1300 Henley Court
Pullman, WA 99163
509.334.6306
www.digilentinc.com

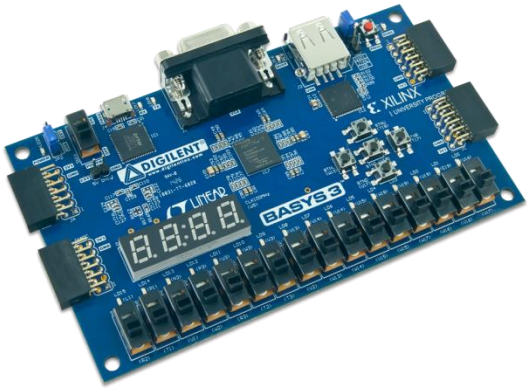
Basys 3™ FPGA Board Reference Manual

Revised April 8, 2016
This manual applies to the Basys 3 rev. C

Overview

The Basys 3 board is a complete, ready-to-use digital circuit development platform based on the latest Artix®-7 Field Programmable Gate Array (FPGA) from Xilinx®. With its high-capacity FPGA (Xilinx part number XC7A35T-1CPG236C), low overall cost, and collection of USB, VGA, and other ports, the Basys 3 can host designs ranging from introductory combinational circuits to complex sequential circuits like embedded processors and controllers. It includes enough switches, LEDs, and other I/O devices to allow a large number of designs to be completed without the need for any additional hardware, and enough uncommitted FPGA I/O pins to allow designs to be expanded using Digilent Pmods or other custom boards and circuits.

The Artix-7 FPGA is optimized for high performance logic, and offers more capacity, higher performance, and more resources than earlier designs. Artix-7 35T features include:



The Basys 3.

- 33,280 logic cells in 5200 slices (each slice contains four 6-input LUTs and 8 flip-flops)
- 1,800 Kbits of fast block RAM
- Five clock management tiles, each with a phase-locked loop (PLL)
- 90 DSP slices
- Internal clock speeds exceeding 450MHz
- On-chip analog-to-digital converter (XADC)

The Basys 3 also offers an improved collection of ports and peripherals, including:

- 16 user switches
- 4-digit 7-segment display
- 12-bit VGA output
- Digilent USB-JTAG port for FPGA programming and communication
- 16 user LEDs
- Three Pmod ports
- USB-UART Bridge
- USB HID Host for mice, keyboards and memory sticks
- 5 user pushbuttons
- Pmod for XADC signals
- Serial Flash

The Basys 3 works with Xilinx's new high-performance Vivado™ Design Suite. Vivado includes many new tools and design flows that facilitate and enhance the latest design methods. It runs faster, allows better use of FPGA resources, and allows designers to focus their time evaluating design alternatives. The System Edition includes an on-chip logic analyzer, high-level synthesis tool, other cutting-edge tools, and the free WebPACK™ version allows Basys 3 designs to be created at no additional cost.

Copyright Digilent, Inc. All rights reserved.
Other product and company names mentioned may be trademarks of their respective owners.

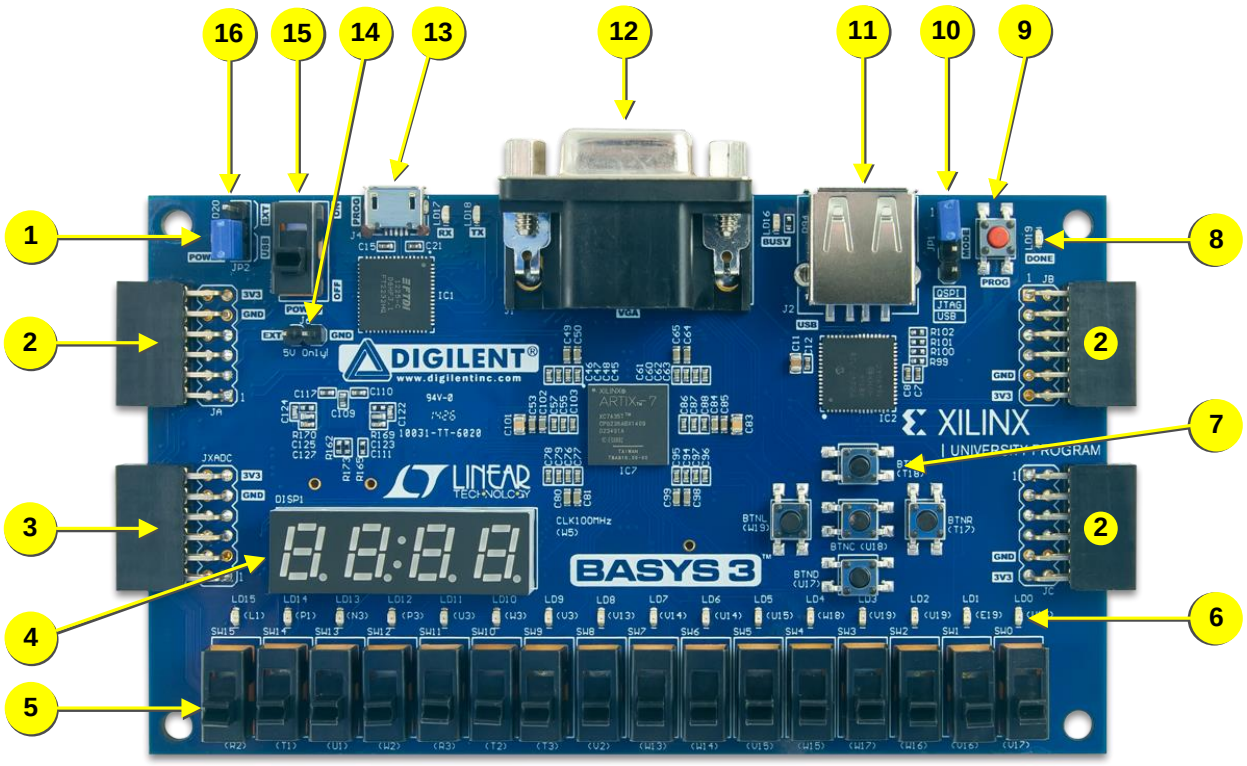


Figure 1. Basys 3 FPGA board with callouts.

Callout	Component Description	Callout	Component Description
1	Power good LED	9	FPGA configuration reset button
2	Pmod port(s)	10	Programming mode jumper
3	Analog signal Pmod port (XADC)	11	USB host connector
4	Four digit 7-segment display	12	VGA connector
5	Slide switches (16)	13	Shared UART/ JTAG USB port
6	LEDs (16)	14	External power connector
7	Pushbuttons (5)	15	Power Switch
8	FPGA programming done LED	16	Power Select Jumper

Table 1. Basys 3 Callouts and component descriptions.

A growing collection of board support IP, reference designs, and add-on boards are available on the Digilent website. See the Basys 3 page at www.digilentinc.com for more information.

1 Power Supplies

The Basys 3 board can receive power from the Digilent USB-JTAG port (J4) or from a 5V external power supply. Jumper JP3 (near the power switch) determines which source is used.

All Basys 3 power supplies can be turned on and off by a single logic-level power switch (SW16). A power-good LED (LD20), driven by the "power good" output of the LTC3633 supply, indicates that the supplies are turned on and operating normally. An overview of the Basys 3 power circuit is shown in Fig. 2.

Copyright Digilent, Inc. All rights reserved.
Other product and company names mentioned may be trademarks of their respective owners.

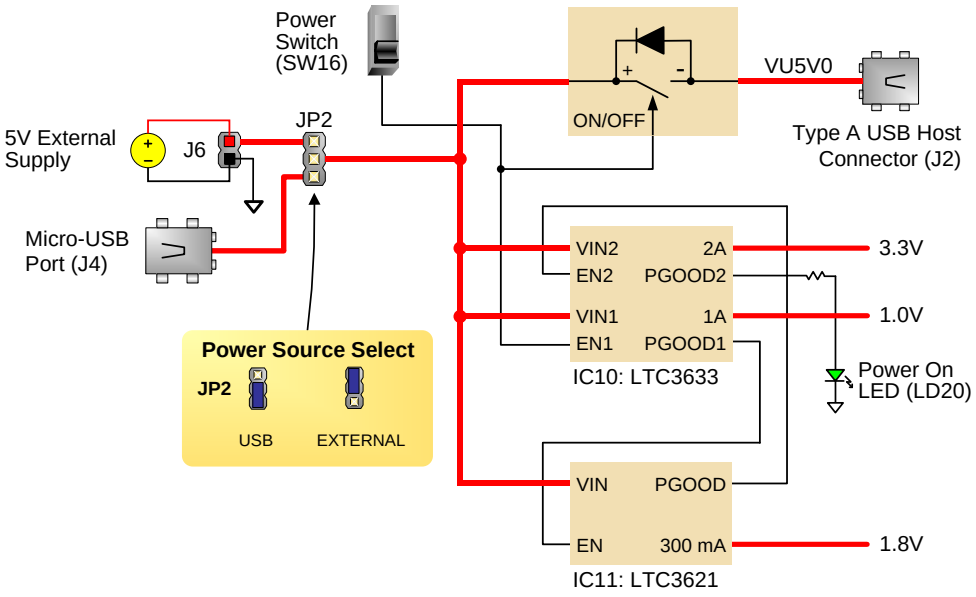


Figure 2. Basys 3 power circuit.

The USB port can deliver enough power for the vast majority of designs. 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 or battery pack can be used.

An external power supply can be used by plugging into the external power header (J6) and setting jumper JP2 to "EXT". The supply must deliver 4.5VDC to 5.5VDC and at least 1A of current (i.e., at least 5W of power). Many suitable supplies can be purchased through Digi-Key or other catalog vendors.

An external battery pack can be used by connecting the battery's positive terminal to the "EXT" pin of J6 and the negative terminal to the "GND" pin of J6. The power provided to USB devices that are connected to Host connector J2 is not regulated. Therefore, it is necessary to limit the maximum voltage of an external battery pack to 5.5V DC. The minimum voltage of the battery pack depends on the application; if the USB Host function (J2) is used, at least 4.6V needs to be provided. In other cases, the minimum voltage is 3.6V.

Voltage regulator circuits from Linear Technology create the required 3.3V, 1.8V, and 1.0V supplies from the main power input. Table 2 provides additional information (typical currents depend strongly on FPGA configuration and the values provided are typical of medium size/speed designs).

Supply	Circuits	Device	Current (max/typical)
3.3V	FPGA I/O, USB ports, Clocks, Flash, PMODs	IC10: LTC3633	2A/0.1 to 1.5A
1.0V	FPGA Core	IC10: LTC3633	2A/ 0.2 to 1.3A
1.8V	FPGA Auxiliary and Ram	IC11: LTC3621	300mA/ 0.05 to 0.15A

Table 2. Basys 3 power supplies.

Copyright Digilent, Inc. All rights reserved.
Other product and company names mentioned may be trademarks of their respective owners.

2 FPGA Configuration

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

1. A PC can use the Digilent USB-JTAG circuitry (portJ4, labeled "PROG") 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 using the SPI port.
3. A programming file can be transferred from a USB memory stick attached to the USB HID port.

