1 Typical Applications

- Add USB host capability to embedded products
- Interface USB Flash drive to MCU/PLD/FPGA – data storage and firmware updates
- USB Flash drive data storage or firmware updates
- USB Flash drive to USB Flash drive file transfer interface
- Digital camera to USB Flash drive*
- PDA to USB Flash drive*
- MP3 Player to USB Flash drive or other USB slave device interface
- OSI Wireless Interface
- USB wireless process controller
- Telecom system calls logging to replace printer log
- Data logging
- Mobile phone to USB Flash drive*
- GPS to mobile phone interface
- Instrumentation USB Flash drive*
- Data-logger USB Flash drive*
- Set Top Box - USB device interface
- GPS tracker with USB Flash disk storage
- USB webcam
- Flash drive to SD Card data transfer
- Vending machine connectivity
- TLM Serial converter
- Geotagging of photos – GPS location linked to image
- Motorcycle system telemetry logging
- Medical systems
- PWM applications for motor control applications e.g. Toys
- FPGA Interfacing

* Or similar USB slave device interface e.g. USB external drive.

1.1 Part Numbers

Part Number	Package
VNC2-64L	64 Pin LQFP
VNC2-64Q	64 Pin QFN
VNC2-48L	48 Pin LQFP
VNC2-48Q	48 Pin QFN
VNC2-32L	32 Pin LQFP
VNC2-32Q	32 Pin QFN

Table 1.1 Part Numbers

Please refer to **section 11** for all package mechanical parameters.

1.2 USB Compliant

At time of writing this data sheet, VNC2 has not completed USB compliancy testing.

2 VNC2 Block Diagram

For a description of each function please refer to Section 4.

