured from any other bitstream source until a JTAG programming occurs.

Programming the Genesys 2 with an uncompressed bitstream using the on-board USB_JTAG circuitry usually takes around four seconds with a 30MHz JTAG clock.

JTAG programming can be done using the hardware server in Vivado or the iMPACT tool included with ISE and the Labtools version of Vivado.

The demonstration project[5] available on the Genesys 2 Resource Center[6] provides an in-depth tutorial on how to program your board.

5.2 Quad-SPI Programming

For the FPGA to be able to configure itself from the SPI Flash, it first needs to be programmed with the bitstream. This is called indirect programming and is a two-step process controlled by Hardware Manager (Vivado) or iMPACT (ISE). First, the FPGA is programmed with a design that can program flash devices, and then data is transferred to the flash device via the FPGA circuit (this complexity is hidden from the user by the Xilinx tools). After the flash device has been programmed, it can automatically configure the FPGA at a subsequent power-on or reset event as determined by the mode jumper setting. Programming files stored in the flash device will remain until they are overwritten, regardless of power-cycle events.

Programming the flash can take as long as four to five minutes, which is mostly due to the lengthy erase process inherent to the memory technology. Once written however, FPGA configuration can be very fast, less than a second. Bitstream compression, SPI bus width, and configuration rate are factors controlled by the Xilinx tools that can affect configuration speed. The on-board flash has a Quad-SPI interface, which supports single (x1), dual (x2) and quad (x4) modes. The quad mode results in the fastest possible data transfer rate. For it to work, the bitstream needs to be generated with the x4 bus width option (Vivado device property) and the non-volatile quad configuration bit in the flash to be enabled. The Genesys 2 is shipped with this bit enabled.

Indirect programming of the flash can be done using the iMPACT tool included with ISE or Hardware Manager of Vivado. The correct part to be set in the tools is s25fl256xxxxxx0 from the manufacturer Spansion.

5.3 USB Host and Micro SD Programming

You can program the FPGA from a pen drive attached to the USB-Host port (J7-top row) or a microSD card inserted into J3 by doing the following:

1. Format the storage device (pen drive or microSD card) with a FAT32 file system.
2. Place a single .bit configuration file in the root directory of the storage device.
3. Attach the storage device to the Genesys 2.
4. Set the JP5 Programming Mode jumper on the Genesys 2 to “USB/SD”.
5. Select the desired storage device using JP4.

[5] https://reference.digilentinc.com/learn/programmable-logic/tutorials/genesys-2-programming-guide/start
[6] https://reference.digilentinc.com/reference/programmable-logic/genesys-2/start

6. Push the PROG button or power-cycle the Genesys 2.

The FPGA will automatically configure with the .bit file on the selected storage device. Any .bit files that are not built for the proper Kintex-7 device will be rejected by the FPGA.

The Auxiliary Function Status or "BUSY" LED (LD11) gives visual feedback on the state of the configuration process when the FPGA is not yet programmed:

- When steadily lit the auxiliary microcontroller is either booting up or currently reading the configuration medium (microSD or pen drive) and downloading a bitstream to the FPGA.
- A slow pulse means the microcontroller is waiting for a configuration medium to be plugged in.
- In case of an error during configuration the LED will blink rapidly. It could be that the device plugged in is not getting recognized, it is not properly formatted or the bitstream is not compatible with the FPGA.

When the FPGA has been successfully configured, the behavior of the LED is application-specific. For example, if a USB keyboard or mouse is plugged in, a rapid short blink will signal the receipt of an HID input report from peripheral.

6 Memory

The Genesys 2 board contains two external memories: a 1GiByte volatile DDR3 memory and a 32MiByte non-volatile serial Flash device. The DDR3 uses two 16-bit wide memory component with industry-standard interface soldered on the board resulting in a 32-bit data bus. The serial Flash is on a dedicated quad-mode (x4) SPI bus.

6.1 DDR3

The Genesys 2 includes two Micron MT41J256M16HA-107 DDR3 memory component creating a single rank, 32-bit wide interface. It is routed to a 1.5V-powered HP (High Performance) FPGA bank with 40 ohm controlled single-ended trace impedance. For data signals 40 ohm DCI terminations in the FPGA are used to match the trace characteristics. Similarly, on the memory side on-die terminations (ODT) are used for impedance matching. Address/Control signals are terminated using discrete resistors.

The highest data rate supported is 1800Mbps.

For proper operation of the memory a memory controller and physical layer (PHY) interface needs to be included in the FPGA design. The Xilinx 7-series memory interface solutions core generated by the MIG (Memory Interface Generator) Wizard hides away the complexities of a DDR3 interface. Depending on the tool used (ISE, EDK or Vivado) the MIG Wizard can generate a native FIFO-style or an AXI4 interface to connect to user logic. This workflow allows the customization of several DDR3 parameters optimized for the particular application.

Table 4 below lists the MIG Wizard settings optimized for the Genesys 2.

Setting	Value
Memory type	DDR3 SDRAM
Max. clock period	1112ps (~900MHz)
Max. data rate	~1800Mbps
Clock ratio	4:1
VCCAUX_IO	2.0V
Memory type	Components
Memory part	MT41J256M16XX-107

Setting	Value
Memory voltage	1.5V
Data width	32
Data mask	Enabled
Input clock period	5004ps (~200MHz)
Output driver impedance	RZQ/7
Chip Select pin	Enabled
R_{tt} (nominal) – On-die termination	RZQ/6
Internal V_{ref}	Disabled
Reference clock	Use system clock
Internal termination impedance	N/A
DCI cascade	Disabled

Table 4. DDR3 Settings for the Genesys 2.

The MIG Wizard will require the fixed pin-out of the memory signals to be entered and validated before generating the IP core. For your convenience an importable UCF/XDC file is provided on the Digilent website to speed up the process.

For more details on the Xilinx memory interface solutions refer to the 7 Series FPGAs Memory Interface Solutions User Guide (ug586)[7].

6.2 Quad-SPI Flash

Non-volatile storage is provided by a Spansion S25FL256S flash memory. FPGA configuration files can be written to it, and mode settings are available to cause the FPGA to automatically read a configuration from this device at power on. A Kintex-7 325T configuration file requires just over 10 MiB (mebibyte) of memory, leaving about 70% of the flash device available for user data. Or, if the FPGA is getting configured from another source, the whole memory can be used for custom data.

The contents of the memory can be manipulated by issuing certain commands on the SPI bus. The implementation of this protocol is outside the scope of this document. All signals in the SPI bus except SCK are general-purpose user I/O pins after FPGA configuration. SCK is an exception because it remains a dedicated pin even after configuration. Access to this pin is provided through a special FPGA primitive called STARTUPE2. The AXI Quad SPI IP core is recommended for easy access to the Flash memory.

NOTE: Refer to the manufacturer's data sheets[8] and Xilinx user guides[9] for more information.

[7] http://www.xilinx.com/support/documentation/ip_documentation/mig_7series/v2_1/ug586_7Series_MIS.pdf
[8] http://www.spansion.com/Support/Datasheets/S25FL128S_256S_00.pdf
[9] http://www.xilinx.com/support/documentation/user_guides/ug470_7Series_Config.pdf

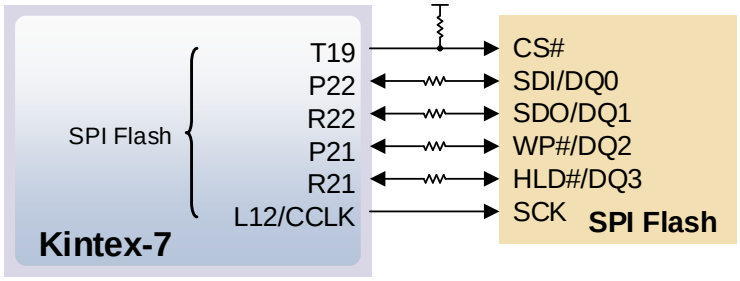


Figure 6. Genesys 2 SPI Flash pin-out.

7 Ethernet PHY

The Genesys 2 board includes a Realtek RTL8211E-VL PHY paired with an RJ-45 Ethernet jack with integrated magnetics to implement a 10/100/1000 Ethernet port for network connection. The PHY interfaces with the FPGA via RGMII for data and MDIO for management. Bank 33 powered at 1.5V is populated with these signals. The auxiliary interrupt (INTB), power management (PMEB) signals are wired to bank 32 and powered at 1.8V. Both of these signals are open-drain outputs from the PHY and need internal pull-ups enabled in the FPGA, if they are used. The reset signal (PHYRSTB) is wired to bank 12, powered at 3.3V. The connection diagram can be seen on Figure 7.