Figure 3 shows the different options available for configuring the FPGA. An on-board "mode" jumper (JP1) selects between the programming modes.

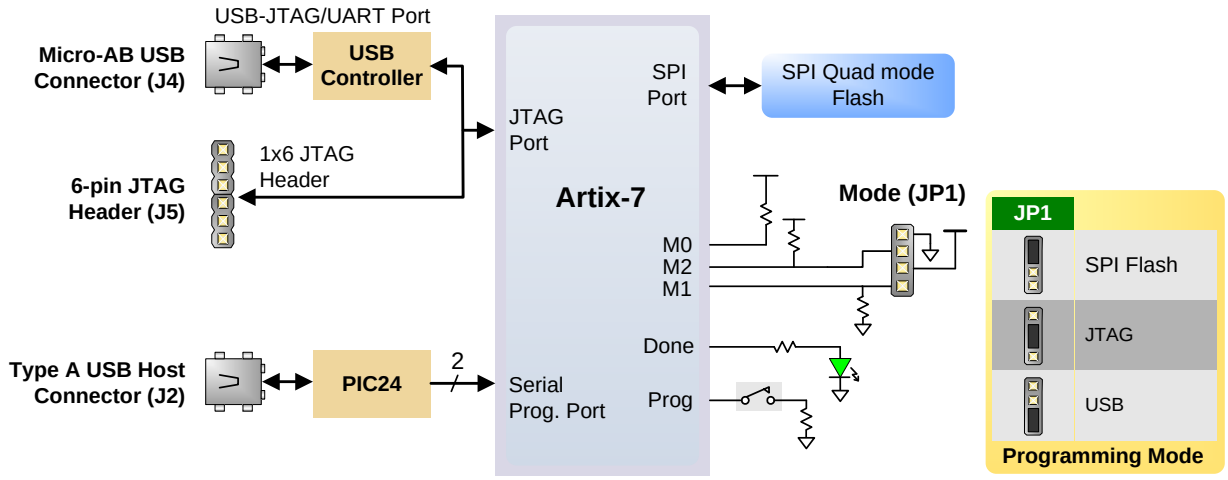


Figure 3. Basys 3 configuration options.

The FPGA configuration data is stored in files called bitstreams that have the .bit file extension. The Vivado software from Xilinx can create bitstreams from VHDL, Verilog®, or schematic-based source files.

Bitstreams are stored in 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.

An Artix-7 35T bitstream is typically 17,536,096 bits and can take a long time to transfer. The time it takes to program the Basys 3 can be decreased by compressing the bitstream before programming, and then allowing the FPGA to decompress the bitsream itself during configuration. Depending on design complexity, compression ratios of 10x can be achieved. Bitstream compression can be enabled within the Xilinx Tools (Vivado) to occur during generation. For instructions on how to do this, consult the Xilinx documentation for the toolset being used.

After being successfully programmed, the FPGA will cause the "DONE" LED to illuminate. Pressing the "PROG" button 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 jumper.

The following sections provide greater detail about programming the Basys 3 using the different methods available.

Copyright Digilent, Inc. All rights reserved.
Other product and company names mentioned may be trademarks of their respective owners.

2.1 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 J4) or an external JTAG programmer, such as the Digilent JTAG-HS2 attached to port J5 (located below port JA). You can perform JTAG programming any time after the Basys 3 has been powered on regardless of what the mode jumper (JP1) 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 (seen in Fig. 3) is useful to prevent the FPGA from being configured from any other bitstream source until a JTAG programming occurs.

Programming the Basys 3 with an uncompressed bitstream using the on-board USB_JTAG circuitry usually takes around five seconds. JTAG programming can be done using the hardware server in Vivado. The demonstration project available at digilentinc.com provides an in-depth tutorial on how to program your board.

2.2 JTAG Programming

When programming a nonvolatile flash device, a bitstream file is transferred to the flash in a two-step process. First, the FPGA is programmed with a circuit 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 (see Fig. 3). 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 one or two 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.

Quad-SPI programming can be performed using Vivado.

2.3 USB Host Programming

You can program the FPGA from a pen drive attached to the USB-HID port (J2) by doing the following:

1. Format the storage device (Pen drive) 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 Basys 3.
4. Set the JP1 Programming Mode jumper on the Basys 3 to "USB".
5. Push the PROG button or power-cycle the Basys 3.

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

The Auxiliary Function Status, or "BUSY" LED (LD16), 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 (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.

Copyright Digilent, Inc. All rights reserved.
Other product and company names mentioned may be trademarks of their respective owners.

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

3 Memory

The Basys 3 board contains a 32Mbit non-volatile serial Flash device, which is attached to the Artix-7 FPGA using a dedicated quad-mode (x4) SPI bus. The connections and pin assignments between the FPGA and the serial flash device are shown in Fig. 4.

FPGA configuration files can be written to the Quad SPI Flash (Spansion part number S25FL032), and mode settings are available to cause the FPGA to automatically read a configuration from this device at power on. An Artix-7 35T configuration file requires just over two Mbytes of memory, leaving approximately 48% of the flash device available for user data.

NOTE: Refer to the manufacturer's data sheets and the reference designs posted on Digilent's website for more information about the memory devices.

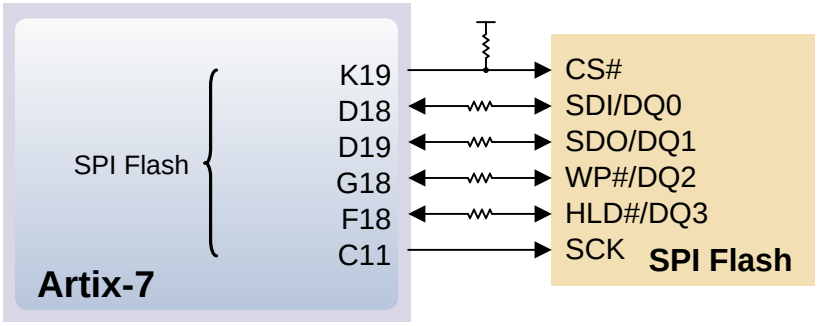


Figure 4. Basys 3 external memory.

4 Oscillators/Clocks

The Basys 3 board includes a single 100 MHz oscillator connected to pin W5 (W5 is a MRCC input on bank 34). The 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 100 MHz input clock. For a full description of these rules and of the capabilities of the Artix-7 clocking resources, refer to the "7 Series FPGAs Clocking Resources User Guide" available from Xilinx.

Xilinx offers the LogiCORE™ Clocking Wizard IP to help users generate the different clocks required for a specific design. This wizard properly instantiates 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 clocking resources that can be inserted into the user's design. The Clocking Wizard can be accessed from within IP Catalog, which can be found under the Project Manager section of the Flow Navigator in Vivado.

Copyright Digilent, Inc. All rights reserved.
Other product and company names mentioned may be trademarks of their respective owners.

5 USB-UART Bridge (Serial Port)

The Basys 3 includes an FTDI FT2232HQ USB-UART bridge (attached to connector J4) that allows you to use PC applications to communicate with the board using standard Windows COM port commands. Free USB-COM port drivers, available from 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). 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 B18 and A18 FPGA pins.

Two on-board status LEDs provide visual feedback on traffic flowing through the port: the transmit LED (LD18) and the receive LED (LD17). 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 Basys 3 to be programmed, communicated with via UART, and powered from a computer attached with a single Micro USB cable. The connections between the FT2232HQ and the Artix-7 are shown in Fig. 6.

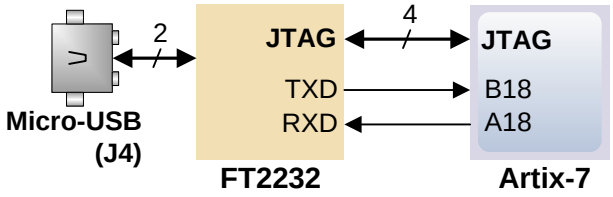


Figure 6. Basys 3 FT2232HQ connections.

6 USB HID Host

The Auxiliary Function microcontroller (Microchip PIC24FJ128) provides the Basys 3 with USB HID host capability. After power-up, the microcontroller is in configuration mode, either downloading a bitstream to the FPGA or waiting for it to be programmed from other sources. Once the FPGA is programmed, the microcontroller switches to application mode, which in this case is USB HID Host mode. Firmware in the microcontroller can drive a mouse or a keyboard attached to the type A USB connector at J2 labeled "USB." 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.

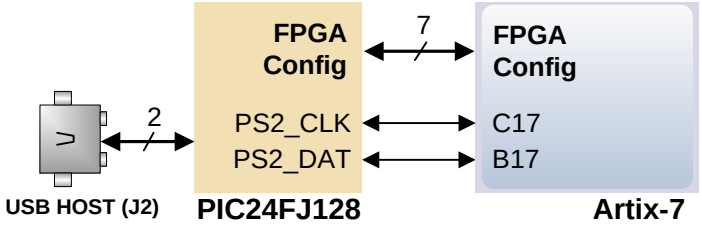
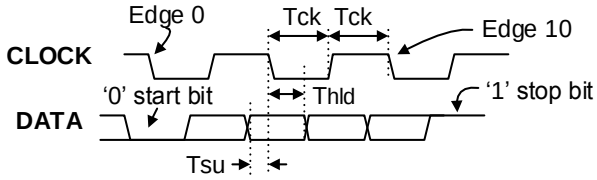


Figure 7. Basys 3 PIC24 connections.

Copyright Digilent, Inc. All rights reserved.
Other product and company names mentioned may be trademarks of their respective owners.

6.1 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 Basys 3, 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 Fig. 8.



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 8. 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 Basys 3, 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.

6.2 Keyboard

The keyboard uses open-collector drivers so the keyboard, or an attached host device, can drive the two-wire bus (if the host device 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 Fig. 9.

Copyright Digilent, Inc. All rights reserved.
Other product and company names mentioned may be trademarks of their respective owners.

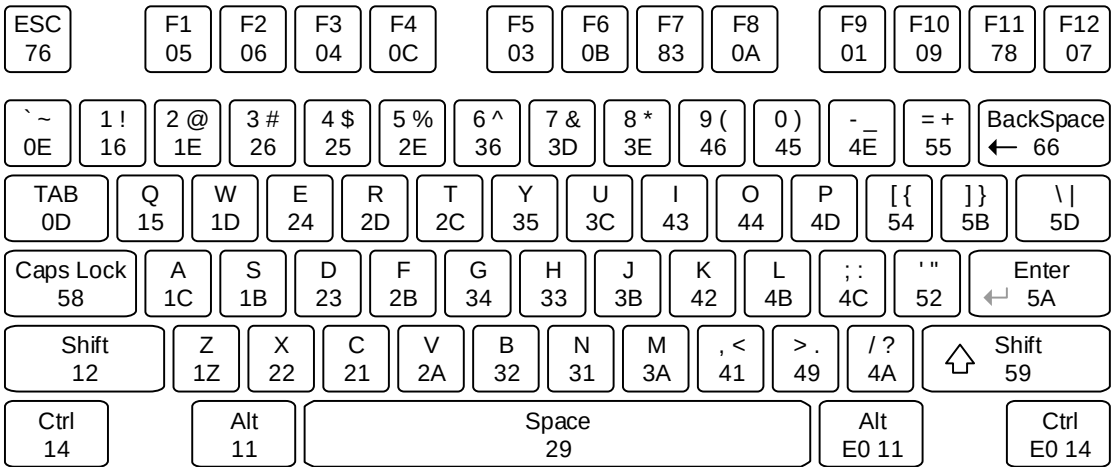


Figure 9. Keyboard scan codes.