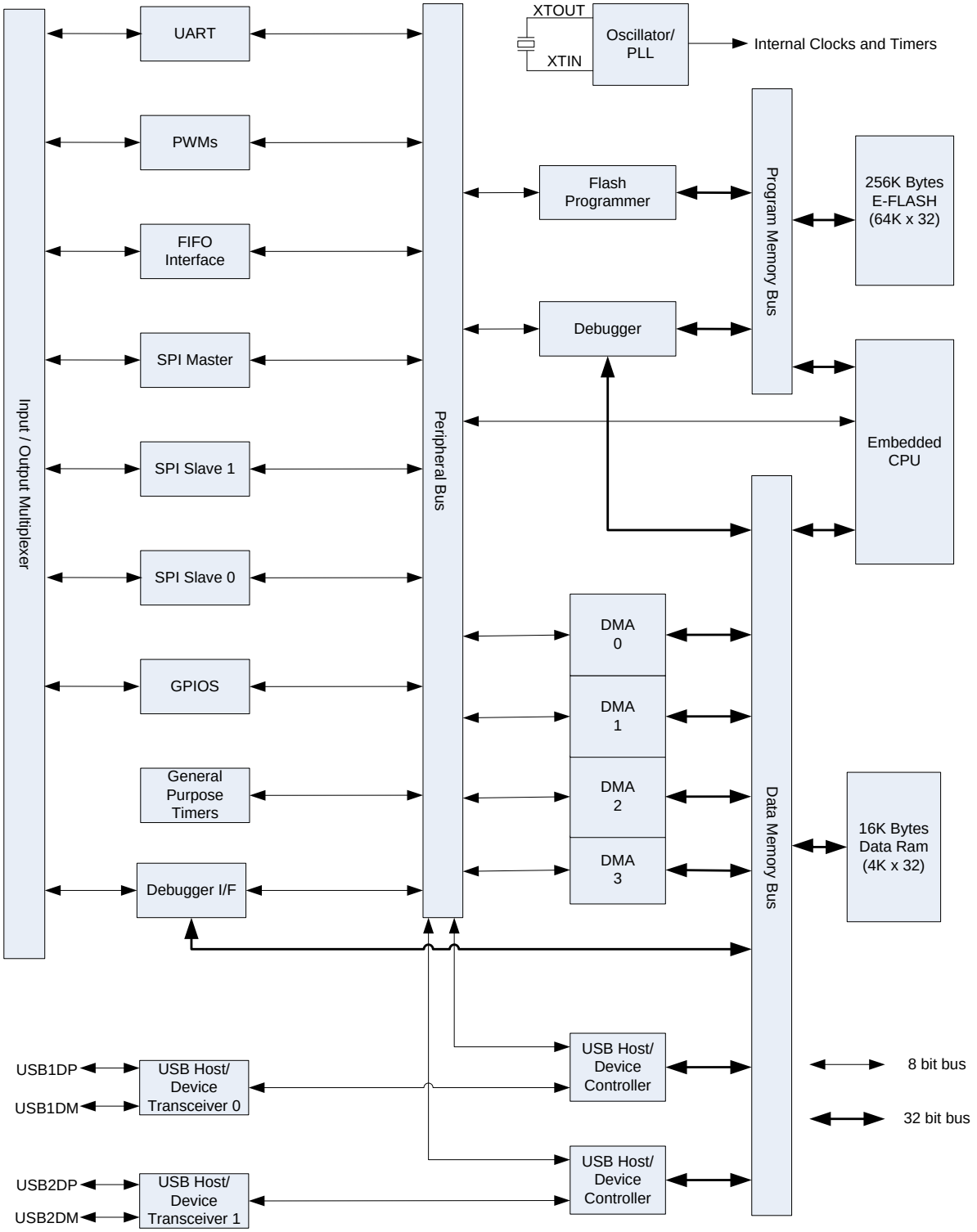

Figure 2.1 Simplified VNC2 Block Diagram

Table of Contents

3 Device Pin Out and Signal Description Summary

VNC2 is available in six packages: 32 pin LQFP, 32 pin QFN, 48 pin LQFP (pin compatible with VNC1L), 48 pin QFN, 64 pin LQFP and 64 pin QFN. **Figure 3.3** shows how the VNC2 pins map to the VNC1L pins (VNC2 pins labelled in bold text):