At power-on reset, the PHY is set to the following defaults using the configuration pins in parenthesis:

- Auto-negotiation enabled, advertising all 10/100/1000 modes (AN[1:0])
- PHY address=00001 (PHY_AD[2:0])
- No delay for TXD and RXD relative to TXC and RXC for data latching (RXDLY, TXDLY)

If an Ethernet cable is plugged in, establishing link is attempted straight after power-up, even if the FPGA is not programmed.

Two status indicator LEDs are on-board near the RJ-45 connector that indicate traffic (LD10) and valid link state (LD9). The table below shows the default behavior.

Function	Designator	State	Description
ACT	LD10	Blinking	Transmitting or receiving
LINK	LD9	On	Link up
		Blinking 0.4s ON, 2s OFF	Link up, Energy Efficient Ethernet (EEE) mode

Table 5. Ethernet status LEDs.

The on-board PHY implements Layer 1 in the Ethernet stack, interfacing between the physical copper medium and the media access control (MAC). The MAC must be implemented in the FPGA and mapped to the PHY's RGMII interface. Vivado-based design can use the Xilinx AXI Ethernet Subsystem IP core to implement the MAC and wire it to the processor and the memory subsystem. At the time of writing the IP core needed to be licensed separately.

On an Ethernet network each node needs a unique MAC address. To this end the Genesys 2 comes with a MAC address pre-programmed in a special one-time programmable region (OTP Region 1) of the Quad-SPI Flash[8]. This

unique identifier can be read with the OTP Read command (0x4B). The out-of-box Ethernet demo uses the unique MAC to allow connecting several Genesys 2 boards to the same network.

A downloadable demonstration project can be found on the Genesys 2 Resource Center.

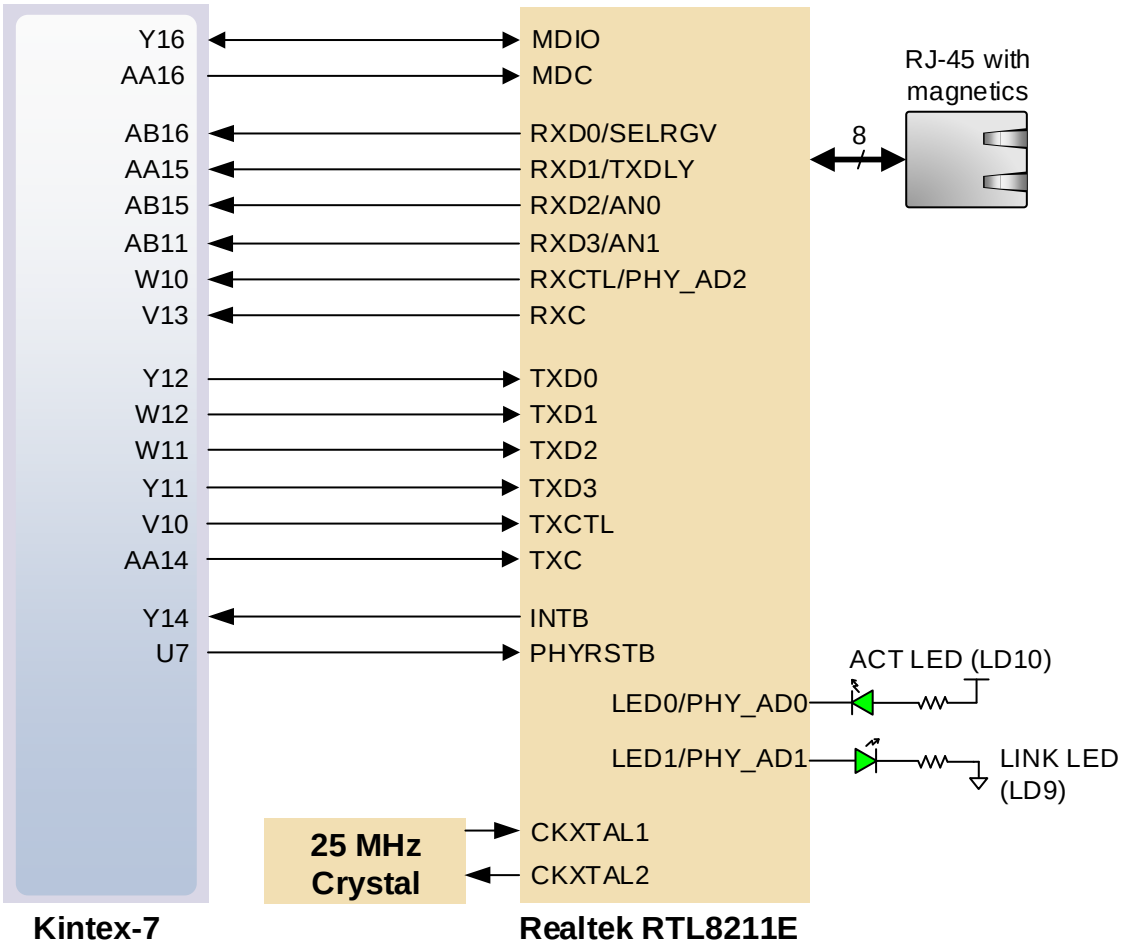


Figure 7. Pin connections between the FPGA and the Ethernet PHY.

8 Oscillators/Clocks

The Genesys 2 board includes several oscillators and crystals, of which two are connected to the FPGA. One differential LVDS 200MHz oscillator is connected to MRCC GPIO pins AD12/AD11 in bank 33. This input clock can drive MMCMs or PLLs to generate clocks of various frequencies and with known phase relationships that may be needed throughout a design. Some rules restrict which MMCMs and PLLs may be driven by the 200MHz input clock. For a full description of these rules and of the capabilities of the Kintex-7 clocking resources, refer to the "7 Series FPGAs Clocking Resources User Guide" (ug472[10]) available from Xilinx.

Xilinx offers the Clocking Wizard IP core to help users generate the different clocks required for a specific design. This wizard will properly instantiate the needed MMCMs and PLLs based on the desired frequencies and phase relationships specified by the user. The wizard will then output an easy to use wrapper component around these

[10] http://www.xilinx.com/support/documentation/user_guides/ug472_7Series_Clocking.pdf

clocking resources that can be inserted into the user's design. The clocking wizard can be accessed from within the Vivado Block Design or Core Generator tools.

The second oscillator outputs a differential LVDS 135MHz clock which enters the FPGA on MGTREFCLK pins. This connects to clock primitives dedicated to the gigabit transceivers and is used for DisplayPort designs.

9 USB UART Bridge (Serial Port)

The Genesys 2 includes an FTDI FT232R USB-UART bridge (attached to connector J15) that lets you use PC applications to communicate with the board using standard Windows COM port commands. Free USB-COM port drivers, available from Windows Update or www.ftdichip.com under the "Virtual Com Port" or VCP heading, convert USB packets to UART/serial port data. Serial port data is exchanged with the FPGA using a two-wire serial port (TXD/RXD) with no handshake signals. 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 Y20 and Y23 FPGA pins.

Two on-board status LEDs provide visual feedback on traffic flowing through the port: the transmit LED (LD13) and the receive LED (LD12). **Signal names that imply direction are from the point-of-view of the DTE (Data Terminal Equipment), in this case the PC.**

The connections between the FT232R and the FPGA are shown in Figure 8.

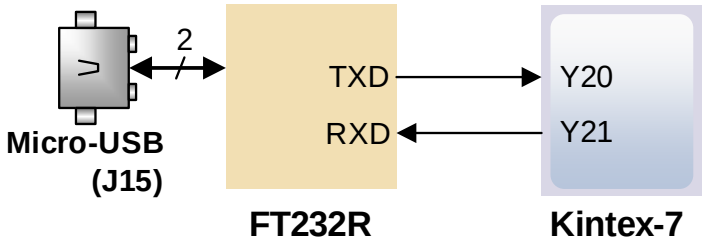


Figure 8. Genesys 2 FT232R connections.

10 PC – FPGA Data Transfer (DPTI / DSPI)

The Genesys 2 provides two interface types that can be used to transfer user data between a PC and an FPGA design. Both of the interfaces have a software component, a Digilent Adept API and a physical interface between the FPGA and the USB controller. Calling API functions on the PC will either present or request data on the FPGA pins according to the chosen protocol. The functionality is implemented using the on-board dual-port FT2232 USB controller. One port is used exclusively for JTAG, while the other either DPTI or DSPI. Since the interfaces share pins, DPTI and DSPI cannot be used simultaneously.

