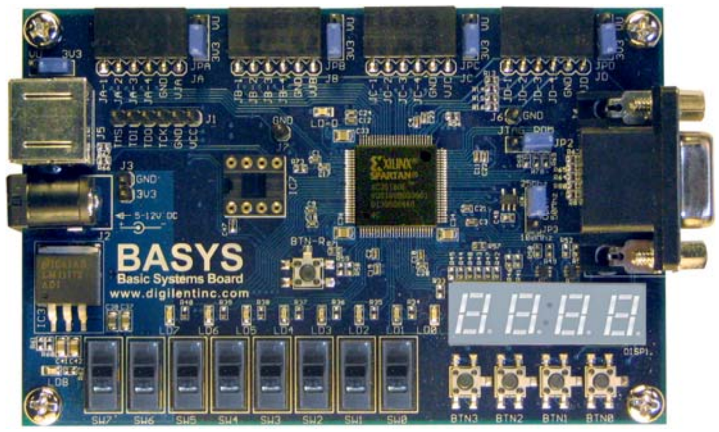
Digilent Basys Board Reference Manual

Revision: August 8, 2006

www.digilentinc.com
215 E Main Suite D | Pullman, WA 99163
(509) 334 6306 Voice and Fax

Overview



Digilent's Basys circuit board is an integrated circuit development platform based on a Xilinx Spartan 3E Field Programmable Gate Array (FPGA). The Basys board provides a low-cost, introductory development platform for engineers to try new designs or learn about FPGAs and modern CAD tools. Basys features include:

- 100,000 gate Xilinx Spartan 3E FPGA;
- JTAG programming port;
- XCF02S Xilinx Platform Flash ROM to store FPGA configurations;
- Large collection of I/O devices including eight LEDs, four-digit seven-segment display, four pushbuttons, and eight slide switches;
- PS/2 and VGA ports;
- User-selectable oscillator (25/50/100MHz), plus a socket for a second oscillator;
- Four 6-pin header ports for user I/O and peripheral module connection;
- ESD and short-circuit protection on all I/O signals.

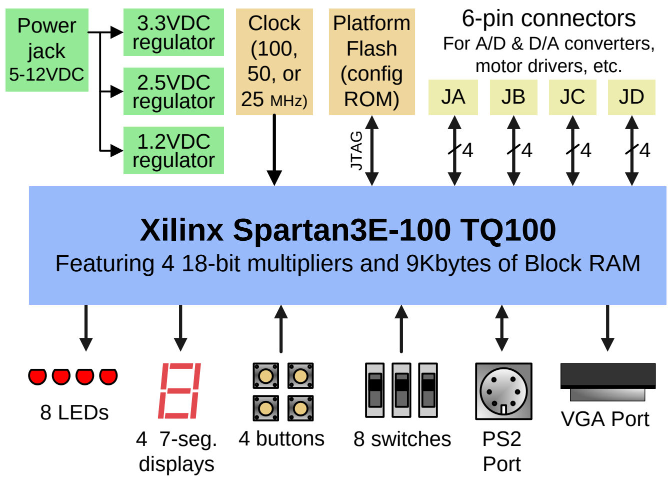

Power jack 5-12VDC
3.3VDC regulator
2.5VDC regulator
1.2VDC regulator
Clock (100, 50, or 25 MHz)
Platform Flash (config ROM)
JTAG
6-pin connectors
For A/D & D/A converters, motor drivers, etc.
JA
JB
JC
JD
4
4
4
4
Xilinx Spartan3E-100 TQ100
Featuring 4 18-bit multipliers and 9Kbytes of Block RAM
8 LEDs
4 7-seg. displays
4 buttons
8 switches
PS2 Port
VGA Port

Figure 1: Basys block diagram

Functional Description

The Basys board provides an inexpensive, robust, and easy-to-use platform that anyone can use to gain experience with FPGA devices and modern design methods. It is centered on the Spartan 3E FPGA, and it contains all needed support circuits so designs can get up and running quickly. The large collection of on-board I/O devices allow many designs to be completed without the need for any other hardware, making the Basys an ideal introductory platform for experimenting with new designs or learning about FPGAs and CAD tools. The four standard 6-pin expansion connectors allow designs to grow beyond the Basys board, either with user-designed boards or breadboards and/or peripheral module (Pmod) boards offered by Digilent. (Pmods are inexpensive analog and digital I/O modules that offer A/D conversion, D/A conversion, motor drive, sensor input, and a host of other features). Signals on each 6-pin expansion connector are protected against damage from ESD and short-circuit connections, ensuring a long operating life in any environment. The Basys board works seamlessly with all versions of the Xilinx ISE tools, including the free WebPack tools. It ships with a power supply and programming cable, so designs can be implemented immediately without the need for any additional hardware.

Device Configuration

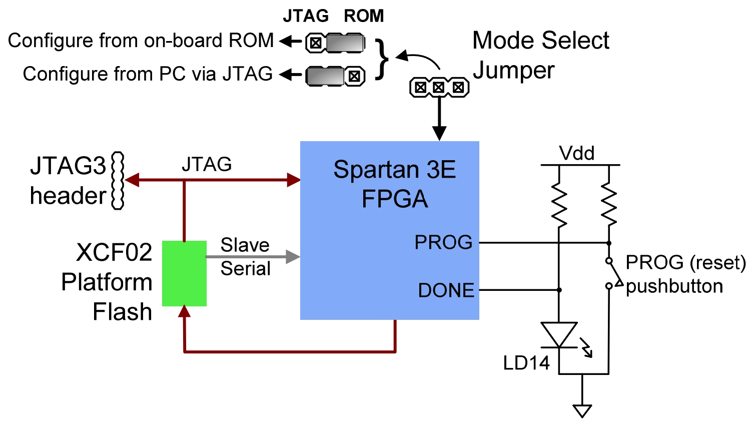
At power-on, the FPGA on the Basys board is not configured and performs no functions - it must be configured (or programmed) by the user before it can perform any functions. Design software, like the free WebPack from Xilinx, can be used to define any number of circuits that can be programmed into the FPGA. Once programmed, the FPGA will retain its configuration only as long is power is applied.