A host device can also send data to the keyboard. Table 3 shows a list of some common commands a host might send.

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.

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 3. Keyboard commands.

6.3 Mouse

Once entered in stream mode and data reporting has been 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 Fig. 10. Data is valid at the falling edge of the clock, and the clock period is 20 to 30 KHz.

Copyright Digilent, Inc. All rights reserved.
Other product and company names mentioned may be trademarks of their respective owners.

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 that the button is being pressed).

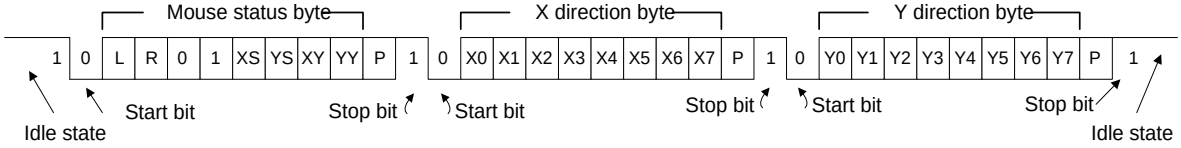


Figure 10. Mouse data format.

The microcontroller also supports Microsoft® IntelliMouse®-type extensions for reporting back a third axis representing the mouse wheel, as shown in Table 4.

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.

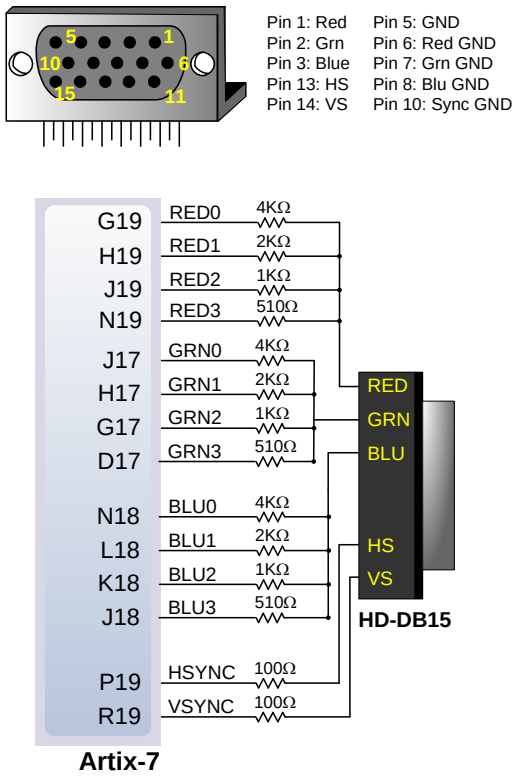
Table 4. Microsoft Intellimouse-type extensions, commands, and actions.

7 VGA Port

NOTE: *A helpful way to understand the way that VGA signals are transmitted is to understand the method of which CRT (Cathode Ray Tubes) function for displaying images. Although the technology may seem outdated, it is from this legacy that many of the signal names and timings have originated.*

The Basys 3 board uses 14 FPGA signals to create a VGA port with 4-bits per color and the two standard sync signals (HS – Horizontal Sync, and VS – Vertical Sync). The color signals use resistor-divider circuits that work in conjunction with the 75 ohm termination resistance of the VGA display to create 16 signal levels each on the red, green, and blue VGA signals. This circuit, shown in Fig. 11, produces video color signals that proceed in equal increments between 0V (fully off) and 0.7V (fully on). Using this circuit, 4096 different colors can be displayed, one for each unique 12-bit pattern. A video controller circuit must be created in the FPGA to drive the sync and color signals with the correct timing in order to produce a working display system.

Copyright Digilent, Inc. All rights reserved.
Other product and company names mentioned may be trademarks of their respective owners.



7.1 VGA System Timing

VGA signal timings are specified, published, copyrighted, and sold by the VESA® organization (www.vesa.org). The following VGA system timing information is provided as an example of how a VGA monitor might be driven in 640 by 480 mode.

NOTE: For more precise information, or for information on other VGA frequencies, refer to documentation available at the VESA website.

CRT-based VGA displays use amplitude-modulated moving electron beams (or cathode rays) to display information on a phosphor-coated screen. LCD displays use an array of switches that can impose a voltage across a small amount of liquid crystal, thereby changing light permittivity through the crystal on a pixel-by-pixel basis. Although the following description is limited to CRT displays, LCD displays have evolved to use the same signal timings as CRT displays (so the "signals" discussion below pertains to both CRTs and LCDs). Color CRT displays use three electron beams (one for red, one for blue, and one for green) to energize the phosphor that coats the inner side of the display end of a cathode ray tube (see Fig. 12).

Copyright Digilent, Inc. All rights reserved.
Other product and company names mentioned may be trademarks of their respective owners.

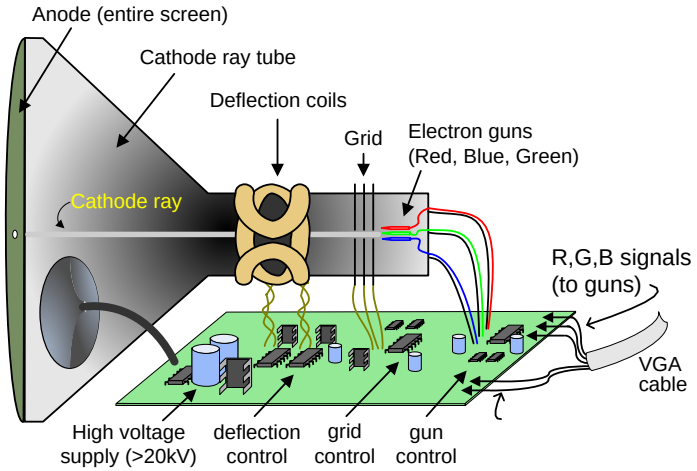


Figure 12. Color CRT display.

Electron beams emanate from "electron guns" which are finely-pointed heated cathodes placed in close proximity to a positively charged annular plate called a "grid." The electrostatic force imposed by the grid pulls rays of energized electrons from the cathodes, and those rays are fed by the current that flows into the cathodes. These particle rays are initially accelerated towards the grid, but they soon fall under the influence of the much larger electrostatic force that results from the entire phosphor-coated display surface of the CRT being charged to 20kV (or more). The rays are focused to a fine beam as they pass through the center of the grids, and then they accelerate to impact on the phosphor-coated display surface. The phosphor surface glows brightly at the impact point, and it continues to glow for several hundred microseconds after the beam is removed. The larger the current fed into the cathode, the brighter the phosphor will glow.

Between the grid and the display surface, the beam passes through the neck of the CRT where two coils of wire produce orthogonal electromagnetic fields. Because cathode rays are composed of charged particles (electrons), they can be deflected by these magnetic fields. Current waveforms are passed through the coils to produce magnetic fields that interact with the cathode rays and cause them to transverse the display surface in a "raster" pattern, horizontally from left to right and vertically from top to bottom, as shown in Fig. 13. As the cathode ray moves over the surface of the display, the current sent to the electron guns can be increased or decreased to change the brightness of the display at the cathode ray impact point.

Information is only displayed when the beam is moving in the "forward" direction (left to right and top to bottom), and not during the time the beam is reset back to the left or top edge of the display. Much of the potential display time is therefore lost in "blanking" periods when the beam is reset and stabilized to begin a new horizontal or vertical display pass. The size of the beams, the frequency at which the beam can be traced across the display, and the frequency at which the electron beam can be modulated determine the display resolution.

Modern VGA displays can accommodate different resolutions, and a VGA controller circuit dictates the resolution by producing timing signals to control the raster patterns. The controller must produce synchronizing pulses at 3.3V (or 5V) to set the frequency at which current flows through the deflection coils, and it must ensure that video data is applied to the electron guns at the correct time. Raster video displays define a number of "rows" that corresponds to the number of horizontal passes the cathode makes over the display area, and a number of "columns" that corresponds to an area on each row that is assigned to one "picture element" or pixel. Typical displays use from 240 to 1200 rows and from 320 to 1600 columns. The overall size of a display and the number of rows and columns determines the size of each pixel.

Copyright Digilent, Inc. All rights reserved.
Other product and company names mentioned may be trademarks of their respective owners.

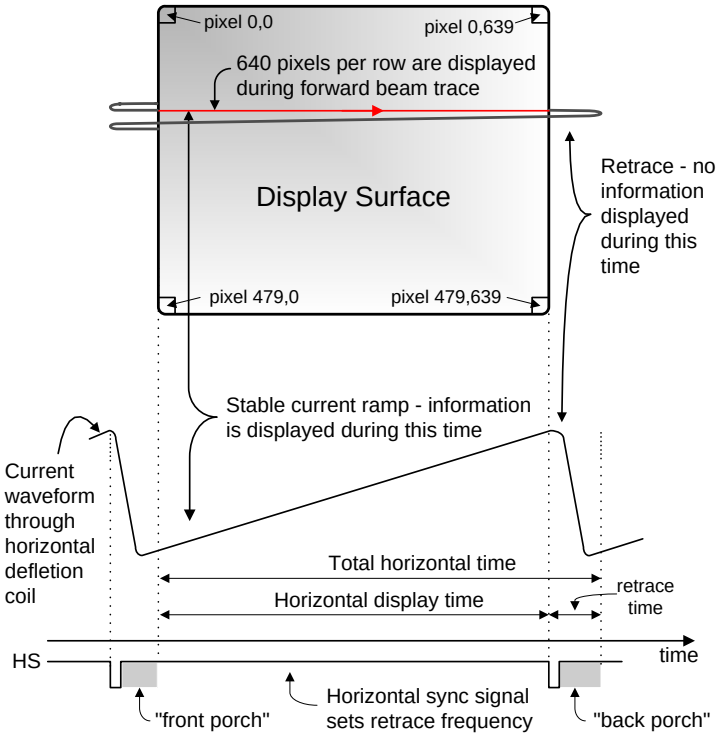
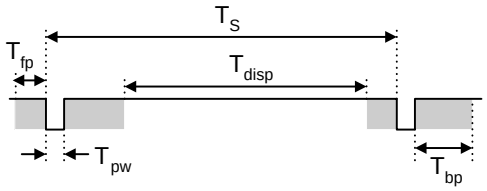


Figure 13. VGA horizontal synchronization.