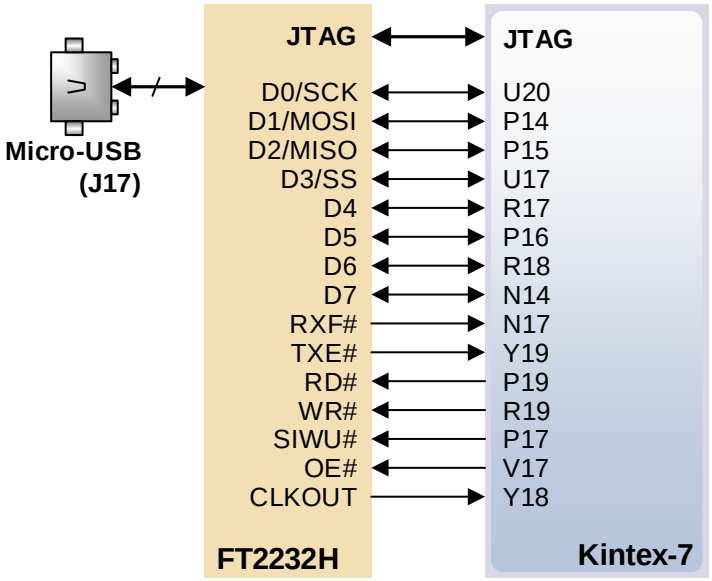


Figure 9. USB-FPGA interfaces provided by the USB JTAG port.

10.1 Parallel Transfer Interface (DPTI)

DPTI is an 8-bit wide parallel FIFO-style data interface. It offers higher bandwidth than DSPI. In FTDI-terminology, DPTI is equivalent to "FT245-style Asynchronous or Synchronous FIFO Interface". It is available in both synchronous and asynchronous modes, configurable from the DPTI API. In synchronous mode, data transfer is timed by the clock provided by the USB controller that is input to the FPGA. In asynchronous mode data transfer is happening on transitions of read and write control signals. The USB controller emulates a FIFO memory, providing status signals about the availability of data to be read or free space for data to be written. The FPGA controls data transfer by read, write and output enable signals.

Signal	Direction (FPGA)	Description
D[7:0]	I/O	Data bus.
RXF#	Input	When low, data is available for reading from the FIFO.
TXE#	Input	When low, data can be written to the FIFO.
RD#	Output	A low-pulse triggers data to be read out from the FIFO.
WR#	Output	A low-pulse triggers data to be written to the FIFO.
SIWU#	Input	Send immediate or Wake-up function. In normal mode a low pulse triggers sending a data packet with the data currently in the FIFO, even if below the normal packet size. In suspend mode a low pulse can wake up the Host PC.
OE#	Output	When low, the data bus is driven by the USB controller (read transfer). When high, the bus is driven by the FPGA (write transfer).
CLKO	Input	60 MHz clock used in synchronous mode. Data is launched and can be captured on the rising edge.

Table 6. DPTI signal description.

For more information, see the FT2232H data sheet[11].

[11] http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT2232H.pdf

10.2 Serial Peripheral Interface (DSPI)

An industry-standard SPI interface can also be used for transferring data. It uses only four signals for serial full-duplex communication. The USB controller acts as a SPI master, with the FPGA taking the slave role. The USB controller initiates a transaction after API function calls and transfers data in both directions simultaneously.

Signal	Direction (FPGA)	Description
SCK	Input	Data bus.
MOSI	Output	When low, data is available for reading from the FIFO.
MISO	Input	When low, data can be written to the FIFO.
SS	Input	A low-pulse triggers data to be read out from the FIFO.

Table 7. DSPI signal description.

For more information, see the FT2232H data sheet[11].

11 USB HID Host

The Auxiliary Function microcontroller (Microchip PIC24FJ128) provides the Genesys 2 with USB HID host capability. After power-up, the microcontroller is in configuration mode, either downloading a bitstream to the FPGA, or waiting on it to be programmed from other sources. Once the FPGA is programmed, the microcontroller switches to application mode, which is USB HID Host in this case. Firmware in the microcontroller can drive a mouse or a keyboard attached to the type A USB connector at J7 labeled "USB HID". J7 is a dual-row USB A receptacle, with the top row connected to the Auxiliary Function microcontroller. Hub support is not currently available, so only a single mouse or a single keyboard can be used. The PIC24 drives several signals into the FPGA – two are used to implement a standard PS/2 interface for communication with a mouse or keyboard, and the others are connected to the FPGA's two-wire serial programming port, so the FPGA can be programmed from a file stored on a USB pen drive or microSD card.

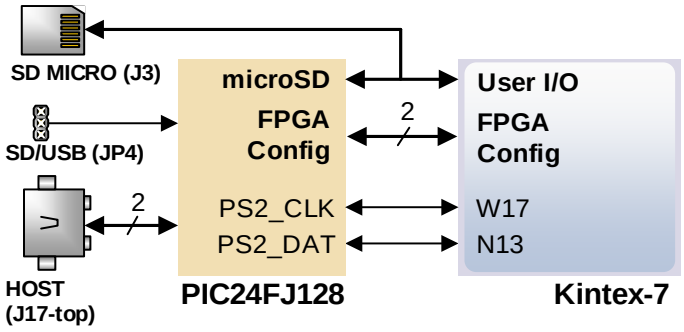
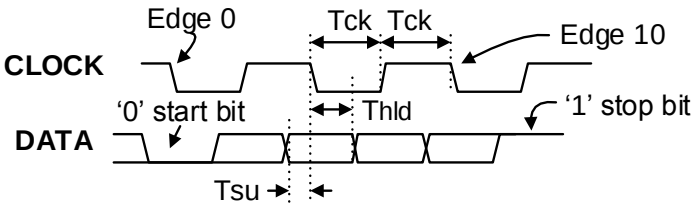


Figure 10. Genesys 2 PIC24 Connections.

11.1 HID Controller

The Auxiliary Function microcontroller hides the USB HID protocol from the FPGA and emulates an old-style PS/2 bus. The microcontroller behaves just like a PS/2 keyboard or mouse would. This means new designs can re-use existing PS/2 IP cores. Mice and keyboards that use the PS/2 protocol use a two-wire serial bus (clock and data) to communicate with a host. On the Genesys 2, the microcontroller emulates a PS/2 device, while the FPGA plays the role of the host. Both the mouse and the keyboard use 11-bit words that include a start bit, data byte (LSB first),

odd parity, and stop bit, but the data packets are organized differently, and the keyboard interface allows bi-directional data transfers (so the host device can illuminate state LEDs on the keyboard). Bus timings are shown in Figure 11.



Symbol	Parameter	Min	Max
T_{CK}	Clock time	30us	50us
T_{SU}	Data-to-clock setup time	5us	25us
T_{HLD}	Clock-to-data hold time	5us	25us

Figure 11. PS/2 Device-to-Host Timing Diagram.

The clock and data signals are only driven when data transfers occur; otherwise they are held in the idle state at logic '1.' This requires that when the PS/2 signals are used in a design, internal pull-ups must be enabled in the FPGA on the data and clock pins. The clock signal is normally driven by the device, but may be held low by the host in special cases. The timings define signal requirements for mouse-to-host communications and bi-directional keyboard communications. A PS/2 interface circuit can be implemented in the FPGA to create a keyboard or mouse interface.

When a keyboard or mouse is connected to the Genesys 2, a "self-test passed" command (0xAA) is sent to the host. After this, commands may be issued to the device. Since both the keyboard and the mouse use the same PS/2 port, one can tell the type of device connected using the device ID. This ID can be read by issuing a Read ID command (0xF2). Also, a mouse sends its ID (0x00) right after the "self-test passed" command, which distinguishes it from a keyboard.

11.2 Keyboard

PS/2 uses open-collector drivers so the keyboard or an attached host can drive the two-wire bus (if the host will not send data to the keyboard, then the host can use input-only ports).

PS/2-style keyboards use scan codes to communicate key press data. Each key is assigned a code that is sent whenever the key is pressed. If the key is held down, the scan code will be sent repeatedly about once every 100ms. When a key is released, an F0 key-up code is sent, followed by the scan code of the released key. If a key can be shifted to produce a new character (like a capital letter), then a shift character is sent in addition to the scan code, and the host must determine which ASCII character to use. Some keys, called extended keys, send an E0 ahead of the scan code (and they may send more than one scan code). When an extended key is released, an E0 F0 key-up code is sent, followed by the scan code. Scan codes for most keys are shown in Figure 12.