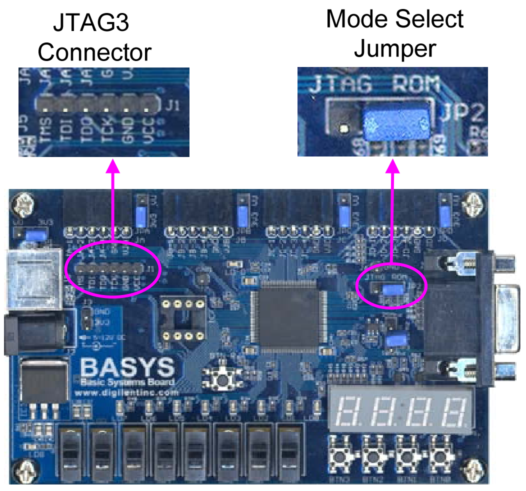
The FPGA can be programmed in two ways: directly from a PC, and from an on-board Platform Flash ROM that is also user-programmable. A jumper on the Basys board determines which source (PC or ROM) the FPGA will use to load its configuration. Note that a demonstration configuration is loaded into the Basys board during manufacturing. If that configuration has not been overwritten, it can be automatically loaded into the FPGA by setting the Mode Select Jumper to “ROM”, and cycling power or pressing the reset button (labeled “BTN_R”).

The FPGA and the Platform Flash ROM can be programmed from a PC using Digilent’s Adept software or Xilinx’s iMPACT software (both are available for free download). Digilent’s JTAG3 cable can be used to connect the board to a PC for programming with either software package.

To program the Basys board, connect the programming cable to the board and to a PC, and apply power to the Basys board. Start the programming software, and wait for the FPGA and the Platform Flash ROM to be automatically identified. To program the FPGA, select the desired .bit file; to program the Platform Flash, select the desired .mcs file. Right-click on the device to be programmed, and select the “program” function. The configuration file will be sent to the FPGA or Platform Flash, and the software will indicate whether programming was successful.

Both the FPGA and Platform Flash ROM will always appear in the scan chain. After the Platform Flash ROM has been loaded with a configuration file, the FPGA can automatically load that file at power-on if programming mode control jumper is loaded in the ROM position.





A reset button is provided (labeled “BTN_R”) that can erase the configuration in the FPGA, and start a new programming cycle. An LED labeled “LD_D” will illuminate whenever the FPGA has been successfully configured.

Power Supply

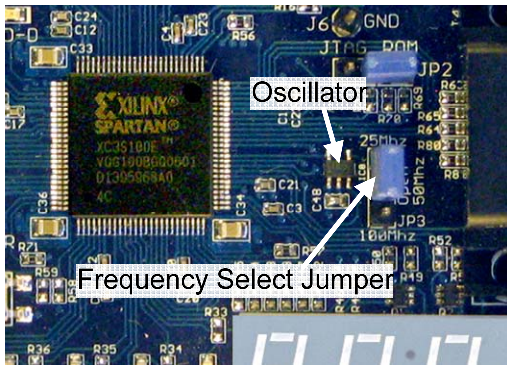
The Basys board can be powered from any DC supply that produces a voltage in the 5VDC-12VDC range. The power jack on the Basys board requires a center-positive, 2.1mm power supply connector as is commonly found on wall-plug power supplies (Basys ships with a 5VDC wall-plug supply). Voltages higher than 12V may permanently damage Basys.

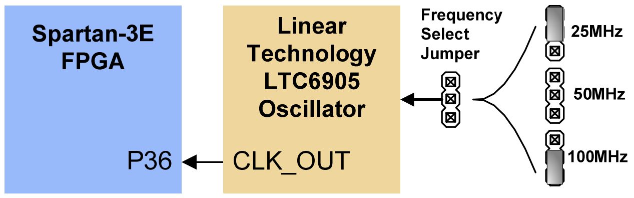
The voltage input to the power jack is routed to the four 6-pin expansion connectors and to a LM1117 voltage regulator that produces the 3.3V supply for the board. The 2.5V and 1.2V supplies required by the FPGA are produced by post-regulating the 3.3V supply. Total board current is dependant on FPGA configuration, clock frequency, and external connections. In test circuits with roughly 20K gates routed, a 50MHz clock source, and all LEDs illuminated, about 200mA of current is drawn from the 1.2V supply, 50mA from the 2.5V supply, and 100mA from the 3.3V supply. Required current will increase if larger circuits are configured in the FPGA, and if peripheral boards are attached.

The Basys board uses a four layer PCB, with the inner layers dedicated to VCC and GND planes. The FPGA and the other ICs on the board all have a large complement of bypass capacitors placed as close as possible to each VCC pin. The power supply routing and bypass capacitors result in a very clean, low-noise power supply.