3.1 Pin Out - 32 pin LQFP

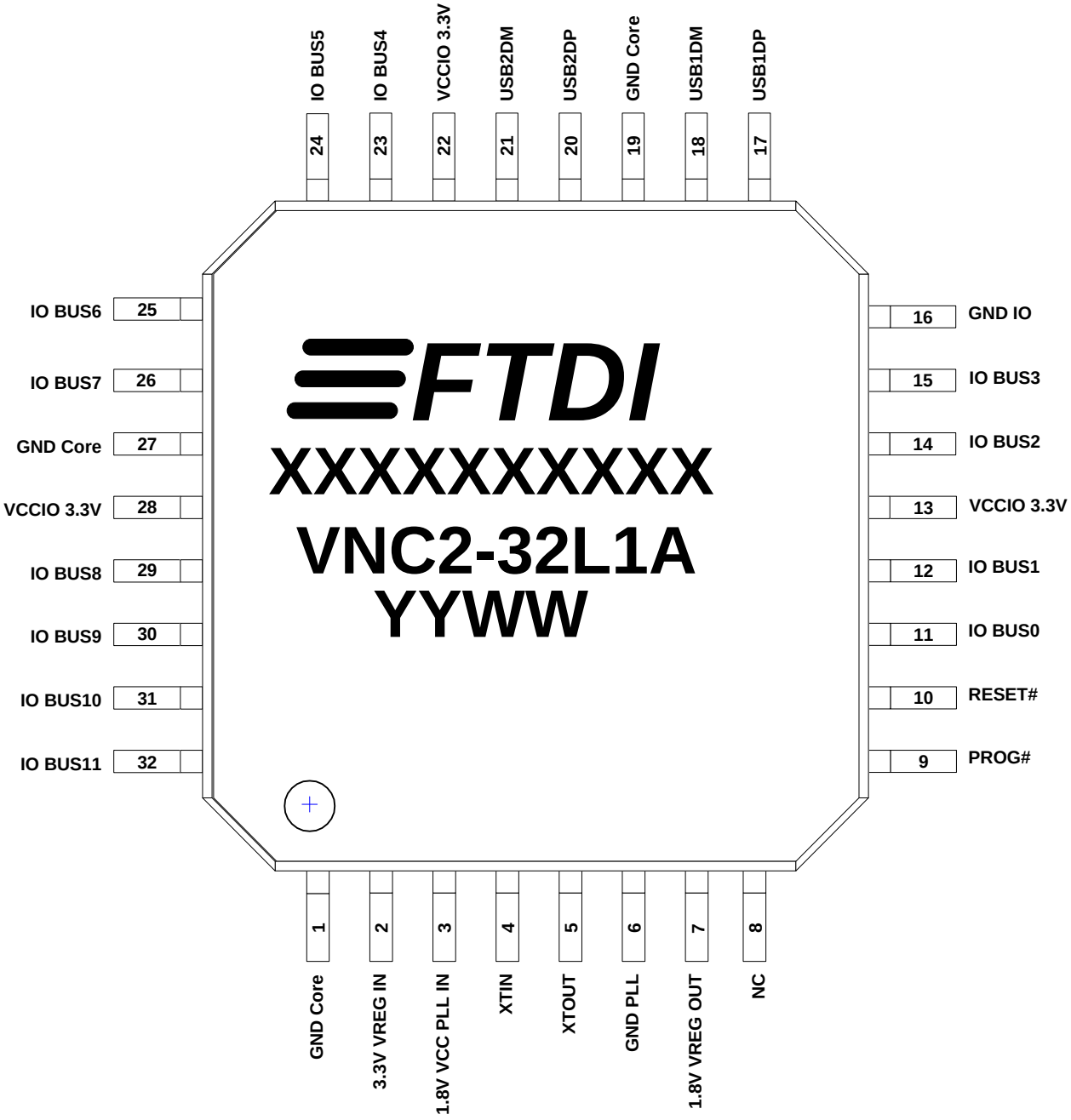

Figure 3.1 32 Pin LQFP – Top Down View

3.2 Pin Out - 32 pin QFN