Video data typically comes from a video refresh memory; with one or more bytes assigned to each pixel location (the Basys 3 uses 12-bits per pixel). The controller must index into video memory as the beams move across the display, and retrieve and apply video data to the display at precisely the time the electron beam is moving across a given pixel.

A VGA controller circuit must generate the HS and VS timings signals and coordinate the delivery of video data based on the pixel clock. The pixel clock defines the time available to display one pixel of information. The VS signal defines the "refresh" frequency of the display, or the frequency at which all information on the display is redrawn. The minimum refresh frequency is a function of the display's phosphor and electron beam intensity, with practical refresh frequencies falling in the 50Hz to 120Hz range. The number of lines to be displayed at a given refresh frequency defines the horizontal "retrace" frequency. For a 640-pixel by 480-row display using a 25 MHz pixel clock and 60 +/-1Hz refresh, the signal timings shown in Fig. 14 can be derived. Timings for sync pulse width and front and back porch intervals (porch intervals are the pre- and post-sync pulse times during which information cannot be displayed) are based on observations taken from actual VGA displays.

Copyright Digilent, Inc. All rights reserved.
Other product and company names mentioned may be trademarks of their respective owners.



Symbol	Parameter	Vertical Sync			Horiz. Sync	
		Time	Clocks	Lines	Time	Clks
T_S	Sync pulse	16.7ms	416,800	521	32 us	800
T_{disp}	Display time	15.36ms	384,000	480	25.6 us	640
T_{pw}	Pulse width	64 us	1,600	2	3.84 us	96
T_{fp}	Front porch	320 us	8,000	10	640 ns	16
T_{bp}	Back porch	928 us	23,200	29	1.92 us	48

Figure 14. Signal timings for a 640-pixel by 480 row display using a 25 MHz pixel clock and 60 Hz vertical refresh.

A VGA controller circuit, such as the one diagramed in Fig. 15, decodes the output of a horizontal-sync counter driven by the pixel clock to generate HS signal timings. You can use this counter to locate any pixel location on a given row. Likewise, the output of a vertical-sync counter that increments with each HS pulse can be used to generate VS signal timings, and you can use this counter to locate any given row. These two continually running counters can be used to form an address into video RAM. No time relationship between the onset of the HS pulse and the onset of the VS pulse is specified, so you can arrange the counters to easily form video RAM addresses, or to minimize decoding logic for sync pulse generation.

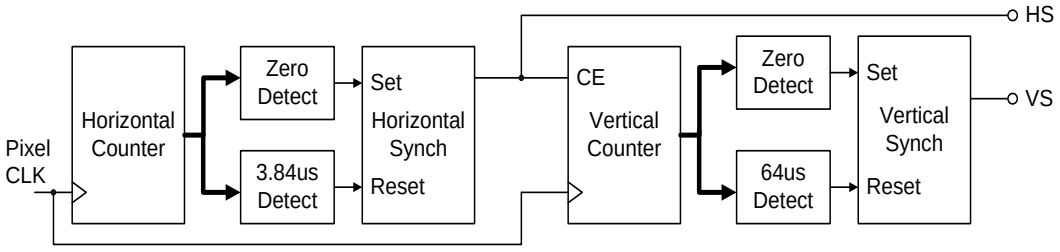


Figure 15. VGA display controller block diagram.

8 Basic I/O

The Basys 3 board includes sixteen slide switches, five push buttons, sixteen individual LEDs, and a four-digit seven-segment display, as shown in Fig. 16. 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. Slide switches generate constant high or low inputs depending on their position.

Copyright Digilent, Inc. All rights reserved.
Other product and company names mentioned may be trademarks of their respective owners.

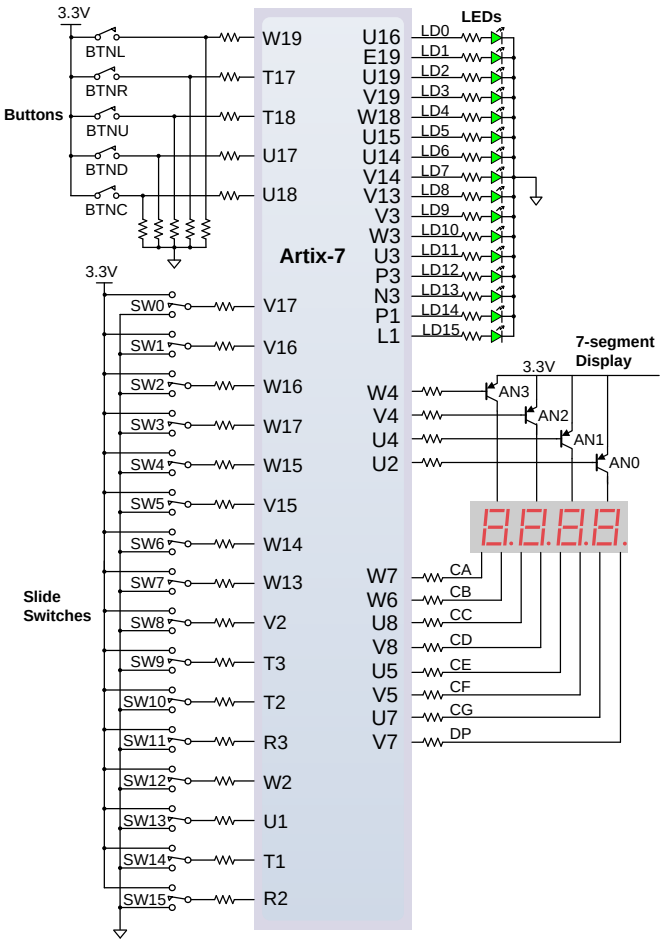


Figure 16. General purpose I/O devices on the Basys 3.

The sixteen 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, which are not user-accessible, indicate power-on, FPGA programming status, and USB port status.

8.1 Seven-Segment Display

The Basys 3 board contains one four-digit common anode seven-segment LED display. Each of the four digits is composed of seven segments arranged in a "figure 8" pattern, with an LED embedded in each segment. Segment LEDs can be individually illuminated, so any one of 128 patterns can be displayed on a digit by illuminating certain LED segments and leaving the others dark, as shown in Fig. 17. Of these 128 possible patterns, the ten corresponding to the decimal digits are the most useful.

Copyright Digilent, Inc. All rights reserved.
Other product and company names mentioned may be trademarks of their respective owners.

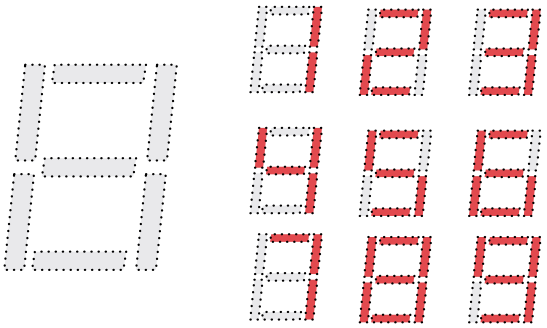


Figure 17. An un-illuminated seven-segment display and nine illumination patterns corresponding to decimal digits.

The anodes of the seven LEDs forming each digit are tied together into one "common anode" circuit node, but the LED cathodes remain separate, as shown in Fig. 18. The common anode signals are available as four "digit enable" input signals to the 4-digit display. The cathodes of similar segments on all four displays are connected into seven circuit nodes labeled CA through CG (for example, the four "D" cathodes from the four digits are grouped together into a single circuit node called "CD"). These seven cathode signals are available as inputs to the 4-digit display. This signal connection scheme creates a multiplexed display, where the cathode signals are common to all digits but they can only illuminate the segments of the digit whose corresponding anode signal is asserted.

To illuminate a segment, the anode should be driven high while the cathode is driven low. However, since the Basys 3 uses transistors to drive enough current into the common anode point, the anode enables are inverted. Therefore, both the AN0..3 and the CA..G/DP signals are driven low when active.

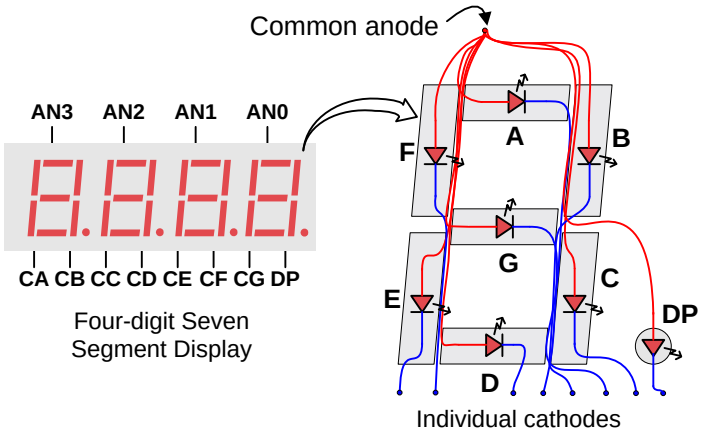


Figure 18. Common anode circuit node.

A scanning display controller circuit can be used to show a four-digit number on this display. This circuit drives the anode signals and corresponding cathode patterns of each digit in a repeating, continuous succession at an update rate that is faster than the human eye can detect. Each digit is illuminated just one-fourth of the time, but because the eye cannot perceive the darkening of a digit before it is illuminated again, the digit appears continuously illuminated. If the update, or "refresh", rate is slowed to around 45Hz, a flicker can be noticed in the display.

For each of the four digits to appear bright and continuously illuminated, all four digits should be driven once every 1 to 16ms, for a refresh frequency of about 1 KHz to 60Hz. For example, in a 62.5Hz refresh scheme, the entire display would be refreshed once every 16ms, and each digit would be illuminated for 1/4 of the refresh cycle, or 4ms. The controller must drive the cathodes low with the correct pattern when the corresponding anode signal is driven high. To illustrate the process, if AN0 is asserted while CB and CC are asserted, then a "1" will be displayed

Copyright Digilent, Inc. All rights reserved.
Other product and company names mentioned may be trademarks of their respective owners.

in digit position 1. Then, if AN1 is asserted while CA, CB, and CC are asserted, a "7" will be displayed in digit position 2. If AN0, CB, and CC are driven for 4ms, and then AN1, CA, CB, and CC are driven for 4ms in an endless succession, the display will show "71" in the first two digits. An example timing diagram for a four-digit controller is shown in Fig. 19.

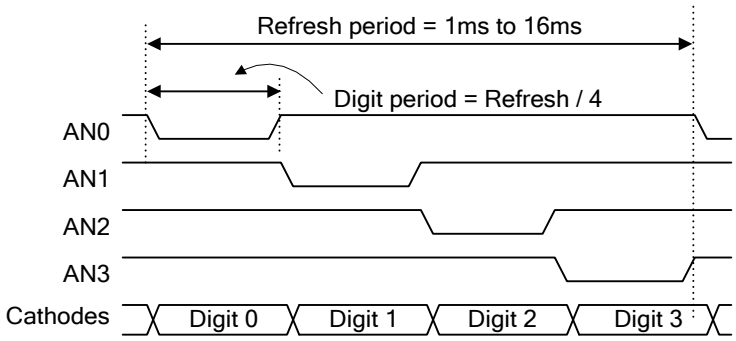


Figure 19. Four digit scanning display controller timing diagram.