Oscillators

The Basys board includes a primary, user-settable silicon oscillator that produces 25MHz, 50MHz, or 100MHz based on the position of the clock select jumper at JP3. A socket for a second oscillator is also provided at IC7 (the IC7 socket can accommodate any 3.3V CMOS oscillator in a half-size DIP package). The primary and secondary oscillators are connected to global clock input pins at pin 36 and pin 84 respectively. Both clock inputs can drive the clock synthesizer DLL on the Spartan 3E, allowing for a wide range if internal frequencies, from 4 times the input frequency to any integer divisor of the input frequency.



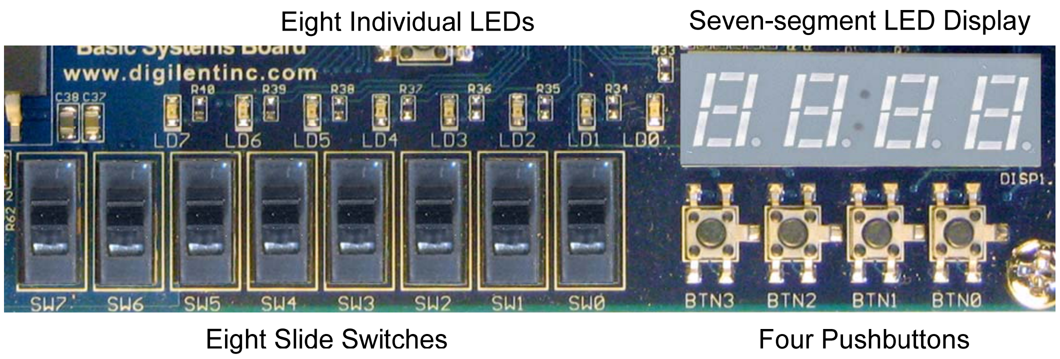


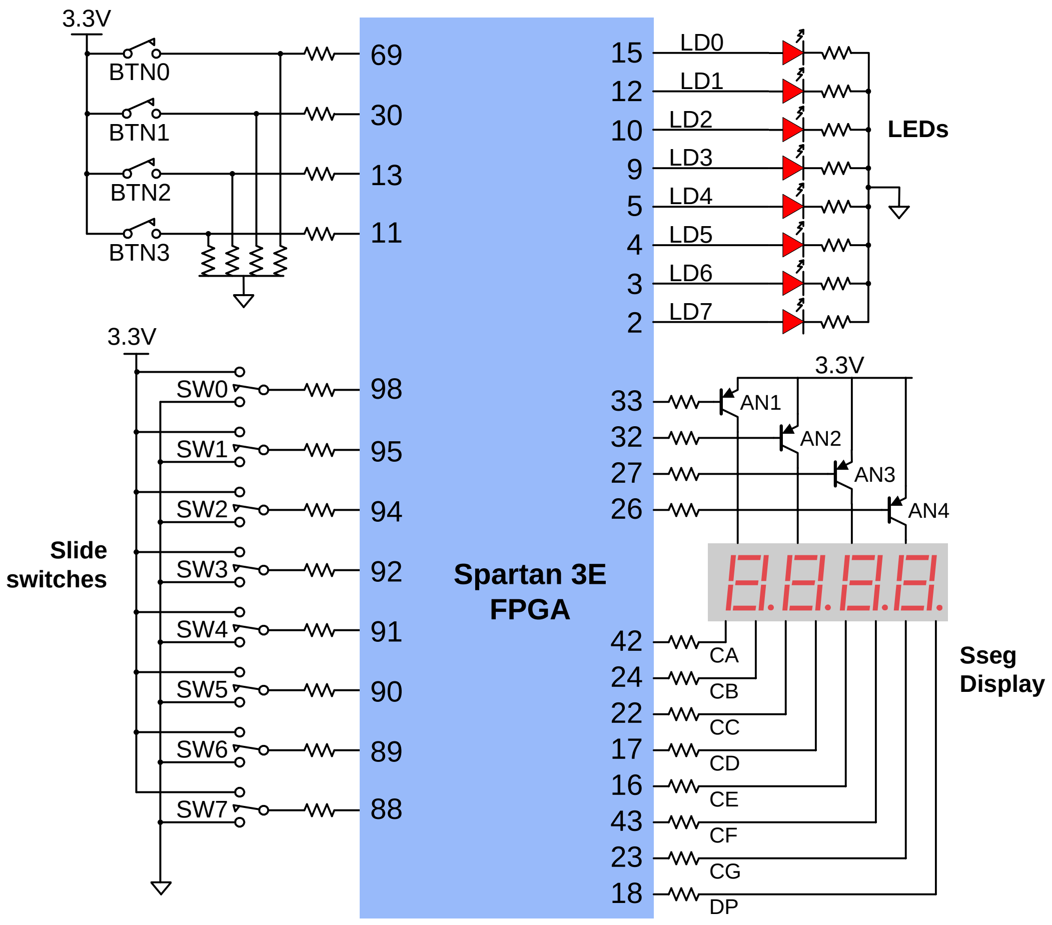
User I/O

The Basys board includes several Input and Output devices, and several data ports so that many designs can be implemented without the need for any other components.

Inputs: Slide Switches and Pushbuttons

Four pushbuttons and eight slide switches are provided for circuit inputs. Pushbutton inputs are normally low, and they are driven high only when the pushbutton is pressed. Slide switches generate constant high or low inputs depending on their position. Pushbutton and slide switch inputs use a series resistor for protection against short circuits (a short circuit would occur if an FPGA pin assigned to a pushbutton or slide switch was inadvertently defined as an output).