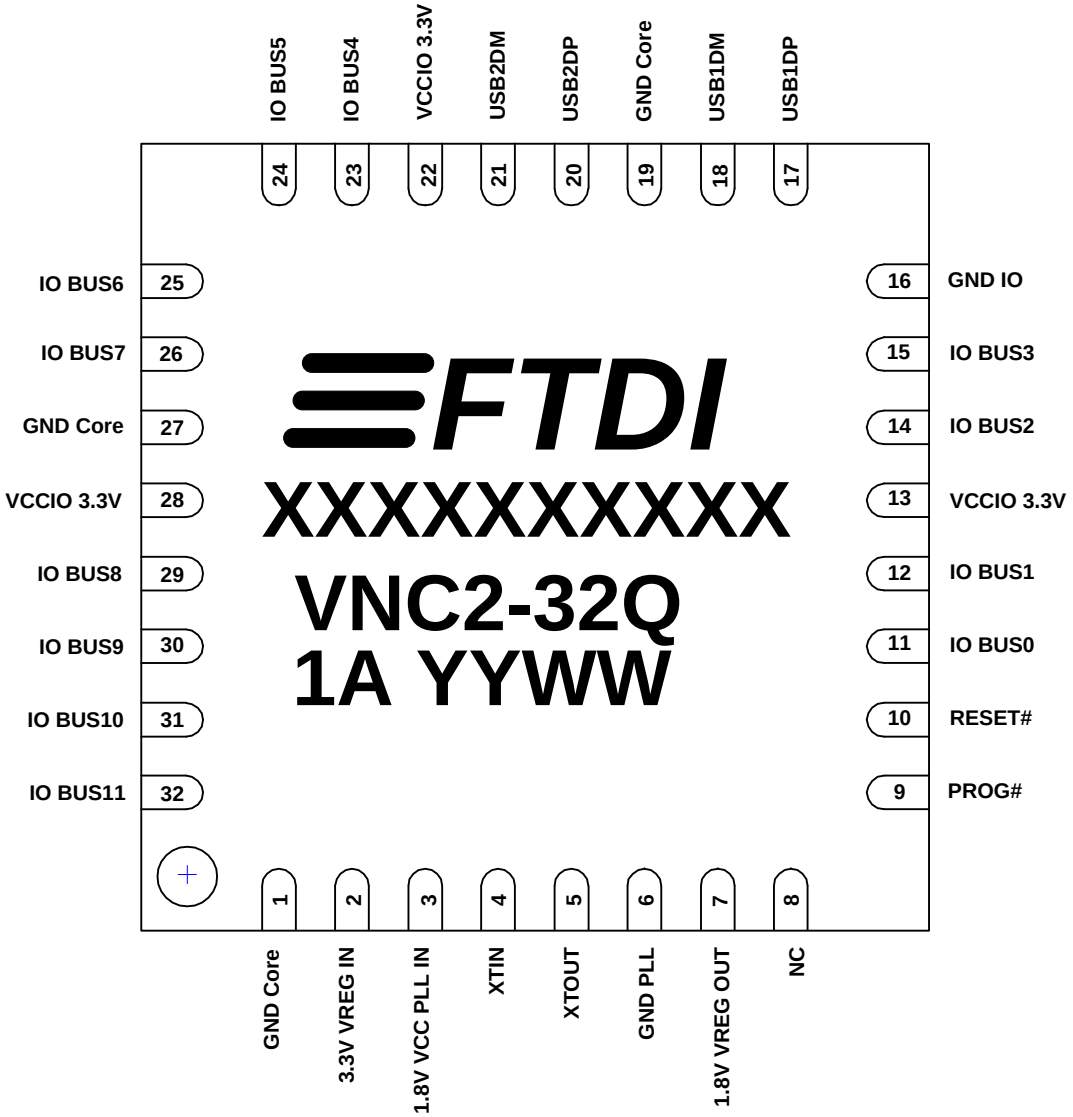

Figure 3.2 32 Pin QFN – Top Down View

3.3 Pin Out - 48 pin LQFP

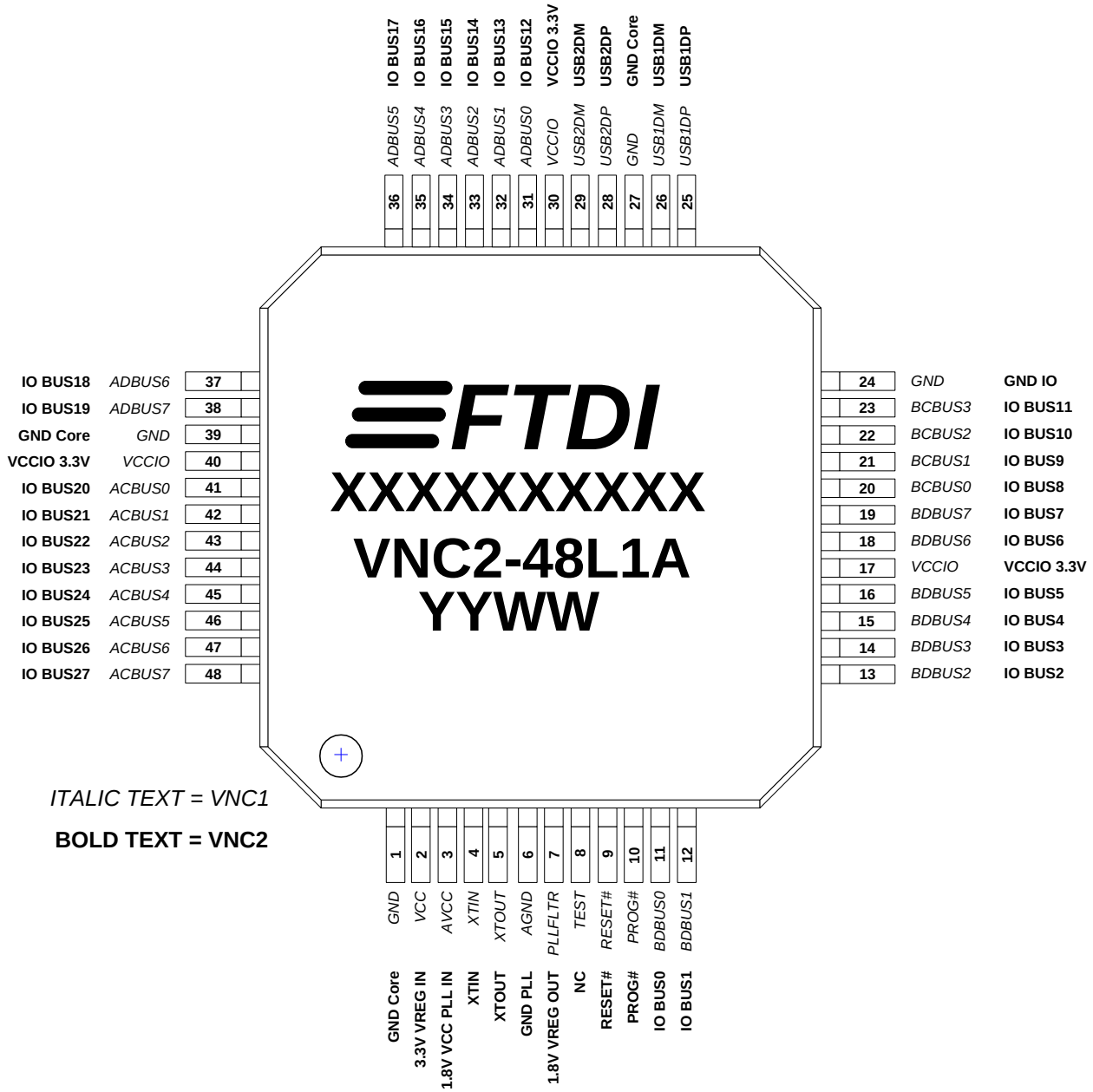