9 Pmod Ports

The Pmod ports are arranged in a 2x6 right-angle, and are 100-mil female connectors that mate with standard 2x6 pin headers. Each 12-pin Pmod port provides two 3.3V VCC signals (pins 6 and 12), two Ground signals (pins 5 and 11), and eight logic signals, as shown in Fig. 20. The VCC and Ground pins can deliver up to 1A of current. Pmod data signals are not matched pairs, and they are routed using best-available tracks without impedance control or delay matching. Pin assignments for the Pmod I/O connected to the FPGA are shown in Table 6.

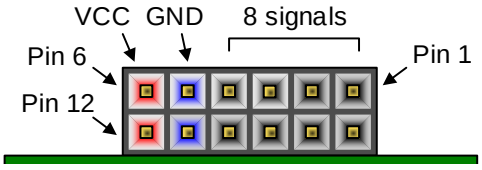


Figure 20. Pmod ports; front view as loaded on PCB.

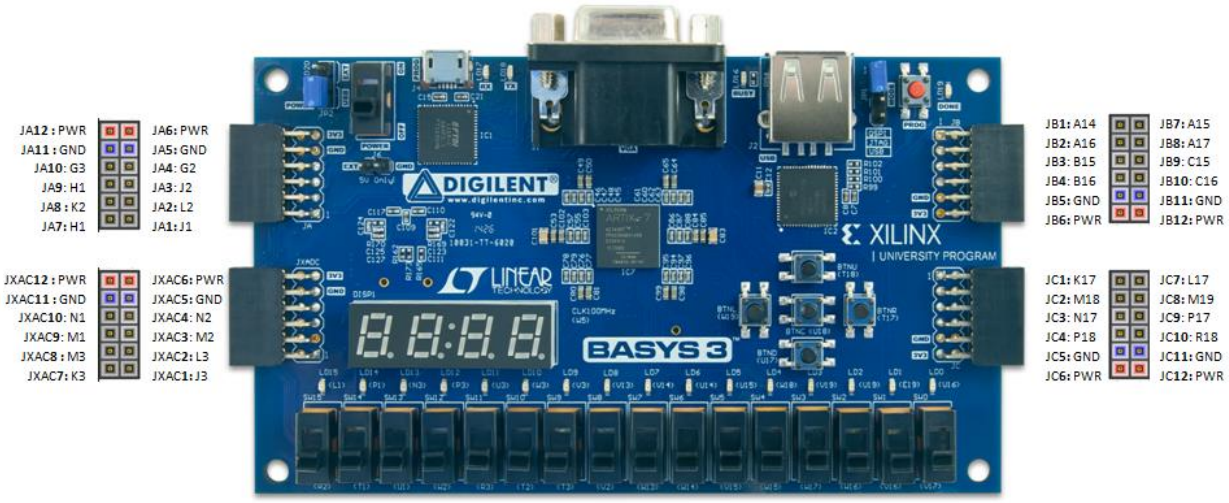
Pmod JA	Pmod JB	Pmod JC	Pmod XDAC
JA1: J1	JB1: A14	JC1: K17	JXADC1: J3
JA2: L2	JB2: A16	JC2: M18	JXADC2: L3
JA3: J2	JB3: B15	JC3: N17	JXADC3: M2
JA4: G2	JB4: B16	JC4: P18	JXADC4: N2
JA7: H1	JB7: A15	JC7: L17	JXADC7: K3
JA8: K2	JB8: A17	JC8: M19	JXADC8: M3
JA9: H2	JB9: C15	JC9: P17	JXADC9: M1
JA10: G3	JB10: C16	JC10: R18	JXADC10: N1

Table 6. Basys 3 Pmod pin assignment.

Copyright Digilent, Inc. All rights reserved.
Other product and company names mentioned may be trademarks of their respective owners.

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

Basys3: Pmod Pin-Out Diagram



9.1 Dual Analog/Digital Pmod

The on-board Pmod expansion port, 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 Artix-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 Basys 3 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 laid out on the PCB. The filter does not have capacitors C33-C36. 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 Artix-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). The DRP also provides access to voltage monitors that are present on each of the FPGA's power rails, and a temperature sensor that is internal to 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."

Copyright Digilent, Inc. All rights reserved.
Other product and company names mentioned may be trademarks of their respective owners.

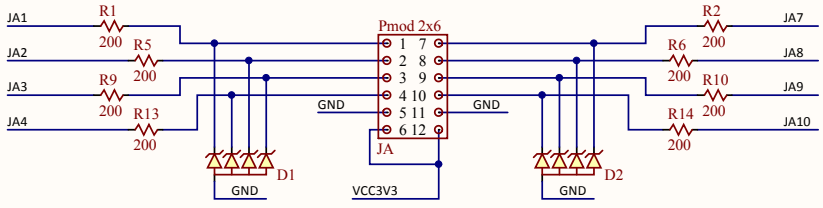
10 Built-In Self-Test

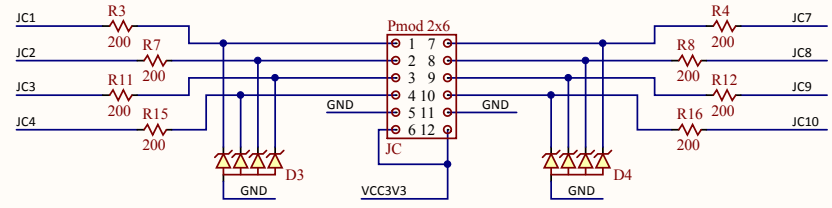
A demonstration configuration is loaded into the SPI Flash device on the Basys 3 board during manufacturing. The source code and prebuilt bitstream for this design are available for download from the Digilent website. If the demo configuration is present in the SPI Flash device and the Basys 3 board is powered on in SPI mode, the demo project will allow basic hardware verification. Here is an overview of how this demo drives the different onboard components:

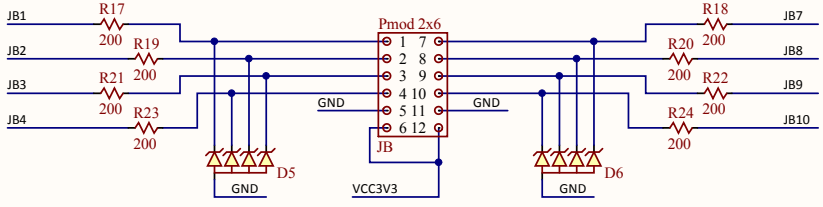
- The user LEDs are illuminated when the corresponding user switch is placed in the on position.
- The VGA port displays feedback from a USB Mouse.
- Connecting a mouse to the USB-HID Mouse port will allow the pointer on the VGA display to be controlled.
- On power-up, each digit of the seven-segment display will display a counter output from 0-9 that increments once a second.
- Pressing BTNU, BTNL, BTNR, or BTND will cause a digit of the seven-segment display to go blank.
- Pressing BTNC will reset the design.
- On power-up, a welcome message is sent over the UART. Also, every time a button is pressed a message is sent. The UART can be connected to using a terminal program with 9600 Baud, 8 data bits, 1 stop bit, and no parity.

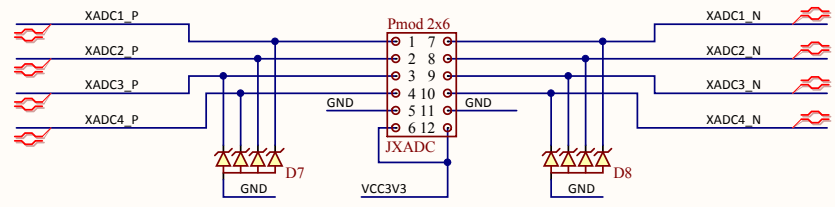
All Basys 3 boards are 100% tested during the manufacturing process. If any device on the Basys 3 board fails test or is not responding properly, it is likely that damage occurred during transport or during use. Typical damage includes stressed solder joints and contaminants in switches and buttons resulting in intermittent failures. Stressed solder joints can be repaired by reheating and reflowing solder and contaminants can be cleaned with off-the-shelf electronics cleaning products. If a board fails test within the warranty period, it will be replaced at no cost. Contact Digilent for more details.

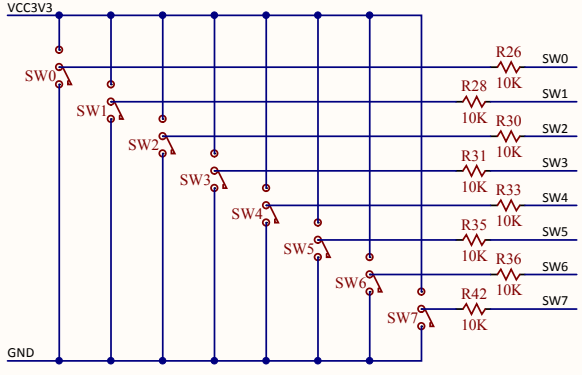
Copyright Digilent, Inc. All rights reserved.
Other product and company names mentioned may be trademarks of their respective owners.