Eight Individual LEDs
Seven-segment LED Display
Eight Slide Switches
Four Pushbuttons
LD7 LD6 LD5 LD4 LD3 LD2 LD1 LD0
SW7 SW6 SW5 SW4 SW3 SW2 SW1 SW0
BTN3 BTN2 BTN1 BTN0
DISP1



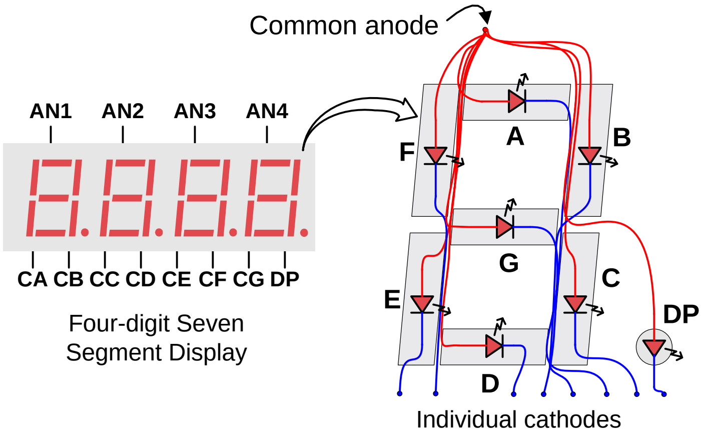
Outputs: LEDs

Eight LEDs are provided for circuit outputs. LED anodes are driven from the FPGA via 390-ohm resistors, so a logic '1' output will illuminate them with 3-4ma of drive current. A ninth LED is provided as a power-on LED, and a tenth LED indicates FPGA programming status.

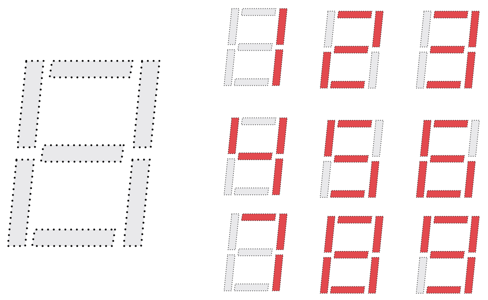
Outputs: Seven-Segment Display

The Basys board contains a 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. Of these 128 possible patterns, the ten corresponding to the decimal digits are the most useful.

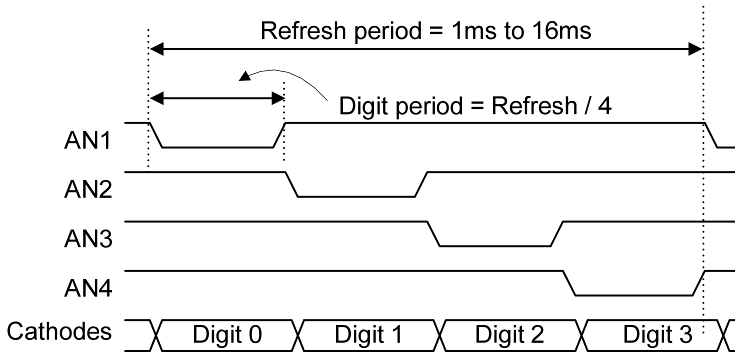
The anodes of the seven LEDs forming each digit are tied together into one "common anode" circuit node, but the LED cathodes remain separate. The common anode signals are available as four "digit enable" input signals to the 4-digit display. The cathodes of similar segments on all four displays are connected into seven circuit nodes labeled CA through CG (so, 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.



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 respond. Each digit is illuminated just one-quarter 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 a given point (around 45 hertz), then most people will begin to see the display flicker.


An un-illuminated seven-segment display, and nine illumination patterns corresponding to decimal digits

In order 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 1KHz to 60Hz). For example, in a 60Hz refresh scheme, the entire display would be refreshed once every 16ms, and each digit would be illuminated for ¼ of the refresh cycle, or 4ms. The controller must assure that the correct cathode pattern is present when the corresponding anode signal is driven. To illustrate the process, if AN1 is asserted while CB and CC are asserted, then a "1" will be displayed in digit position 1. Then, if AN2 is asserted while CA, CB and CC are asserted, then a "7" will be displayed in digit position 2. If A1 and CB, CC are driven for 4ms, and then A2 and CA, CB, CC are driven for 4ms in an endless succession, the

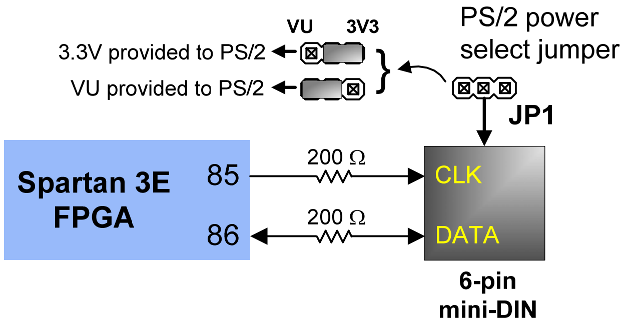


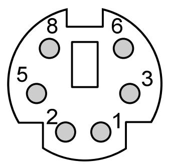
display will show “17” in the first two digits. An example timing diagram for a four-digit controller is provided.