Figure 3.3 48 Pin LQFP – Top Down View

3.4 Pin Out - 48 pin QFN

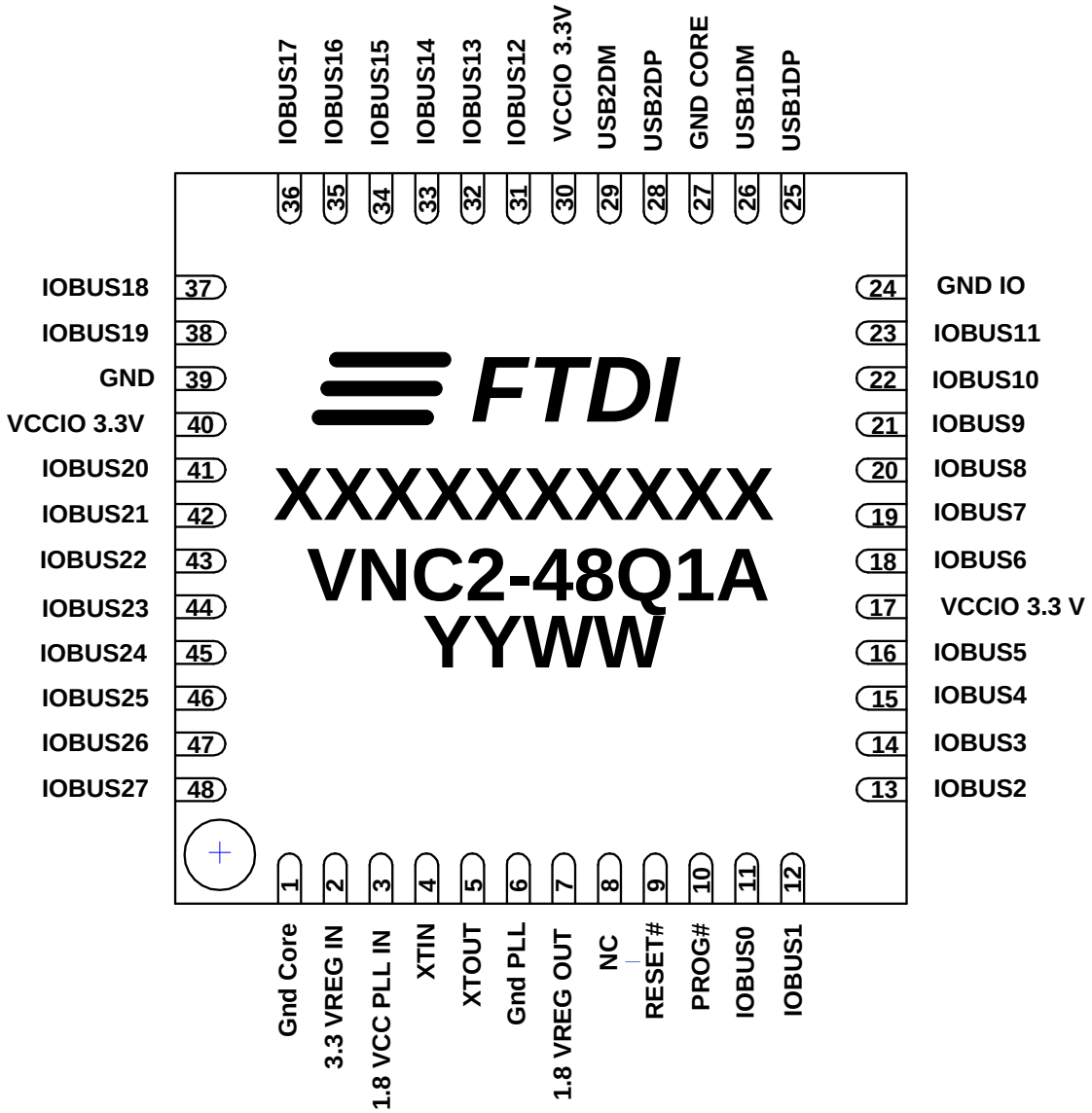