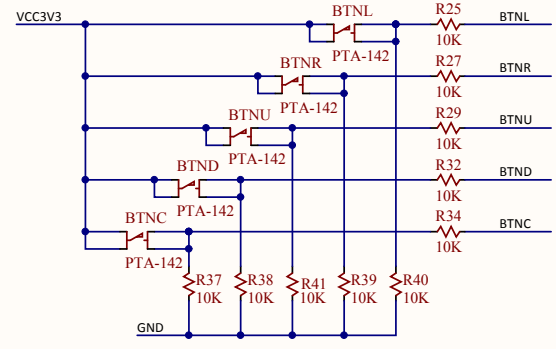


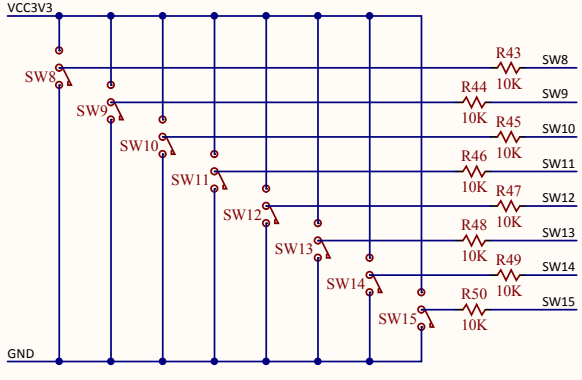










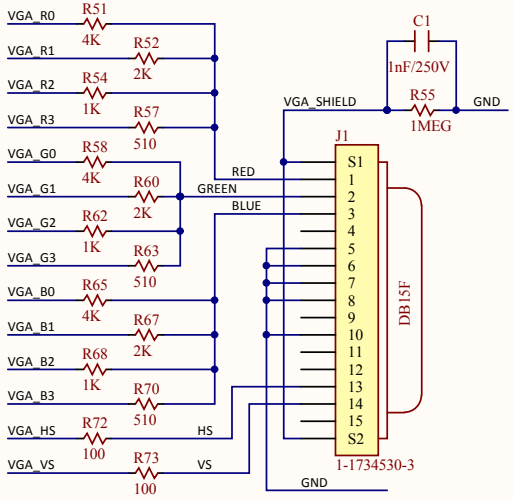


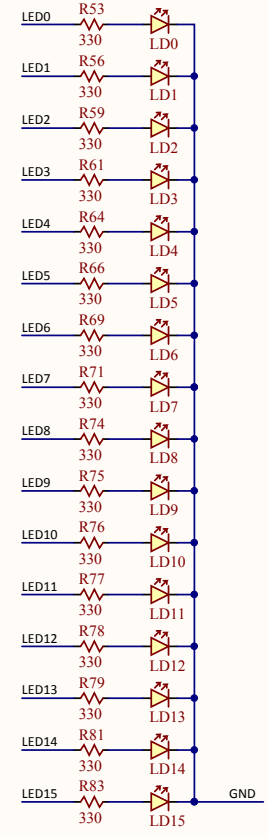
F1 Foot
F2 Foot
F3 Foot
F4 Foot

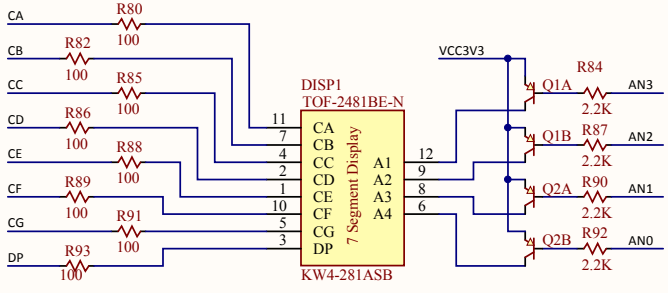
Digilent Inc. | CE | ROHS | Chinese ROHS | Basys 3 | XUP | Linear Tech

Title	Basys 3
Rev	C.0
Copyright	2014
Circuit	PMOD, I/O
Doc#	500-183
Engineer	MTA
Author	GMA
Date	5/21/2014
Sheet#	1 out of 8

DIGILENT® BEYOND THEORY

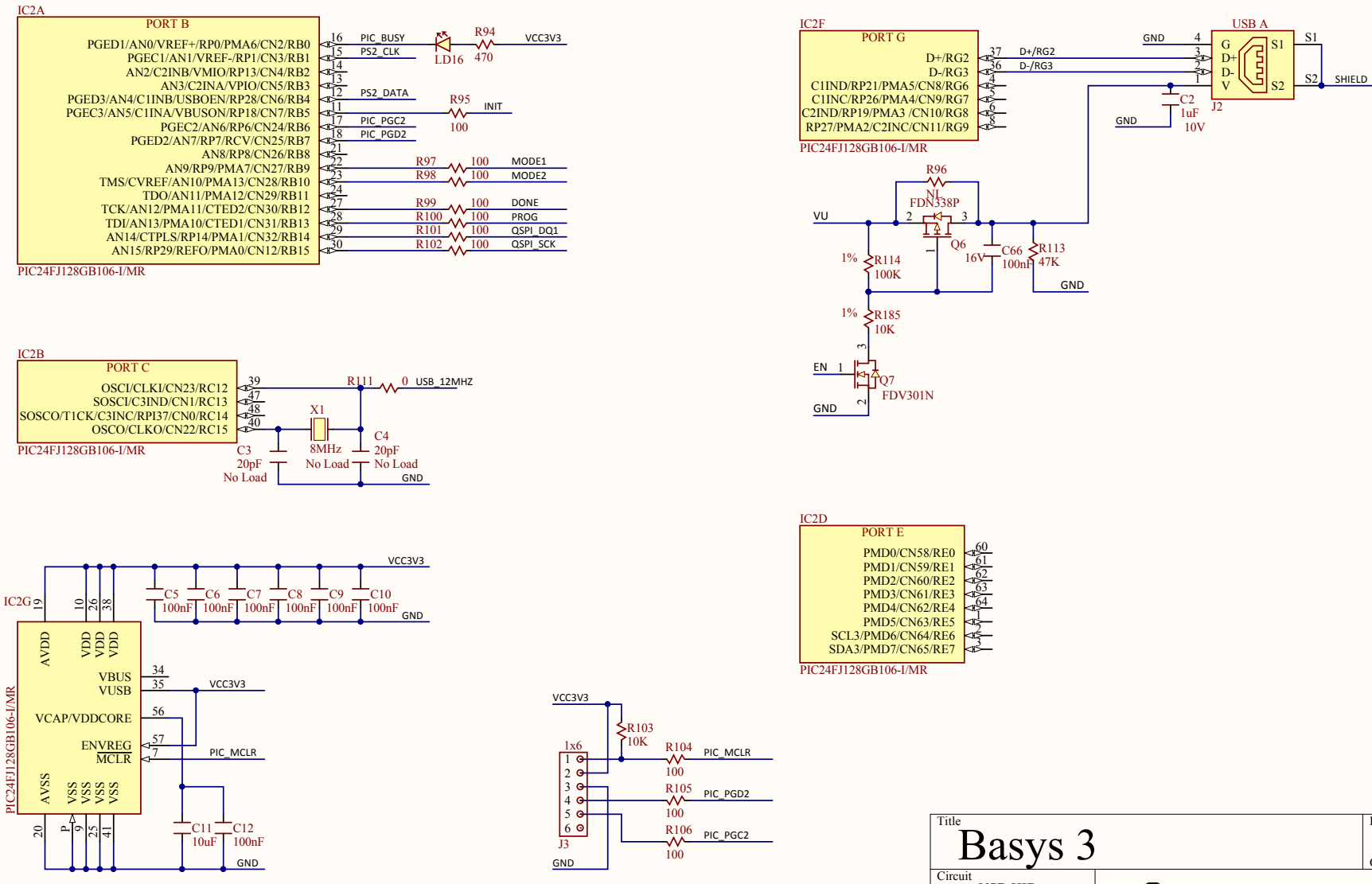


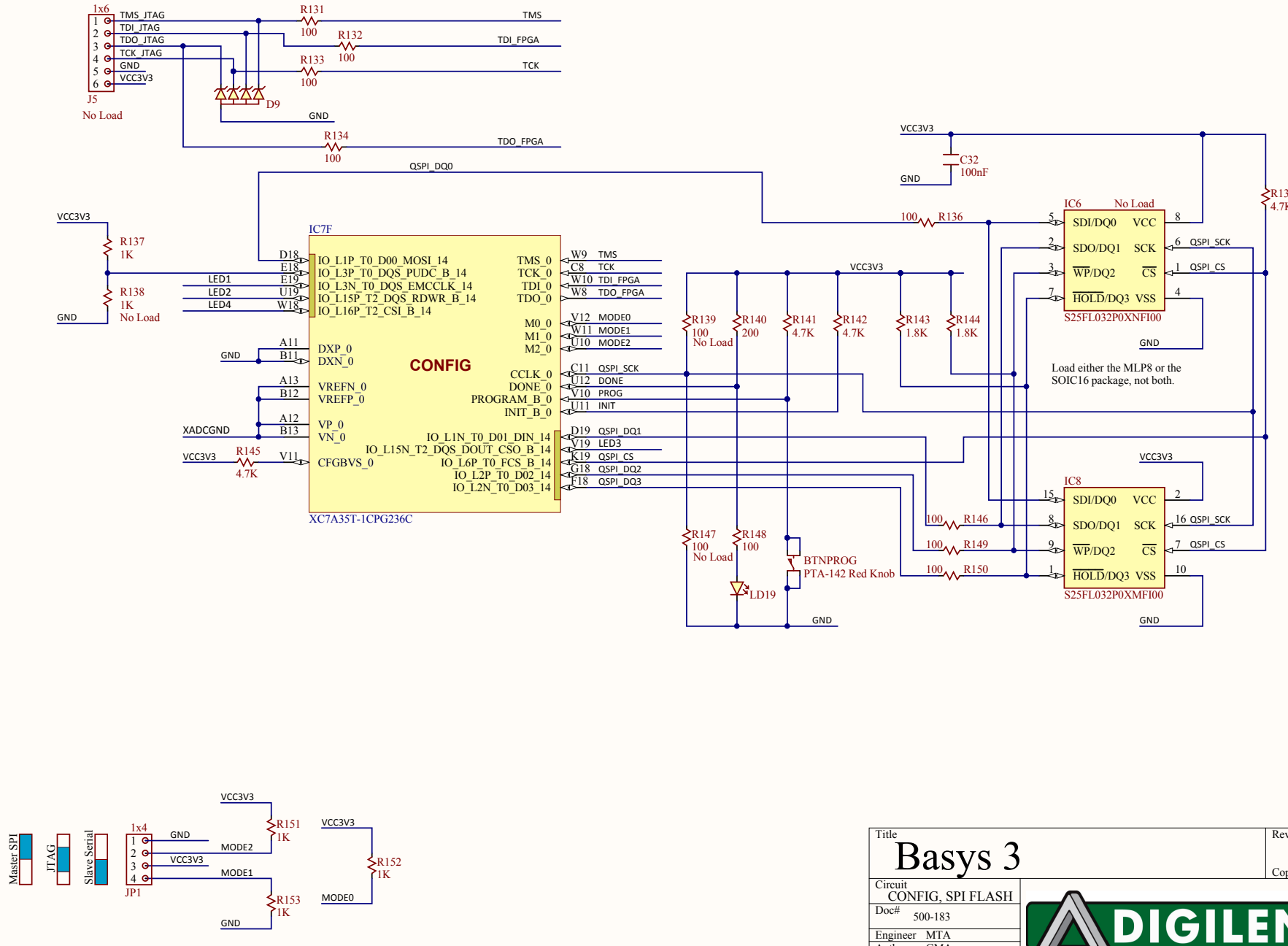




Title	Basys 3
Rev	C.0
Copyright	2014
Circuit	7 SEG, VGA, LEDs
Doc#	500-183
Engineer	MTA
Author	GMA
Date	5/21/2014
Sheet#	2 out of 8