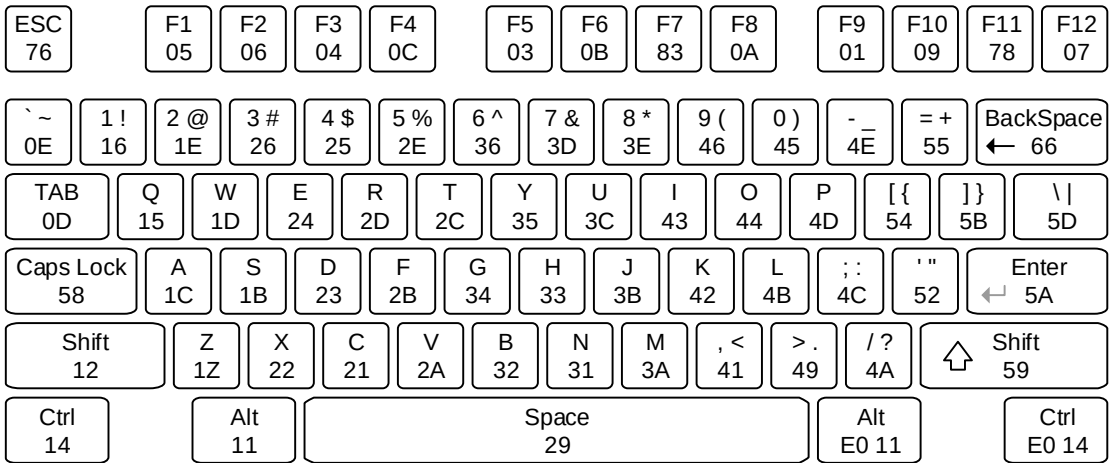


Figure 12. Keyboard scan codes.

A host device can also send data to the keyboard.

Table 8 shows a list of some common commands a host might send.

Command	Action
ED	Set Num Lock, Caps Lock, and Scroll Lock LEDs. Keyboard returns FA after receiving ED, then host sends a byte to set LED status: bit 0 sets Scroll Lock, bit 1 sets Num Lock, and bit 2 sets Caps lock. Bits 3 to 7 are ignored.
EE	Echo (test). Keyboard returns EE after receiving EE
F3	Set scan code repeat rate. Keyboard returns F3 on receiving FA, then host sends second byte to set the repeat rate.
FE	Resend. FE directs keyboard to re-send most recent scan code.
FF	Reset. Resets the keyboard.

Table 8. Keyboard commands.

The keyboard can send data to the host only when both the data and clock lines are high (or idle). Because the host is the bus master, the keyboard must check to see whether the host is sending data before driving the bus. To facilitate this, the clock line is used as a “clear to send” signal. If the host drives the clock line low, the keyboard must not send any data until the clock is released. The keyboard sends data to the host in 11-bit words that contain a ‘0’ start bit, followed by 8-bits of scan code (LSB first), followed by an odd parity bit and terminated with a ‘1’ stop bit. The keyboard generates 11 clock transitions (at 20 to 30 kHz) when the data is sent, and data is valid on the falling edge of the clock.

11.3 Mouse

Once entered in stream mode and data reporting enabled the mouse outputs a clock and data signal when it is moved: otherwise, these signals remain at logic ‘1.’ Each time the mouse is moved, three 11-bit words are sent from the mouse to the host device, as shown in Fig. 10. Each of the 11-bit words contains a ‘0’ start bit, followed by 8 bits of data (LSB first), followed by an odd parity bit, and terminated with a ‘1’ stop bit. Thus, each data transmission contains 33 bits, where bits 0, 11, and 22 are ‘0’ start bits, and bits 11, 21, and 33 are ‘1’ stop bits. The three 8-bit data fields contain movement data as shown in the figure above. Data is valid at the falling edge of the clock, and the clock period is 20 to 30 kHz.

The mouse assumes a relative coordinate system wherein moving the mouse to the right generates a positive number in the X field, and moving to the left generates a negative number. Likewise, moving the mouse up

generates a positive number in the Y field, and moving down represents a negative number (the XS and YS bits in the status byte are the sign bits – a '1' indicates a negative number). The magnitude of the X and Y numbers represent the rate of mouse movement – the larger the number, the faster the mouse is moving (the XV and YV bits in the status byte are movement overflow indicators – a '1' means overflow has occurred). If the mouse moves continuously, the 33-bit transmissions are repeated every 50ms or so. The L and R fields in the status byte indicate Left and Right button presses (a '1' indicates the button is being pressed).

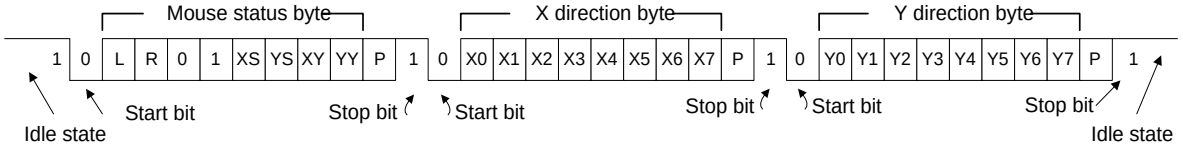


Figure 13. Mouse Data Format.

The microcontroller also supports Microsoft Intellimouse-type extensions for reporting back a third axis representing the mouse wheel, as shown in

Table 9.

Command	Action
EA	Set stream mode. The mouse responds with "acknowledge" (0xFA) then resets its movement counters and enters stream mode.
F4	Enable data reporting. The mouse responds with "acknowledge" (0xFA) then enables data reporting and resets its movement counters. This command only affects behavior in stream mode. Once issued, mouse movement will automatically generate a data packet.
F5	Disable data reporting. The mouse responds with "acknowledge" (0xFA) then disables data reporting and resets its movement counters.
F3	Set mouse sample rate. The mouse responds with "acknowledge" (0xFA) then reads one more byte from the host. This byte is then saved as the new sample rate, and a new "acknowledge" packet is issued.
FE	Resend. FE directs mouse to re-send last packet.
FF	Reset. The mouse responds with "acknowledge" (0xFA) then enters reset mode.

Table 9. Microsoft Intellimouse type extensions, commands and actions.

12 User USB 2.0

When the fixed USB roles of the Genesys2 are not enough, an on-board USB 2.0 transceiver (PHY) provides physical layer implementation for any USB 2.0 user-application. It connects to a USB A (J7-bottom row) and a USB AB micro (J6) receptacle in parallel, enabling device, host, and OTG USB roles without the need for cable adaptors. Use only one of the connectors at a time, the one fitting the desired USB role.

The transceiver is a TUSB1210 with a UTMI+ low pin interface (ULPI) towards the FPGA. ULPI is a 12-pin SDR interface clocked at 60 MHz, resulting in the maximum data rate of 480 Mbps. On the Genesys 2 the transceiver provides the ULPI clock, which is wired to a clock-capable input pin of the FPGA. Figure 14 shows the connection diagram of the PHY.

The part of the USB 2.0 stack above the physical layer has to be implemented in the FPGA. Xilinx offers an AXI USB 2.0 Device Controller IP that needed separate licensing at the time of writing.

Auxiliary circuitry, like an electronic power switch and jumpers are used to implement different USB roles. In Host applications, the switch can be commanded through the PHY to power VBUS. The switch has built-in current limit and short circuit protection. VBUS error conditions are signaled to the FPGA by means of a status pin. In Device role, this switch should be disabled. A jumper (JP1) in series with the power switch can be used to make sure VBUS is not inadvertently powered. Another jumper (JP2) (dis)connects a 150 µF capacitor to/from the VBUS. Use it to implement the different bus capacitance requirements of the USB roles.