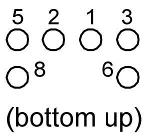
Ports and External Connectors

PS/2 Port

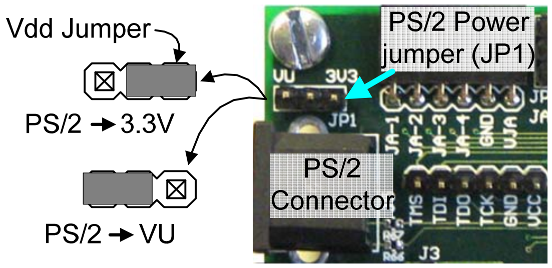
The Basys board includes a 6-pin mini-DIN connector that can accommodate a PS/2 mouse or keyboard connection. Most PS/2 devices can operate from a 3.3V supply, but some devices, like older keyboards, require a 5VDC supply. A jumper on the Basys board (JP1) can be used to select whether 3.3V or VU is supplied to the PS/2 Vdd pin. If 5V is required, set JP1 to VU and ensure that Basys is powered with a 5VDC wall-plug supply (note that Basys ships with a 5VDC wall-plug supply).





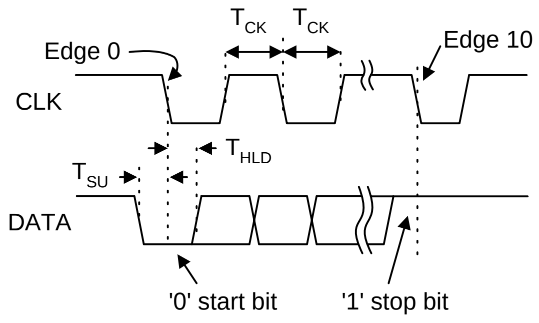


Pin1:	Data
Pin2:	Data
Pin3:	GND
Pin5:	Vdd
Pin6:	Clock
Pin8:	Clock



PS/2 connector pin locations, pin assignments, and jumper settings

Both the mouse and keyboard use a two-wire serial bus (clock and data) to communicate with a host device. Both use 11-bit words that include a start, stop and odd parity 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 the figure. The clock and data signals are only driven when data transfers occur, and otherwise they are held in the “idle” state at logic ‘1’. The timings define signal requirements for mouse-to-host communications and bi-directional keyboard communications.



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

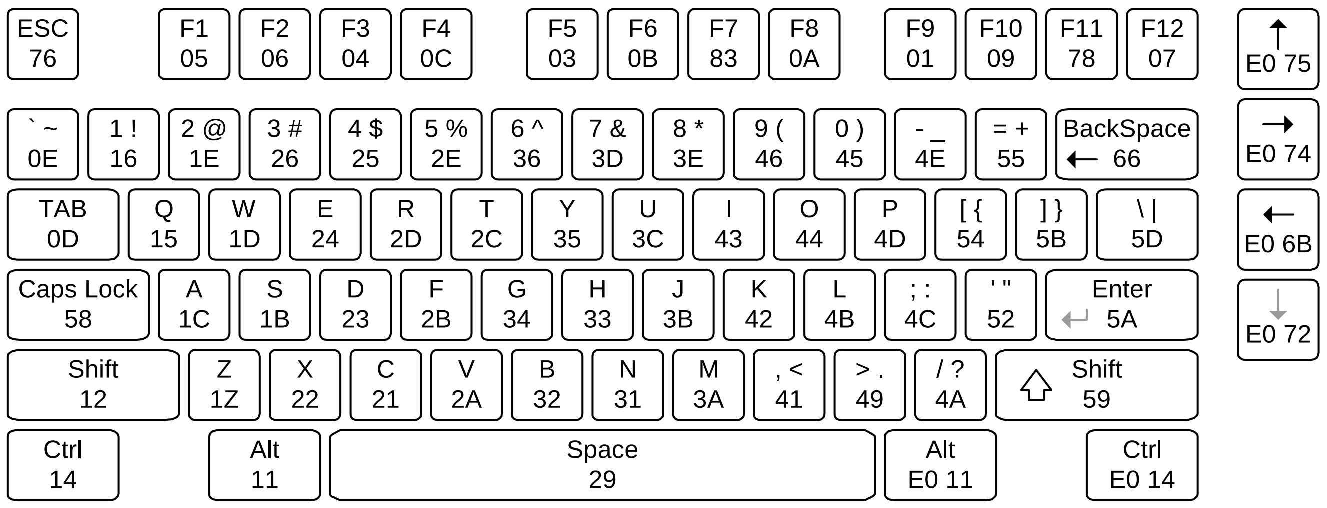
Keyboard

The keyboard uses open-collector drivers so that either 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).

PS2-style keyboards use scan codes to communicate key press data (nearly all keyboards in use today are PS2 style). Each key has a single, unique scan code that is sent whenever the corresponding key is pressed. If the key is pressed and held, the scan code will be sent repeatedly about once every 100ms. When a key is released, a “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 original scan code, and the host device must determine which 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 the figure below.

A host device can also send data to the keyboard. Below is a short list of some often-used commands.

ED	Set Num Lock, Caps Lock, and Scroll Lock LEDs. After receiving an "ED", the keyboard returns an "FA"; then the 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. Upon receiving an echo command, the keyboard replies with "EE".
F3	Set scan code repeat rate. The keyboard acknowledges receipt of an "F3" by returning an "FA", after which the host sends a second byte to set the repeat rate.
FE	Resend. Upon receiving FE, the keyboard re-sends the last scan code sent.
FF	Reset. Resets the keyboard.