Figure 3.4 48 Pin QFN – Top Down View

3.5 Pin Out - 64 pin LQFP

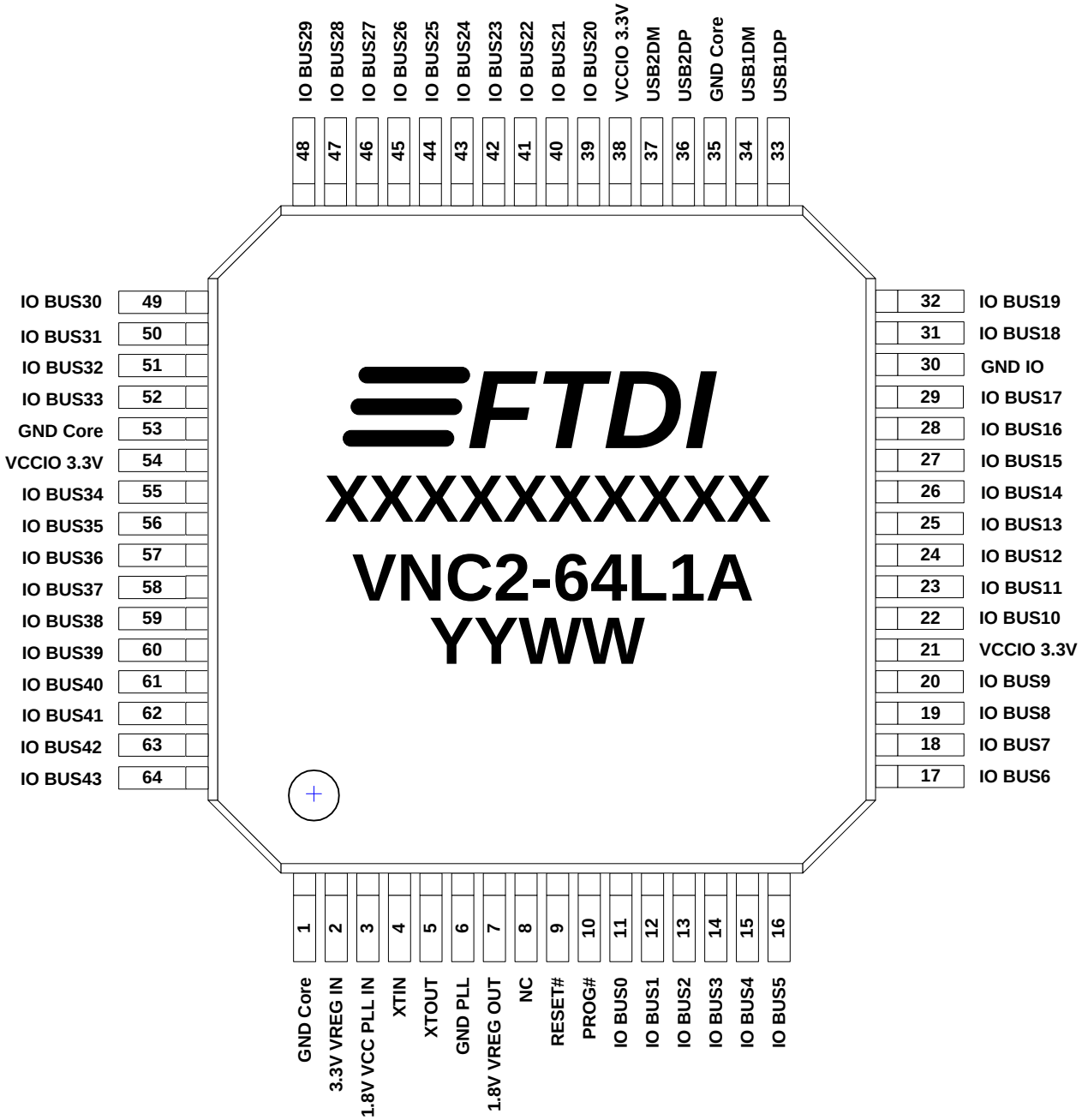

Figure 3.5 64 Pin LQFP – Top Down View