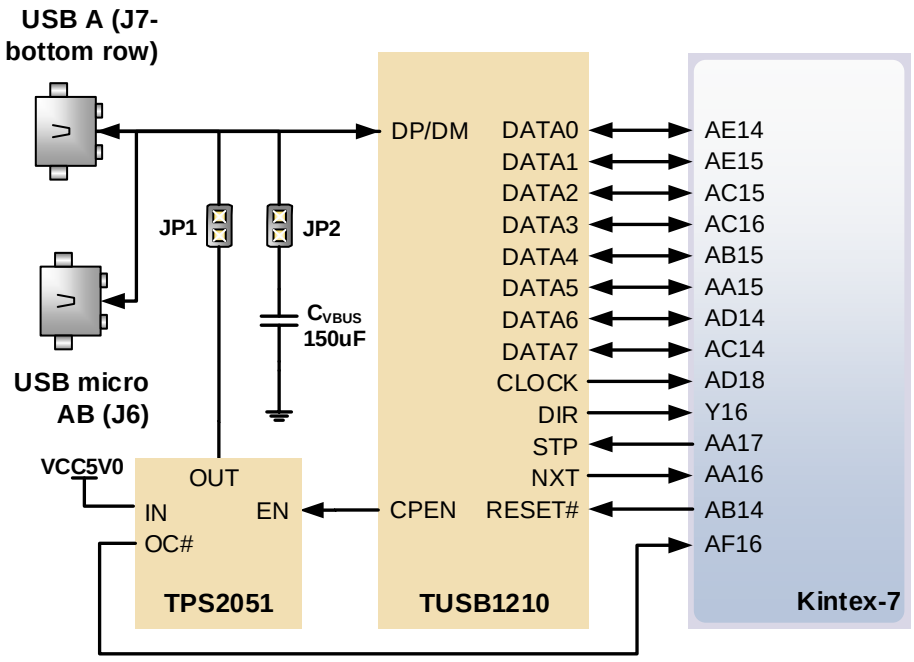


Figure 14. USB 2.0 Transceiver PHY connection diagram.

For more information, refer to the TUSB1210 datasheet[12] and the USB 2.0 specifications[13].

13 Basic I/O

The Genesys 2 board includes eight slide switches, six push buttons, and eight individual LEDs. The pushbuttons and slide switches are connected to the FPGA via series resistors to prevent damage from inadvertent short circuits (a short circuit could occur if an FPGA pin assigned to a pushbutton or slide switch was inadvertently defined as an output). The five pushbuttons arranged in a plus-sign configuration are "momentary" switches that normally generate a low output when they are at rest, and a high output only when they are pressed. The red pushbutton labeled “CPU RESET,” on the other hand, generates a high output when at rest and a low output when pressed. The CPU RESET button is intended to be used in processor designs to reset the processor, but you can also use it as a general purpose pushbutton. Slide switches generate constant high or low inputs depending on their position.

[12] http://www.ti.com/lit/gpn/tusb1210
[13] http://www.usb.org/developers/docs/usb20_docs

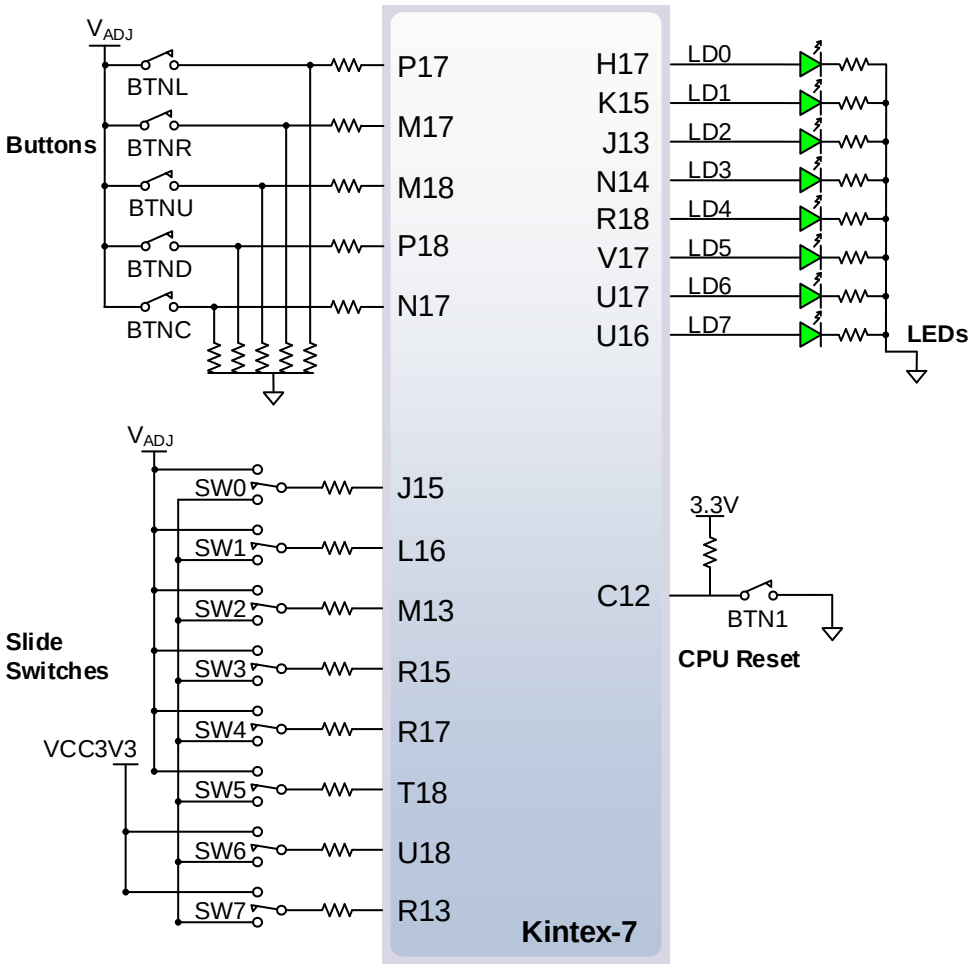


Figure 15. General purpose I/O connections.

The eight individual high-efficiency LEDs are anode-connected to the FPGA via 330-ohm resistors, so they will turn on when a logic high voltage is applied to their respective I/O pin. Additional LEDs that are not user-accessible indicate power-on, FPGA programming status, and USB and Ethernet port status.

14 Pmod Connectors

The Pmod connectors are arranged in a 2x6 right-angle, 100-mil female connectors that mate with standard 2x6 pin headers. Each 12-pin Pmod connector provides two power pins (6 and 12), two ground pins (5 and 11), and eight logic signals, as shown in Fig. 20. The VCC and Ground pins of can deliver up to 1A of current per pin. Pin assignments for the Pmod I/O connected to the FPGA are shown in Figure 16.

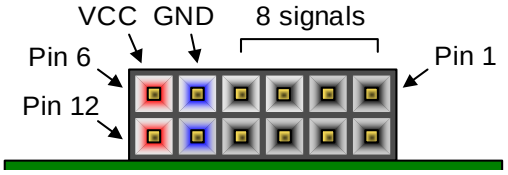


Figure 16. PMOD Connectors- Front view as loaded on PCB.

The Genesys 2 features four Pmod connectors of different "styles" with subtle differences between them.

Table 10 summarizes these differences.

Pmod Connector	Power	Analog/Digital	Routing	Series protection	Recommended usage
JXADC	V_{ADJ}	Dual	Differential; Pairs: 1-7,2-8,3-9,4-10	100 ohm	Analog inputs; LVDS_25 input/output (V_{ADJ}=2.5V)
JA, JB	3.3 V	Digital-only	Differential; Pairs: 1-2,3-4,7-8,9-10	0 ohm	>=10MHz; LVDS_25 input
JC, JD	3.3 V	Digital-only	Single-ended	200 ohm	<10 MHz, LVCMOS33

Table 10. Pmod differences.

JXADC is a dual digital/analog Pmod wired to auxiliary analog FPGA input pins (see description below). JA, JB, JC and JD are digital-only but routed in a different manner. JC and JD signal traces are routed as single-ended signals with no length matching or characteristic impedance guarantees. JA and JB connector signals, on the other hand, are routed differentially in pairs at 100-ohm characteristic impedance. Inter-pair matching is +- 20mm. The series resistor on these two Pmods are replaced by 0 ohm shunts, so care must be exercised not to use voltages higher than 3.3V, or short FPGA-driven pins.
For slow digital signals, below 10 MHz, all Pmod connectors are equivalent, if V_{ADJ} is set to 3.3 V (see the 2 Power Supplies section). For higher frequencies, the routing and series protection need to be taken into account as well.

Pmod Pinouts				
JA1: B13	JB1: G14	JC1: K2	JD1:	JXADC1: A13
JA2: F14	JB2: P15	JC2: E7	JD2:	JXADC2: A15
JA3: D17	JB3: V11	JC3: J3	JD3:	JXADC3: B16
JA4: E17	JB4: V15	JC4: J4	JD4:	JXADC4: B18
JA7: G13	JB7: K16	JC7: K1	JD7:	JXADC7: A14
JA8: C17	JB8: R16	JC8: E6	JD8:	JXADC8: A16
JA9: D18	JB9: T9	JC9: J2	JD9:	JXADC9: B17
JA10: E18	JB10: U11	JC10: G6	JD10:	JXADC10: A18

Table 11. Genesys 2 Pmod Pin Assignments.