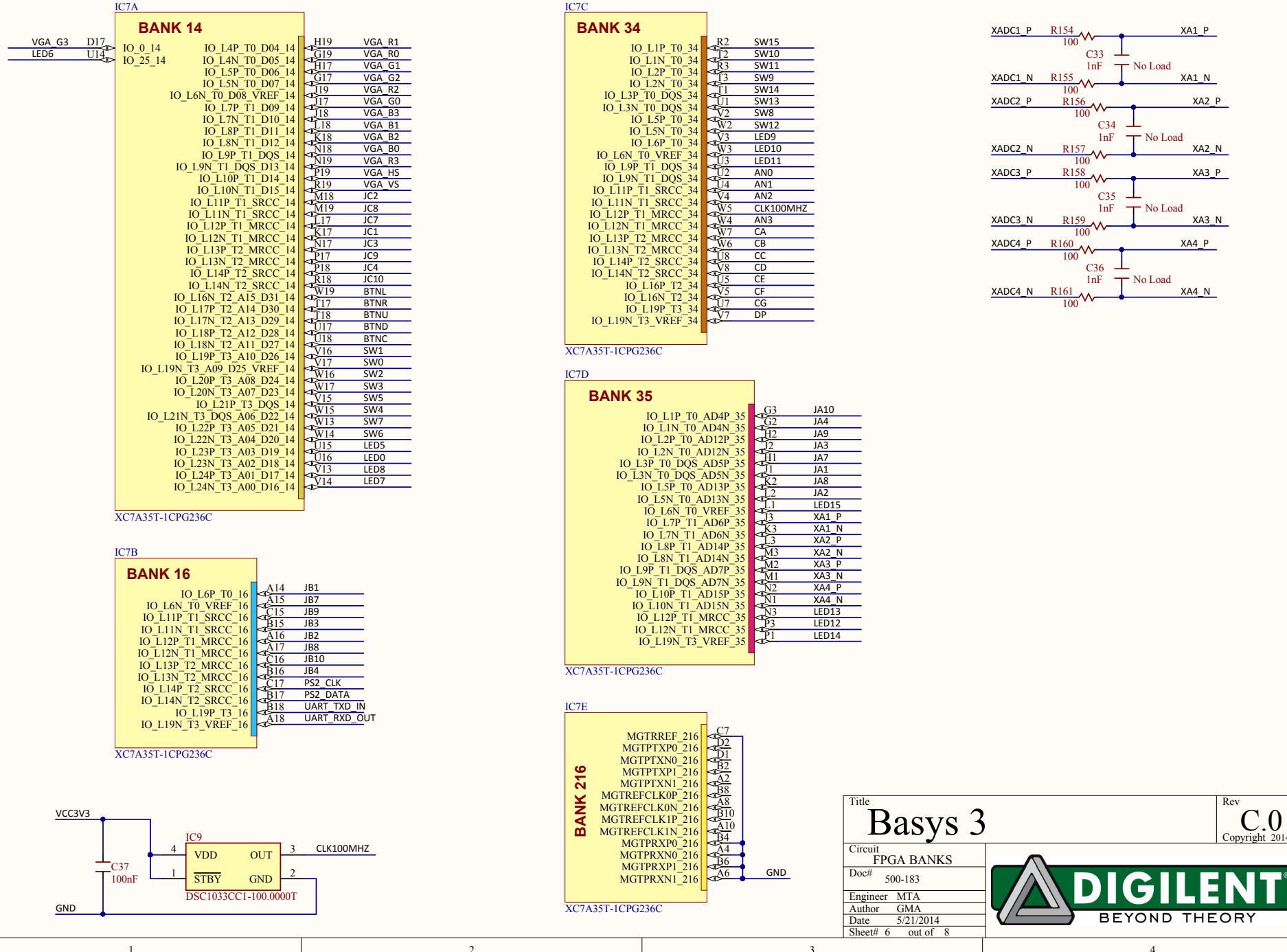
DIGILENT
BEYOND THEORY

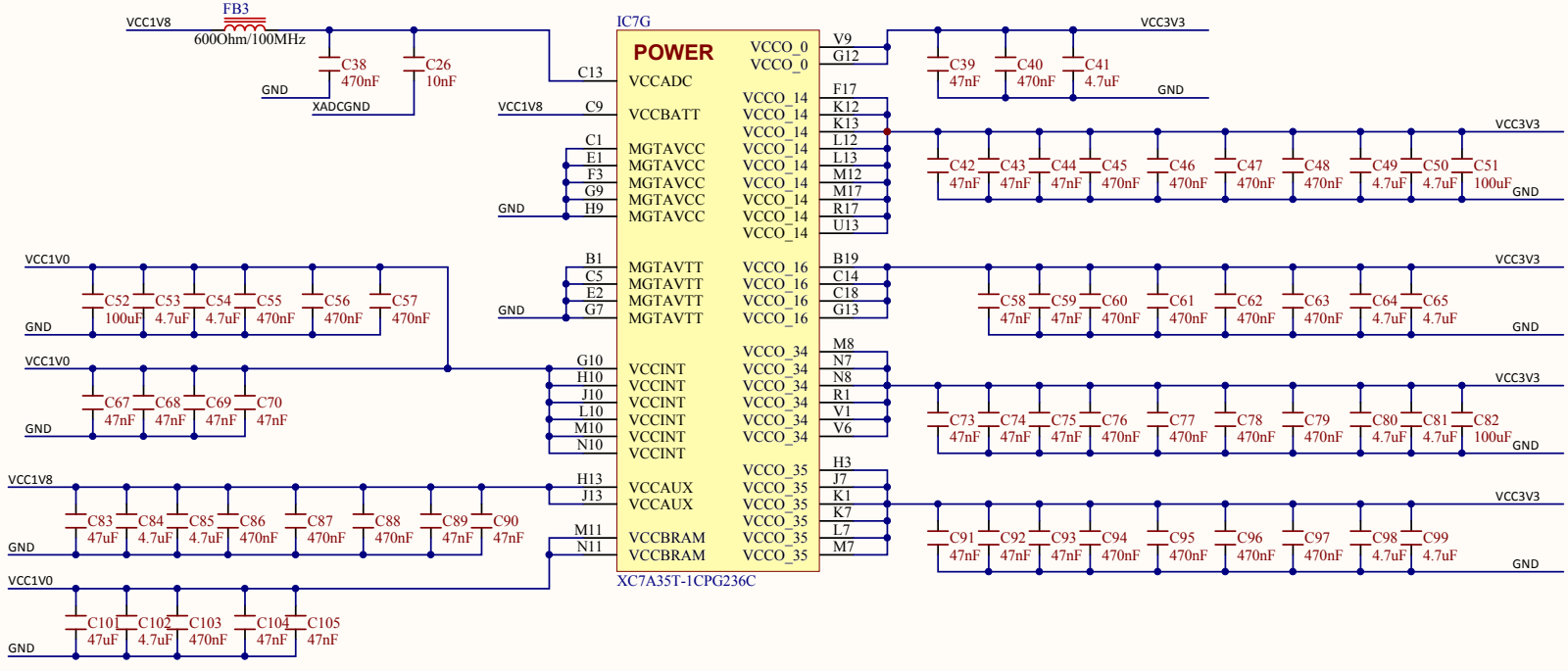

IC2A
PORT B
PGED1/AN0/VREF+/RP0/PMA6/CN2/RB0
PGEC1/AN1/VREF-/RP1/CN3/RB1
AN2/C2INB/VMIO/RP13/CN4/RB2
AN3/C2INA/VPIO/CN5/RB3
PGED3/AN4/C1INB/USBOEN/RP28/CN6/RB4
PGEC3/AN5/C1INA/VBUSON/RP18/CN7/RB5
PGEC2/AN6/RP6/CN24/RB6
PGED2/AN7/RP7/RCV/CN25/RB7
AN8/RP8/CN26/RB8
AN9/RP9/PMA7/CN27/RB9
TMS/CVREF/AN10/PMA13/CN28/RB10
TDO/AN11/PMA12/CN29/RB11
TCK/AN12/PMA11/CTED2/CN30/RB12
TDI/AN13/PMA10/CTED1/CN31/RB13
AN14/CTPLS/RP14/PMA1/CN32/RB14
AN15/RP29/REFO/PMA0/CN12/RB15
PIC_BUSY
PS2_CLK
PS2_DATA
PIC_PGC2
PIC_PGD2
LD16
R94
470
VCC3V3
R95
100
INIT
R97
R98
R99
R100
R101
R102
MODE1
MODE2
DONE
PROG
QSPI_DQ1
QSPI_SCK
PIC24FJ128GB106-I/MR
IC2F
PORT G
D+/RG2
D-/RG3
C1IND/RP21/PMA5/CN8/RG6
C1INC/RP26/PMA4/CN9/RG7
C2IND/RP19/PMA3 /CN10/RG8
RP27/PMA2/C2INC/CN11/RG9
USB A
GND
G
D+
D-
V
S1
S2
SHIELD
C2
1uF
10V
J2
R96
NI
FDN338P
Q6
VU
1%
R114
100K
C66
16V
100nF
R113
47K
R185
10K
EN
Q7
FDV301N
IC2B
PORT C
OSCI/CLKI/CN23/RC12
SOSCI/C3IND/CN1/RC13
SOSCO/T1CK/C3INC/RPI37/CN0/RC14
OSCO/CLKO/CN22/RC15
R111
0
USB_12MHZ
X1
8MHz
No Load
C3
20pF
C4
IC2G
AVDD
VDD
VBUS
VUSB
VCAP/VDDCORE
ENVREG
MCLR
AVSS
VSS
C5
C6
C7
C8
C9
C10
100nF
C11
10uF
C12
IC2D
PORT E
PMD0/CN58/RE0
PMD1/CN59/RE1
PMD2/CN60/RE2
PMD3/CN61/RE3
PMD4/CN62/RE4
PMD5/CN63/RE5
SCL3/PMD6/CN64/RE6
SDA3/PMD7/CN65/RE7
J3
1x6
R103
10K
R104
R105
R106
Title
Basys 3
Rev
C.0
Copyright 2014
Circuit
USB HID
Doc#
500-183
Engineer MTA
Author GMA
Date 5/21/2014
Sheet# 3 out of 8
DIGILENT
BEYOND THEORY

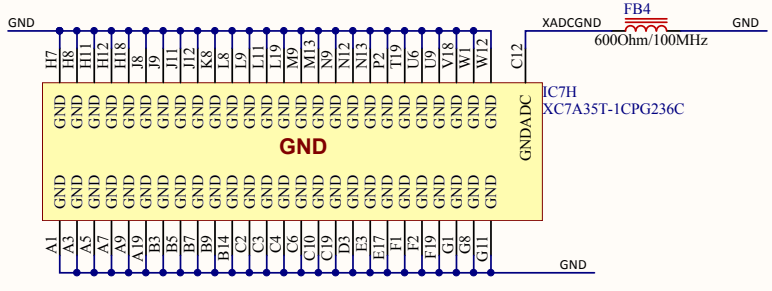


Title	Basys 3
Rev	C.0
Copyright	2014
Circuit	CONFIG, SPI FLASH
Doc#	500-183
Engineer	MTA
Author	GMA
Date	5/21/2014
Sheet#	5 out of 8