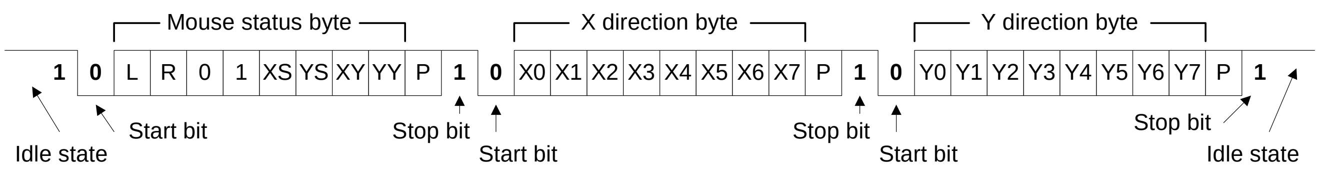
The keyboard should send data to the host only when both the data and clock lines are high (or idle). Since the host is the "bus master", the keyboard should check to see whether the host is sending data before driving the bus. To facilitate this, the clock line can be used as a "clear to send" signal. If the host pulls the clock line low, the keyboard must not send any data until the clock is released (host-to-keyboard data transmission will not be dealt with further here).

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 around 20 - 30KHz) when the data is sent, and data is valid on the falling edge of the clock.

Mouse

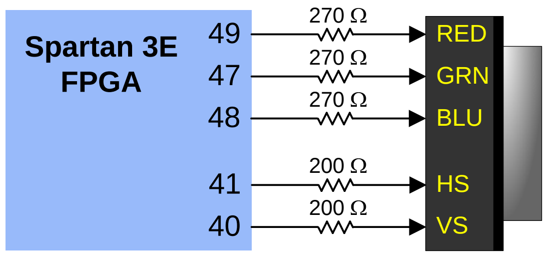
The mouse outputs a clock and data signal when it is moved; otherwise, these signals remain at logic '1'. Each time the mouse is moved, three 11-bit words are sent from the mouse to the host device. Each of the 11-bit words contains a '0' start bit, followed by 8 bits of data (LSB first), followed by 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 below. Data is valid at the falling edge of the clock, and the clock period is 20 to 30KHz.

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).

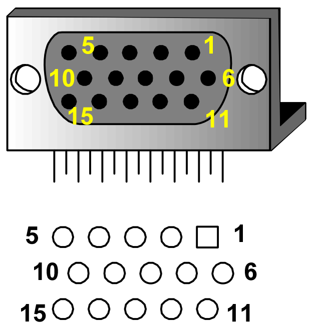


VGA Port

The five standard VGA signals Red, Green, Blue, Horizontal Sync (HS), and Vertical Sync (VS) are routed directly from the FPGA to the VGA connector, resulting in a simple video system that can produce eight colors. The color signals from the FPGA include 270-ohm series resistors that form a divider with the 75-ohm termination resistance of the VGA display. This simple circuit ensures that the video signals cannot exceed the VGA-specified maximum voltage, and results in color signals that are either fully on (.7V) or fully off (0V).



Basys VGA Circuit Diagram



VGA pin assignments			
Pin	**Signal**	**Pin**	**Signal**
1	Red	9	NC
2	Green	10	GND
3	Blue	11	NC
4	NC	12	NC
5	GND	13	HS
6	GND	14	VS
7	GND	15	NC
8	GND		

VGA signal color mapping			
Signal / Color	Red	Green	Blue
Black	0	0	0
Blue	0	0	1
Green	0	1	0
Cyan	0	1	1
Red	1	0	0
Purple	1	0	1
Yellow	1	1	0
White	1	1	1

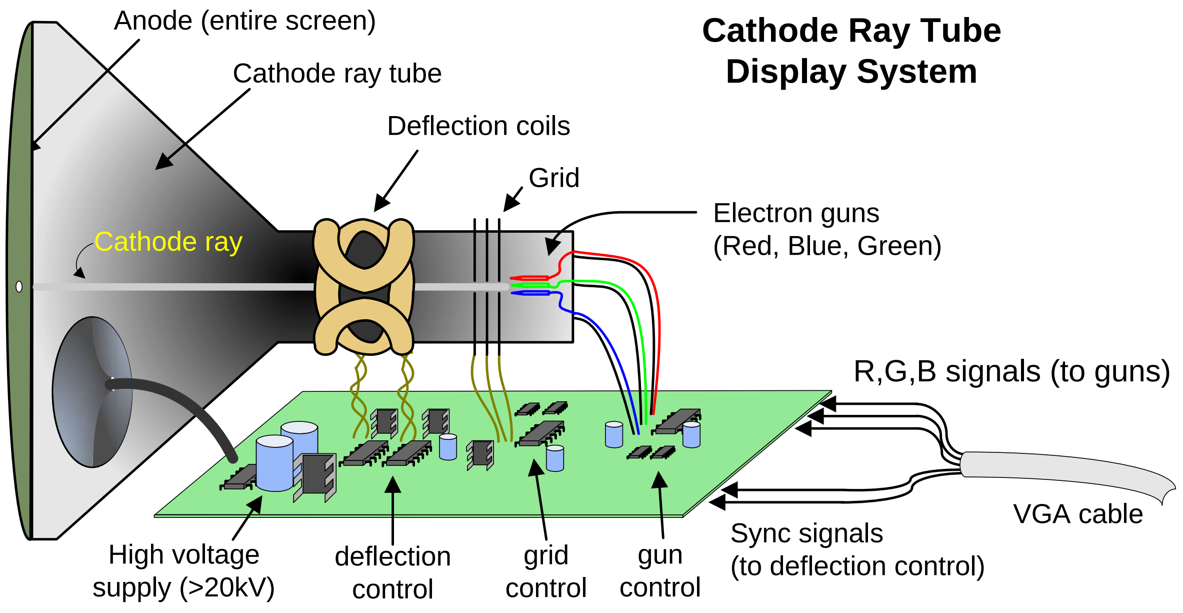
HD DB-15 connector, PCB hole pattern, pin assignments, and color-signal mapping

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

VGA System Timing

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 illustration). 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.