Digilent produces a large collection of Pmod accessory boards that can attach to the Pmod expansion connectors to add ready-made functions like A/D's, D/A's, motor drivers, sensors, and other functions. See www.digilentinc.com for more information.

14.1 Dual Analog/Digital Pmod

The on-board Pmod expansion connector labeled "JXADC" is wired to the auxiliary analog input pins of the FPGA. Depending on the configuration, this connector can be used to input differential analog signals to the analog-to-digital converter inside the Kintex-7 (XADC). Any or all pairs in the connector can be configured either as analog input or digital input-output.

The Dual Analog/Digital Pmod on the Genesys 2 differs from the rest in the routing of its traces. The eight data signals are grouped into four pairs, with the pairs routed closely coupled for better analog noise immunity. Furthermore, each pair has a partially loaded anti-alias filter (100 ohm, 1 nF) laid out on the PCB. The filter does not have capacitors C151, C152, C153, and C154. In designs where such filters are desired, the capacitors can be manually loaded by the user.

NOTE: The coupled routing and the anti-alias filters might limit the data speeds when used for digital signals.

The XADC core within the Kintex-7 is a dual channel 12-bit analog-to-digital converter capable of operating at 1 MSPS. Either channel can be driven by any of the auxiliary analog input pairs connected to the JXADC header. The XADC core is controlled and accessed from a user design via the Dynamic Reconfiguration Port (DRP). This includes access to the temperature sensor and voltage monitors inside the FPGA. For more information on using the XADC core, refer to the Xilinx document titled "7 Series FPGAs and Zynq-7000 All Programmable SoC XADC Dual 12-Bit 1 MSPS Analog-to-Digital Converter User Guide" (ug480[14]).

15 High Pin Count FMC Connector

The Genesys 2 includes a FPGA Mezzanine Card (FMC) Standard-conforming carrier card connector that enables connecting mezzanine modules compliant with the same standard. Genesys 2-based designs can now be easily extended with custom or off-the-shelf high-performance modules.

The actual connector used is a 400-pin Samtec ASP-134486-01, the high-pin count, 10mm stacking height variant of the standard. It is fully bonded to the FPGA, with the exception of CLK3_BIDIR. The 80 differential pairs wired to regular FPGA user I/O pins are grouped into FMC banks: 34 pairs in LA, 24 pairs in HA and 22 pairs in HB. LA and HA pairs are powered by the Genesys 2 V_{ADJ} rail adjustable in the 1.2V-3.3V range. On the other hand, HB signals are referenced to a voltage rail provided by the FMC mezzanine card (VIO_B_M2C). For this reason, a whole FPGA bank is dedicated to HB signals exclusively. This bank remains unpowered, and HB signals cannot be used, if no FMC mezzanine card, or one that does not provide VIO_B_M2C is connected to the Genesys 2.

Thanks to the flexible voltage range supported by the Genesys 2 it allows high compatibility with existing and future FMC modules. The Genesys 2 opens the door to the full range of I/O standards supported by the Kintex-7 HR (High Range) I/O architecture over the FMC connector.

The pin-out of the FMC connector can be found in the UCF/XDC constraints file available on www.digilentinc.com.

For above-gigabit speed rates, all ten gigabit transceiver lanes and two accompanying clock pairs are wired to GTX transceiver banks. Kintex-7 transceiver banks group four lanes and two reference clock input together, and are also called quads.

Each transceiver lane includes a receive pair and a transmit pair. Lanes DP0-DP3 are wired to quad 115. Lanes DP4-DP7 go to quad 116, along with the two reference clocks GBTCLK0 and GBTCLK1. The last two lanes DP8 and DP9 are connected to quad 117, while the rest of the pins in these three quads are left unused.

Since an MGTREFCLK can be routed to both the quad above and below its own, both reference clocks can be used to clock any channel in the three quads.

Table 12, Table 13, and

Table 14 show how the FMC gigabit signals are mapped to pins and GTX primitives. Refer to the 7 Series FPGAs GTX/GTH Transceivers User Guide (ug476[15]) for more information.

[14] http://www.xilinx.com/support/documentation/user_guides/ug480_7Series_XADC.pdf
[15] http://www.xilinx.com/support/documentation/user_guides/ug476_7Series_Transceivers.pdf

Quad	Primitive		Pin type	Pin	FMC signal
115	GTXE2_CHANNEL	X0Y0	MGTXTXP/N0	Y2/Y1	DP0_C2M_P/N
			MGTXRXP/N0	AA4/AA3	DP0_M2C_P/N
		X0Y1	MGTXTXP/N1	V2/V1	DP1_C2M_P/N
			MGTXRXP/N1	Y6/Y5	DP1_M2C_P/N
		X0Y2	MGTXTXP/N2	U4/U3	DP2_C2M_P/N
			MGTXRXP/N2	W4/W3	DP2_M2C_P/N
		X0Y3	MGTXTXP/N3	T2/T1	DP3_C2M_P/N
			MGTXRXP/N3	V6/V5	DP3_M2C_P/N

Table 12. Quad 115 pin-out.

Quad	Primitive		Pin type	Pin	FMC signal
116	GTXE2_CHANNEL	X0Y4	MGTXTXP/N0	P2/P1	DP4_C2M_P/N
			MGTXRXP/N0	T6/T5	DP4_M2C_P/N
		X0Y5	MGTXTXP/N1	N4/N3	DP5_C2M_P/N
			MGTXRXP/N1	R4/R3	DP5_M2C_P/N
		X0Y6	MGTXTXP/N2	M2/M1	DP6_C2M_P/N
			MGTXRXP/N2	P6/P5	DP6_M2C_P/N
		X0Y7	MGTXTXP/N3	L4/L3	DP7_C2M_P/N
			MGTXRXP/N3	M6/M5	DP7_M2C_P/N
	IBUFDS_GTE2	X0Y2	MGTREFCLKP/N0	L8/L7	GBTCLK0_P/N
		X0Y3	MGTREFCLKP/N1	N8/N7	GBTCLK1_P/N

Table 13. Quad 116 pin-out.

Quad	Primitive		Pin type	Pin	FMC signal
117	GTXE2_CHANNEL	X0Y8	MGTXTXP/N0	K2/K1	DP8_C2M_P/N
			MGTXRXP/N0	K6/K5	DP8_M2C_P/N
		X0Y9	MGTXTXP/N1	J4/J3	DP9_C2M_P/N
			MGTXRXP/N1	H6/H5	DP9_M2C_P/N

Table 14. Quad 117 pin-out.

16 MicroSD Slot

The Genesys 2 provides a microSD slot for both FPGA configuration and user access. The on-board Auxiliary Function microcontroller shares the SD card bus with the FPGA. Before the FPGA is configured the microcontroller must have access to the SD card via an SPI interface. Once a bit file is downloaded to the FPGA (from any source), the microcontroller powers off the SD slot and relinquishes control of the bus. The FPGA design will find the SD card in an unpowered state.

All of the SD pins on the FPGA are wired to support full SD speeds in native interface mode, as shown in Figure 17. The SPI interface is also available, if needed. Once control over the SD bus is passed from the microcontroller to the FPGA, the SD_RESET signal needs to be actively driven low by the FPGA to power the microSD card slot.

To talk to an SD card, several communication layers need to be implemented in the FPGA. The physical layer (de)serializes command and data packets over either the SD native or SPI interface. The data link layer should implement the SD state machine, issuing initialization and read/write commands specific to the SD standard. The data link layer provides access to raw blocks/sectors on the SD card. To access a formatted card, a file system layer should abstract sectors into files and directories. On top of the file system layer comes the actual application.
For more information on implementing an SD card controller, refer to the SD card specification available at www.sdcard.org.

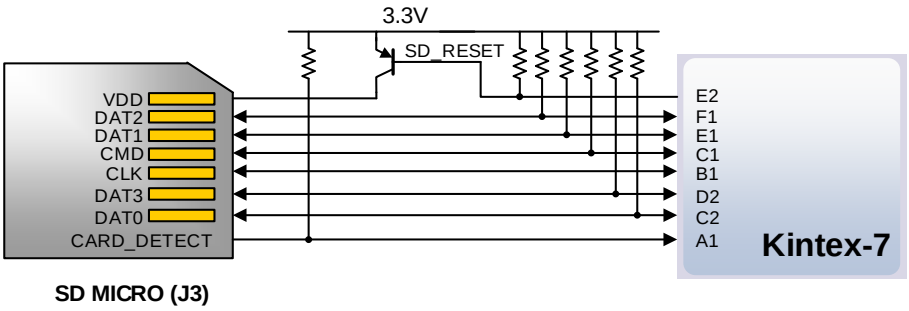


Figure 17. Kintex-7 microSD card connector interface (PIC24 connections not shown).