DIGILENT® BEYOND THEORY

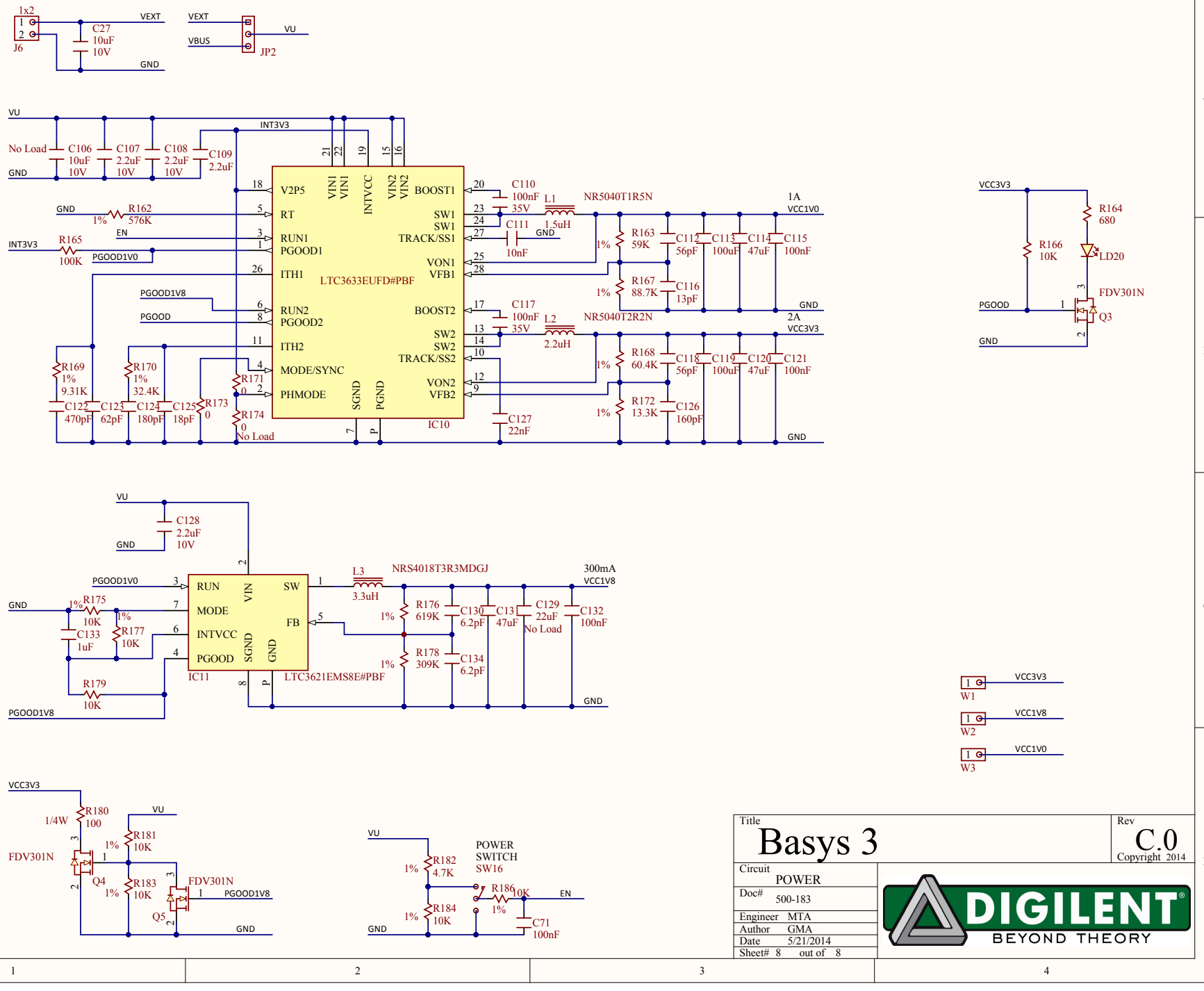

IC7A
BANK 14
VGA_G3 D17
LED6 U14
IO_0_14
IO_25_14
IO_L4P_T0_D04_14 H19 VGA_R1
IO_L4N_T0_D05_14 G19 VGA_R0
IO_L5P_T0_D06_14 H17 VGA_G1
IO_L5N_T0_D07_14 G17 VGA_G2
IO_L6N_T0_D08_VREF_14 J19 VGA_R2
IO_L7P_T1_D09_14 J17 VGA_G0
IO_L7N_T1_D10_14 J18 VGA_B3
IO_L8P_T1_D11_14 L18 VGA_B1
IO_L8N_T1_D12_14 K18 VGA_B2
IO_L9P_T1_DQS_14 N18 VGA_B0
IO_L9N_T1_DQS_D13_14 N19 VGA_R3
IO_L10P_T1_D14_14 P19 VGA_HS
IO_L10N_T1_D15_14 R19 VGA_VS
IO_L11P_T1_SRCC_14 M18 JC2
IO_L11N_T1_SRCC_14 M19 JC8
IO_L12P_T1_MRCC_14 L17 JC7
IO_L12N_T1_MRCC_14 K17 JC1
IO_L13P_T2_MRCC_14 N17 JC3
IO_L13N_T2_MRCC_14 P17 JC9
IO_L14P_T2_SRCC_14 P18 JC4
IO_L14N_T2_SRCC_14 R18 JC10
IO_L16N_T2_A15_D31_14 W19 BTNL
IO_L17P_T2_A14_D30_14 T17 BTNR
IO_L17N_T2_A13_D29_14 T18 BTNU
IO_L18P_T2_A12_D28_14 U17 BTND
IO_L18N_T2_A11_D27_14 U18 BTNC
IO_L19P_T3_A10_D26_14 V16 SW1
IO_L19N_T3_A09_D25_VREF_14 V17 SW0
IO_L20P_T3_A08_D24_14 W16 SW2
IO_L20N_T3_A07_D23_14 W17 SW3
IO_L21P_T3_DQS_14 V15 SW5
IO_L21N_T3_DQS_A06_D22_14 W15 SW4
IO_L22P_T3_A05_D21_14 W13 SW7
IO_L22N_T3_A04_D20_14 W14 SW6
IO_L23P_T3_A03_D19_14 U15 LED5
IO_L23N_T3_A02_D18_14 U16 LED0
IO_L24P_T3_A01_D17_14 V13 LED8
IO_L24N_T3_A00_D16_14 V14 LED7
XC7A35T-1CPG236C
IC7B
BANK 16
IO_L6P_T0_16 A14 JB1
IO_L6N_T0_VREF_16 A15 JB7
IO_L11P_T1_SRCC_16 C15 JB9
IO_L11N_T1_SRCC_16 B15 JB3
IO_L12P_T1_MRCC_16 A16 JB2
IO_L12N_T1_MRCC_16 A17 JB8
IO_L13P_T2_MRCC_16 C16 JB10
IO_L13N_T2_MRCC_16 B16 JB4
IO_L14P_T2_SRCC_16 C17 PS2_CLK
IO_L14N_T2_SRCC_16 B17 PS2_DATA
IO_L19P_T3_16 B18 UART_TXD_IN
IO_L19N_T3_VREF_16 A18 UART_RXD_OUT
XC7A35T-1CPG236C
VCC3V3
C37
100nF
GND
IC9
VDD
STBY
OUT
GND
4
1
3
2
CLK100MHZ
DSC1033CC1-100.0000T
IC7C
BANK 34
IO_L1P_T0_34 R2 SW15
IO_L1N_T0_34 T2 SW10
IO_L2P_T0_34 R3 SW11
IO_L2N_T0_34 T3 SW9
IO_L3P_T0_DQS_34 T1 SW14
IO_L3N_T0_DQS_34 U1 SW13
IO_L5P_T0_34 V2 SW8
IO_L5N_T0_34 W2 SW12
IO_L6P_T0_34 V3 LED9
IO_L6N_T0_VREF_34 W3 LED10
IO_L9P_T1_DQS_34 U3 LED11
IO_L9N_T1_DQS_34 U2 AN0
IO_L11P_T1_SRCC_34 U4 AN1
IO_L11N_T1_SRCC_34 V4 AN2
IO_L12P_T1_MRCC_34 W5 CLK100MHZ
IO_L12N_T1_MRCC_34 W4 AN3
IO_L13P_T2_MRCC_34 W7 CA
IO_L13N_T2_MRCC_34 W6 CB
IO_L14P_T2_SRCC_34 U8 CC
IO_L14N_T2_SRCC_34 V8 CD
IO_L16P_T2_34 U5 CE
IO_L16N_T2_34 V5 CF
IO_L19P_T3_34 U7 CG
IO_L19N_T3_VREF_34 V7 DP
XC7A35T-1CPG236C
IC7D
BANK 35
IO_L1P_T0_AD4P_35 G3 JA10
IO_L1N_T0_AD4N_35 G2 JA4
IO_L2P_T0_AD12P_35 H2 JA9
IO_L2N_T0_AD12N_35 J2 JA3
IO_L3P_T0_DQS_AD5P_35 H1 JA7
IO_L3N_T0_DQS_AD5N_35 J1 JA1
IO_L5P_T0_AD13P_35 K2 JA8
IO_L5N_T0_AD13N_35 L2 JA2
IO_L6N_T0_VREF_35 L1 LED15
IO_L7P_T1_AD6P_35 J3 XA1_P
IO_L7N_T1_AD6N_35 K3 XA1_N
IO_L8P_T1_AD14P_35 L3 XA2_P
IO_L8N_T1_AD14N_35 M3 XA2_N
IO_L9P_T1_DQS_AD7P_35 M2 XA3_P
IO_L9N_T1_DQS_AD7N_35 M1 XA3_N
IO_L10P_T1_AD15P_35 N2 XA4_P
IO_L10N_T1_AD15N_35 N1 XA4_N
IO_L12P_T1_MRCC_35 N3 LED13
IO_L12N_T1_MRCC_35 P3 LED12
IO_L19N_T3_VREF_35 P1 LED14
XC7A35T-1CPG236C
IC7E
BANK 216
MGTRREF_216 C7
MGTPTXP0_216 D2
MGTPTXN0_216 D1
MGTPTXP1_216 B2
MGTPTXN1_216 A2
MGTREFCLK0P_216 B8
MGTREFCLK0N_216 A8
MGTREFCLK1P_216 B10
MGTREFCLK1N_216 A10
MGTPRXP0_216 B4
MGTPRXN0_216 A4
MGTPRXP1_216 B6
MGTPRXN1_216 A6
GND
XC7A35T-1CPG236C
XADC1_P R154 100 XA1_P
C33 1nF No Load
XADC1_N R155 100 XA1_N
XADC2_P R156 100 XA2_P
C34 1nF No Load
XADC2_N R157 100 XA2_N
XADC3_P R158 100 XA3_P
C35 1nF No Load
XADC3_N R159 100 XA3_N
XADC4_P R160 100 XA4_P
C36 1nF No Load
XADC4_N R161 100 XA4_N
Title
Basys 3
Rev
C.0
Copyright 2014
Circuit
FPGA BANKS
Doc#
500-183
Engineer MTA
Author GMA
Date 5/21/2014
Sheet# 6 out of 8
DIGILENT
BEYOND THEORY





Title	Basys 3
Rev	C.0
Copyright	2014
Circuit	FPGA POWER
Doc#	500-183
Engineer	MTA
Author	GMA
Date	5/21/2014
Sheet#	7 out of 8

DIGILENT® BEYOND THEORY



Title	Basys 3
Rev	C.0
Copyright	2014
Circuit	POWER
Doc#	500-183
Engineer	MTA
Author	GMA
Date	5/21/2014
Sheet#	8 out of 8

DIGILENT®
BEYOND THEORY