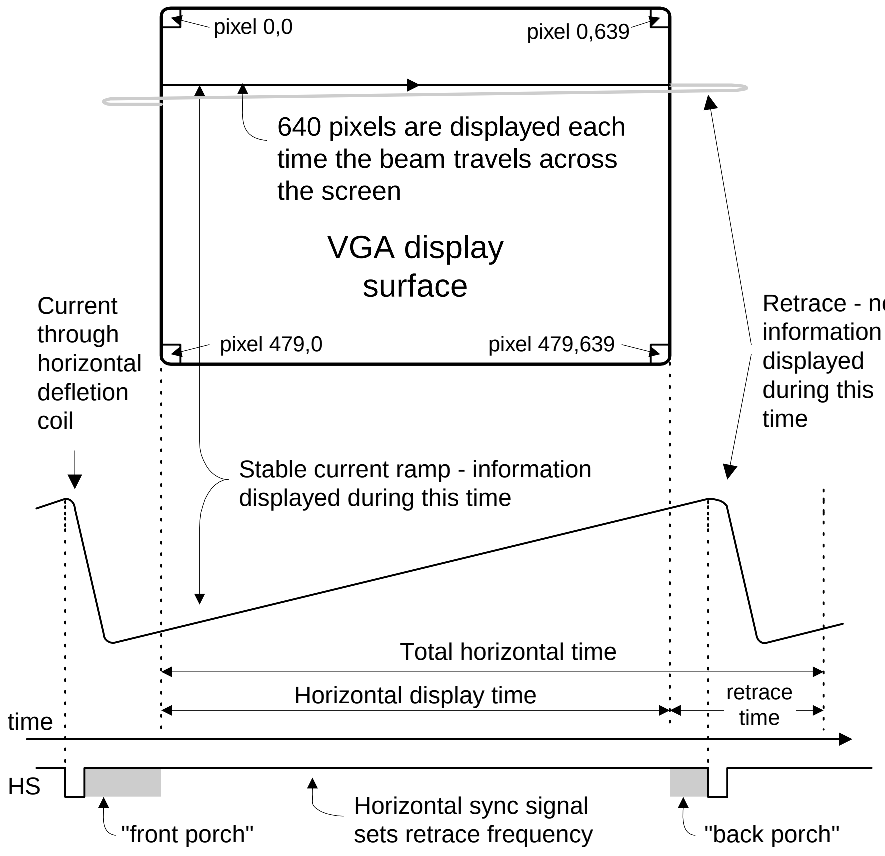


Between the grid and the display surface, the beam passes through the neck of the CRT where two coils of wire produce orthogonal electromagnetic fields. Because cathode rays are composed of charged particles (electrons), they can be deflected by these magnetic fields. Current waveforms are passed through 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 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.

Video data typically comes from a video refresh memory, with one or more bytes assigned to each pixel location (the Basys uses three 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.

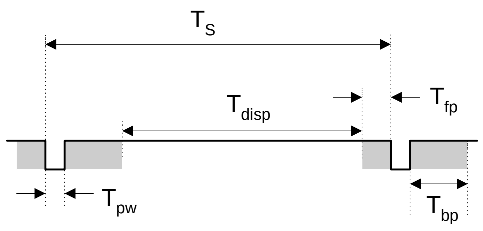
A VGA controller circuit must generate the HS and VS timings signals and coordinate the delivery of video data based on the pixel clock. The pixel clock defines the time available to display one pixel of information. The VS signal defines the “refresh” frequency of the display, or the frequency at which 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 25MHz pixel clock and 60 +/- 1Hz refresh, the signal timings shown in the table below 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.

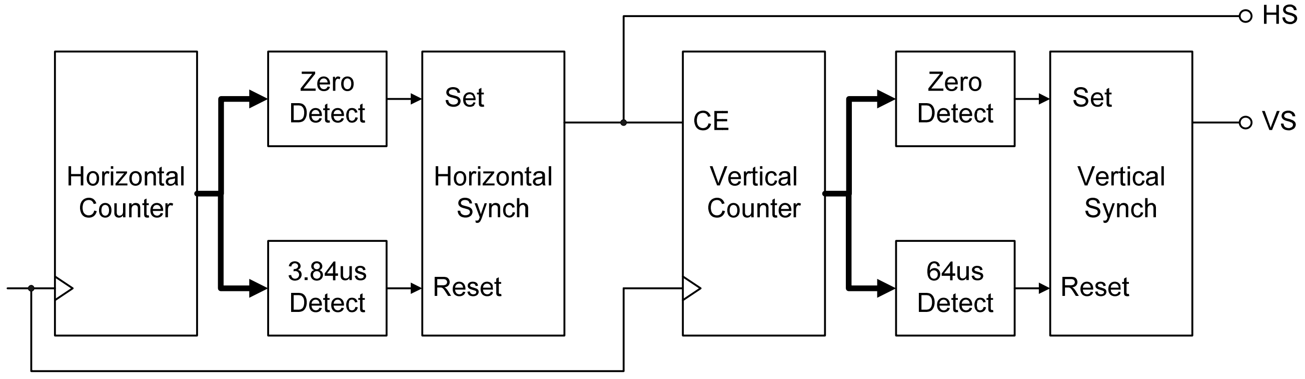

pixel 0,0
pixel 0,639
640 pixels are displayed each time the beam travels across the screen
VGA display surface
pixel 479,0
pixel 479,639
Current through horizontal defletion coil
Retrace - no information displayed during this time
Stable current ramp - information displayed during this time
Total horizontal time
Horizontal display time
retrace time
time
HS
"front porch"
Horizontal sync signal sets retrace frequency
"back porch"

A VGA controller circuit decodes the output of a horizontal-sync counter driven by the pixel clock to generate HS signal timings. This counter can be used 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 this counter can be used 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 the designer can arrange the counters to easily form video RAM addresses, or to minimize decoding logic for sync pulse generation.