17 HDMI

The Genesys 2 board contains two buffered HDMI ports: one source port J4 (output), and one sink port J5 (input). Both ports use HDMI type-A receptacles and include HDMI buffer TMDS141. The buffers work by terminating, equalizing, conditioning and forwarding the HDMI stream between the connector and FPGA pins.

Both HDMI and DVI systems use the same TMDS signaling standard, directly supported by Kintex-7 user I/O infrastructure. Also, HDMI sources are backward compatible with DVI sinks and vice versa. Thus, simple passive adaptors (available at most electronics stores) can be used to drive a DVI monitor or accept a DVI input. The HDMI receptacle only includes digital signals, so only DVI-D mode is possible.

The 19-pin HDMI connectors include three differential data channels, one differential clock channel five GND connections, a one-wire Consumer Electronics Control (CEC) bus, a two-wire Display Data Channel (DDC) bus, a Hot Plug Detect (HPD) signal, a 5V power pin capable of delivering up to 50mA, and one reserved (RES) pin. All are wired to the FPGA with the exception of RES.

Pin/Signal	J4 (source)		J5 (sink)	
	Description	FPGA pin	Description	FPGA pin
D[2:0]+/-	Data output		Data input	
CLK+/-	Clock output		Clock input	
CEC	Consumer Electronics Control bidirectional		Consumer Electronics Control bidirectional	
SCL, SDA	DDC bidirectional		DDC bidirectional	
HPD	Hot-plug detect input **(inverted, active-low**)		Hot-plug assert output (**active-high**)	

5V0	Powered from 5V rail		Powers auxiliary signals	

Figure 18. HDMI pin description and assignment.

13.1 TMDS signals

HDMI/DVI is a high-speed digital video stream interface using transition-minimized differential signaling (TMDS). To make proper use of either of the HDMI ports a standard-compliant transmitter or receiver needs to be implemented in the FPGA. The implementation details are outside the scope of this manual.

13.2 Auxiliary signals

Presence of a sink on the cable is announced on the hot-plug detect (HPD) pin. Whenever a sink is ready and wishes to announce its presence, it connects the HPD pin to the 5V0 supply pin. On the Genesys 2 this is achieved by pulling HDMI_RX_HPA high. This signal defaults low. The source reads the HPD pin through an inverting level-translator, so HDMI_TX_HPD reads low when a sink is present.

The Display Data Channel, or DDC, is a collection of protocols that enable communication between the display (sink) and graphics adapter (source). The DDC2B variant is based on I^2C, the bus master being the source and the bus slave the sink. When a source detects high level on the HPD pin, it queries the sink over the DDC bus for video capabilities. It determines whether the sink is DVI or HDMI-capable and what resolutions are supported. Only afterwards will video transmission begin. Refer to VESA E-DDC specifications[16] for more information.

The Consumer Electronics Control, or CEC, is an optional protocol that allows control messages to be passed around an HDMI chain between different products. A common use case is a TV passing control messages originating from a universal remote to a DVR or satellite receiver. It is a one-wire protocol at 3.3V level connected to an FPGA user I/O pin. The wire can be controlled in an open-drain fashion allowing for multiple devices sharing a common CEC wire. Refer to the CEC addendum of HDMI 1.3 or later specifications[17] for more information.

14 DisplayPort

DisplayPort is a relatively new industry standard for digital display technology. Its advantages over existing technologies are: higher bandwidth for greater resolutions and color depths, bi-directional auxiliary channel, variable interface width, and flexible power topologies among others.

DisplayPort defines a high-speed main link carrying audio and video data, an auxiliary channel, and a hot-plug detect signal. The main link is a unidirectional, high-bandwidth and low-latency channel. It consists of one, two, or four AC-coupled differential pairs called lanes. Version 1.1 of the standard[16] defines two link rates: 1.62 and 2.7 Gbps. The lanes carry both data and an embedded clock at the link rate negotiated between Source and Sink, independent of the resolution and color depth of the video stream. The link rate is de-coupled from the pixel rate, resulting in a packetized stream. This differentiates DisplayPort from other digital video standards like DVI/HDMI. Due to the high link rate, the main link can only be implemented on dedicated gigabit transceiver pins of the Kintex-7 architecture.

[16] http://www.vesa.org
[17] http://www.hdmi.org

The auxiliary channel is a bidirectional channel for link management and device control. It is AC-coupled, just like the main link lanes, but uses a different encoding and the lower data rate of 1Mbps. Upon hot-plug detection a Source will attempt to configure the link through link training. Handshaking link parameters happens via the auxiliary channel.

Genesys 2 includes two Mini DisplayPort (mDP) connectors: one wired as Source, the other as Sink. Both support a maximum lane count of four on the main link. MGT quad 118 is dedicated to DisplayPort. On-board there is a 135 MHz reference oscillator mapped to MGTREFCLK0 which should be used to generate the desired link rate. See

Table 15 for pin mapping. Refer to the Xilinx 7 Series FPGAs GTX/GTH Transceivers User Guide (ug476[15]) for more information on how to implement high-speed interfaces.

Quad	Primitive		Pin type	Pin	DisplayPort signal
118	GTXE2_CHANNEL	X0Y12	MGTXTXP/N0	D2/D1	Source Lane 0
			MGTXRXP/N0	E4/E3	Sink Lane 0
		X0Y13	MGTXTXP/N1	C4/C3	Source Lane 1
			MGTXRXP/N1	D6/D5	Sink Lane 1
		X0Y14	MGTXTXP/N2	B2/B1	Source Lane 2
			MGTXRXP/N2	B6/B5	Sink Lane 2
		X0Y15	MGTXTXP/N3	A4/A3	Source Lane 3
			MGTXRXP/N3	A8/A7	Sink Lane 3
	IBUFDS_GTE2	X0Y6	MGTREFCLKP/N0	C8/C7	135 MHz

Table 15. DisplayPort Quad pin-out.

The auxiliary channel is a bidirectional LVDS bus. Depending on the Xilinx tool/IP version, instantiating a differential I/O buffer with LVDS signaling standard might not be possible. The work-around is to have two pairs of pins wired and shorted together as seen in Figure 19. One pair should be implemented as input-only and the other as output-only.
If the tool/IP allows bidirectional LVDS buffers, only one of the pairs needs to be used (it does not matter which), while the other declared as input and not used.

The hot-plug detect (HPD) signal connects to a general user I/O pin and should be configured as an input.

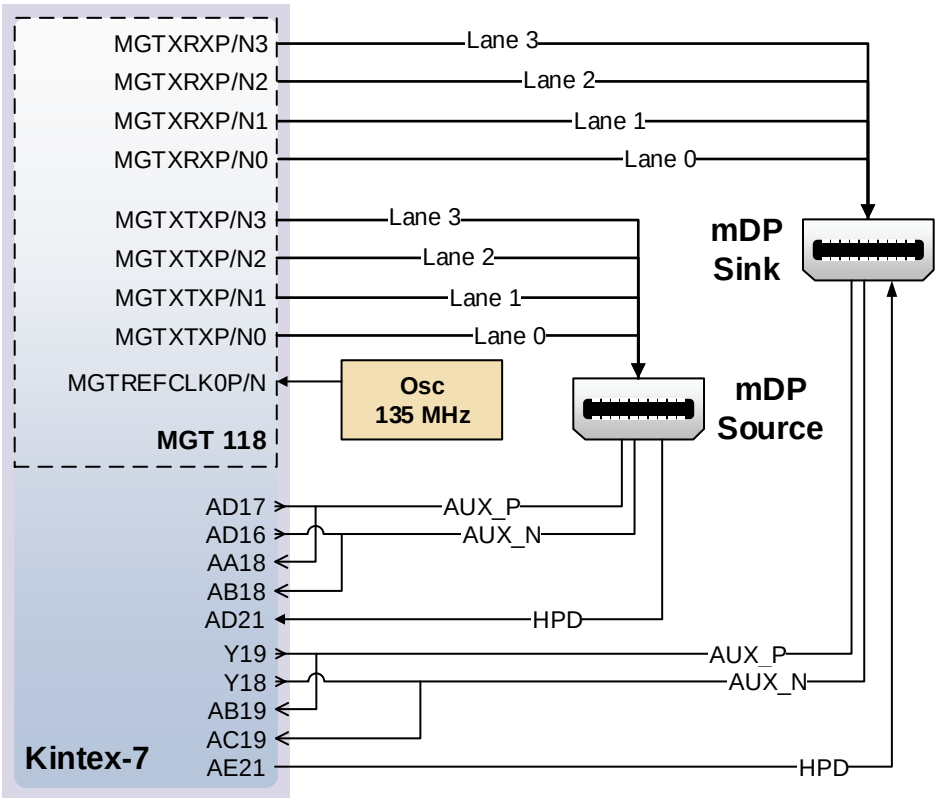


Figure 19. DisplayPort wiring diagram.