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



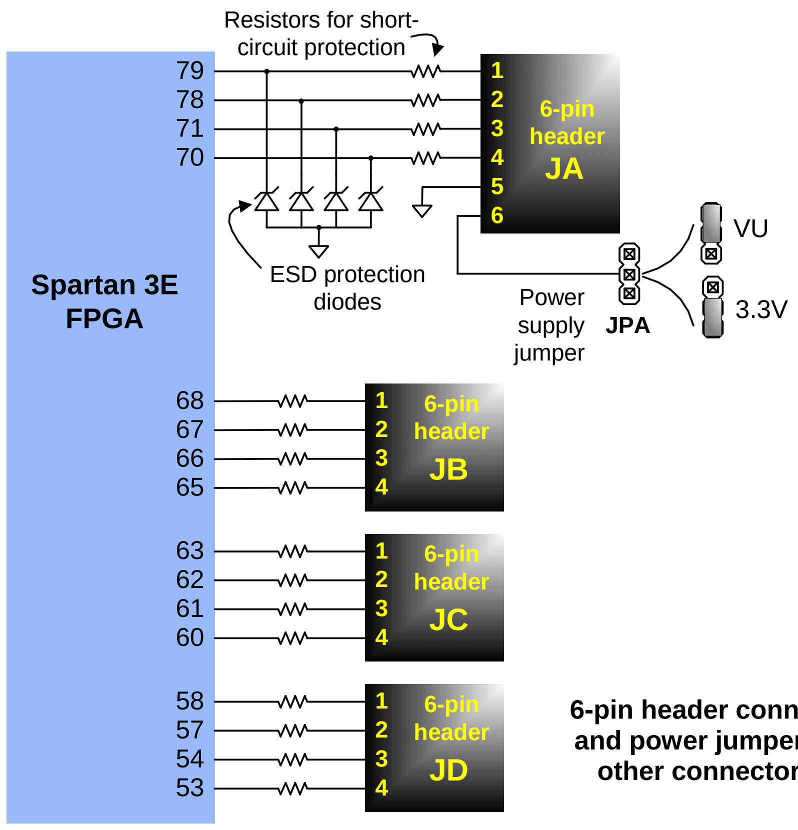


VGA controller signal timings and circuit block diagram

6-pin header connectors

The Basys board provides four 6-pin peripheral module connectors. Each connector provides Vdd, GND, and four unique FPGA signals.

Several 6-pin module boards that can attach to this connector are available from Digilent, including speaker boards, H-bridge boards, sensor boards, etc. Please see www.digilentinc.com for more information.



6-pin header connector circuit diagram. ESD diodes and power jumper shown for JA are present for the other connectors but omitted from the drawing.

FPGA

The pinout for the Spartan 3E-100 FPGA in the VQ100 package is shown in the table below.

Basys FPGA Pin Assignments

Pin	Function	Pin	Function	Pin	Function	Pin	Function	Pin	Function
1	***PROG_B***	21	V_{AUX}	41	VGA-HS	61	JC-3	81	***GND***
2	LD7	22	CC	42	MODE1/C	62	JC-2	82	***3V3***
3	LD6	23	CG	43	MODE0/C	63	JC-1	83	SPI-SS
4	LD5	24	CB	44	***D0***	64	***GND***	84	CLK2
5	LD4	25	***3V3***	45	***3V3***	65	JB-4	85	PS2C
6	V_{INT}	26	AN4	46	V_{AUX}	66	JB-3	86	PS2D
7	***GND***	27	AN3	47	VGA-G	67	JB-2	87	***GND***
8	***3V3***	28	V_{INT}	48	VGA-B	68	JB-1	88	SW7
9	LD3	29	***GND***	49	VGA-R	69	BTN0	89	SW6
10	LD2	30	BTN1	50	***CCLK***	70	JA-4	90	SW5
11	BTN3	31	***3V3***	51	V_{AUX}	71	JA-3	91	SW4
12	LD1	32	AN2	52	***GND***	72	***GND***	92	SW3
13	BTN2	33	AN1	53	JD-4	73	***3V3***	93	***GND***
14	***GND***	34	SPI-MOSI	54	JD-3	74	V_{AUX}	94	SW2
15	LD0	35	SPI-MISO	55	***3V3***	75	***TMS***	95	SW1
16	CE	36	CLK1	56	V_{INT}	76	***TDO***	96	V_{AUX}
17	CD	37	***GND***	57	JD-2	77	***TCK***	97	***3V3***
18	DP	38	SPI-SCK	58	JD-1	78	JA2	98	SW0
19	***GND***	39	***MODE2***	59	***GND***	79	JA1	99	***TMS-EN***
20	***3V3***	40	VGA-VS	60	JC-4	80	V_{INT}	100	***TDI***