The full implementation details of the DisplayPort standard is outside the scope of this document. Refer to the full specifications published by VESA for more details. DisplayPort Source and Sink IPs can be licensed separately from Xilinx.

15 OLED

A Univision Technology Inc. UG-2832HSWEG04 is loaded on the Genesys 2. It is a white monochrome, 128 x 32, 0.91" organic LED display matrix bundled with a Solomon Systech SSD1306 display controller. The display data interface towards the FPGA is a 4-wire serial peripheral interface (SPI). The 4 wires in controller-terminology are CS#, D/C#, SDIN, and SCLK, but CS# is hard-wired to ground. This adds to the reset and two power control signals for proper start-up sequencing. The signals are summarized in Table 16.

Signal	Description	Polarity	FPGA pin
RES#	Reset	Active-low	
CS#	Chip select (always active)	Active-low	N/A
D/C#	Data (high)/Command (low)	Both	
SCLK	Serial Clock	Active-high	
SDIN	Serial Data	Active-high	
VBAT#	Power enable for internal power supply	Active-low	
VDD#	Power enable for digital power	Active-low	

Table 16. OLED signal description.

The serial interface is synchronous to SCLK and must conform the timing specifications below. In most cases, a 10 MHz SCLK and data sent on the falling edge should work.

(V_{DD} - V_{SS} = 1.65V to 3.3V, T_A = 25°C)

Symbol	Parameter	Min	Typ	Max	Unit
t_{cycle}	Clock Cycle Time	100	-	-	ns
t_{AS}	Address Setup Time	15	-	-	ns
t_{AH}	Address Hold Time	15	-	-	ns
t_{CSS}	Chip Select Setup Time	20	-	-	ns
t_{CSH}	Chip Select Hold Time	10	-	-	ns
t_{DSW}	Write Data Setup Time	15	-	-	ns
t_{DHW}	Write Data Hold Time	15	-	-	ns
t_{CLKL}	Clock Low Time	20	-	-	ns
t_{CLKH}	Clock High Time	20	-	-	ns
t_R	Rise Time	-	-	40	ns
t_F	Fall Time	-	-	40	ns

Figure 13-3 : 4-wire Serial interface characteristics

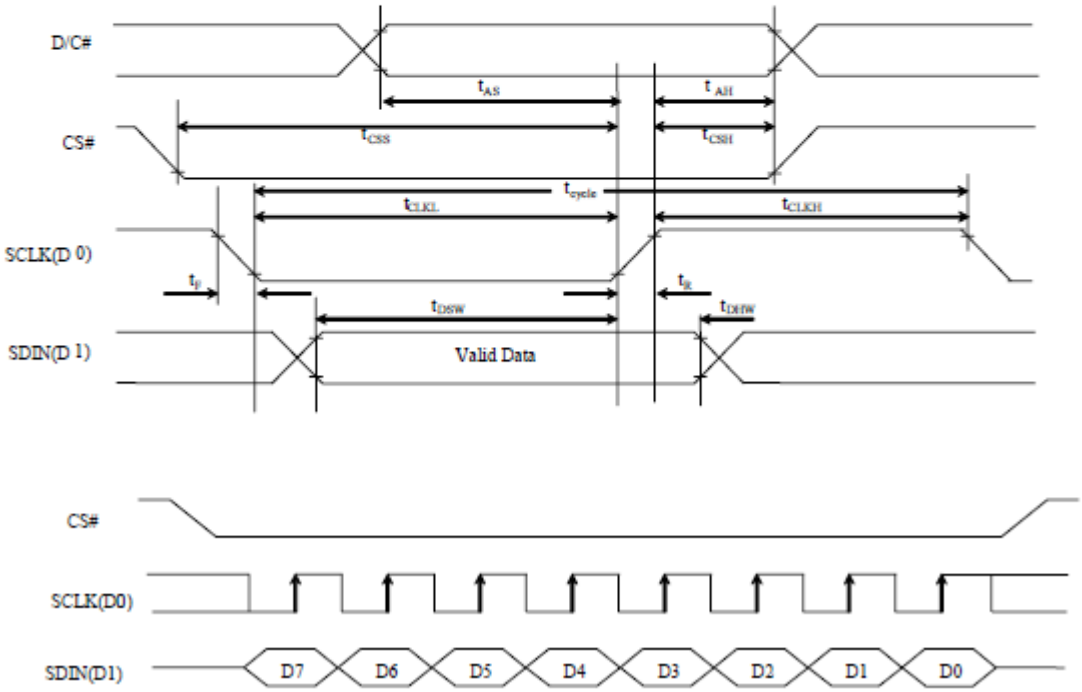


Figure 20. Serial interface timing diagram.

Start-up sequence:

1. Power up VDD by pulling OLED_VDD **low**. Wait 1ms.
2. Pulse RES# low for at least 3us.
3. *Send initialization/configuration commands (see* Table 17).
4. Power up VBAT by pulling OLED_VBAT **low**. Wait 100ms for voltage to stabilize.
5. Clear screen by writing zero to the display buffer.
6. Send “Display On” command (0xAF).

Command function	Command bytes
Charge pump enable	0x8D, 0x14
Set pre-charge period	0xD9, 0xF1

Command function	Command bytes
Contrast control	0x81, 0x0F
Column inversion disable	0xA0
Scan direction	0xC0
COM pins configuration	0xDA, 0x00
Addressing mode: horizontal	0x20

Table 17. OLED configuration commands.

After start-up, writing to the display is done by sending data bytes over the serial interface (D/C# high). Each data bit corresponds to a pixel, with the addressing mode, inversion and scan direction settings determining exactly which.

16 Audio Codec

The Genesys 2 board includes an Analog Devices ADAU1761 SigmaDSP audio codec (IC11) complementing its multimedia features. Four 1/8" (3.5mm) audio jacks are available for line-out (J11-green), headphone-out (J10-black), line-in (J13-blue), and microphone-in (J12-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 in an FPGA design, the audio data needs to be converted. The audio codec bridges the gap between the analog jacks and the digital FPGA pins. 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 I^2S. 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 I^2C. It responds to slave address 0b0111011, 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 I^2C port. The exact frequency depends on the desired sample rate and whether PLL will be used, but 12 MHz is a good start. The clocking infrastructure of 7 series FPGA is more than capable of synthesizing the right frequency from the on-board 100 MHz reference oscillator.

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 2. For example, the OUT port is differential, comprising 4 pins: LOUTP, LOUTN, ROUTP, and ROUTN. 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 data sheet[18].

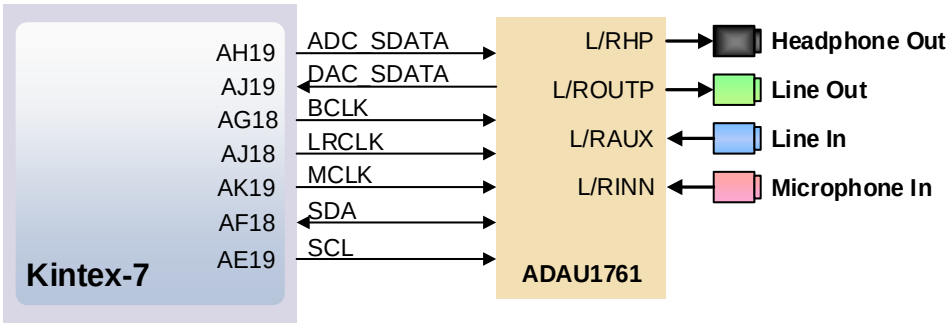


Figure 21. Audio signal connections.

Signal Name	FPGA Pin	Pin Function
ADC_SDATA	AH19	Serialized audio resulting from the analog-to-digital conversion (record).
DAC_SDATA	AJ19	Serialized audio is converted to analog by the codec (playback).
BCLK	AG18	Serial data port clock.
LRCLK	AJ18	Serial data port frame clock.
MCLK	AK19	Master clock.
SDA	AF18	I²C configuration interface.
SCL	AE19	I²C configuration interface.

Table 18. Audio signal description.

[18] http://www.analog.com/media/en/technical-documentation/data-sheets/ADAU1761.pdf