3.6 Pin Out - 64 pin QFN

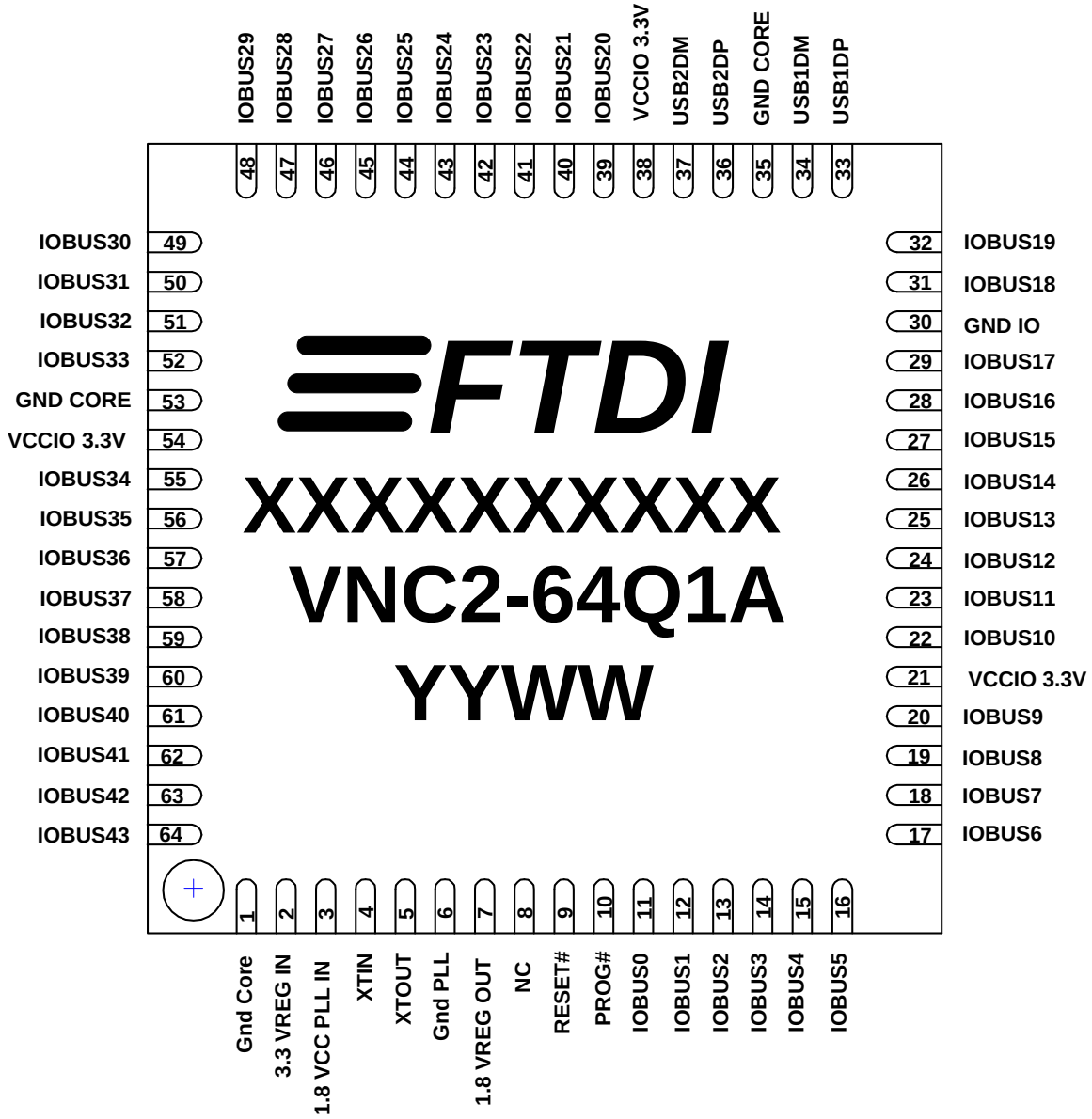

Figure 3.6 64 Pin QFN – Top Down View

3.7 VNC2 Schematic symbol 32 Pin

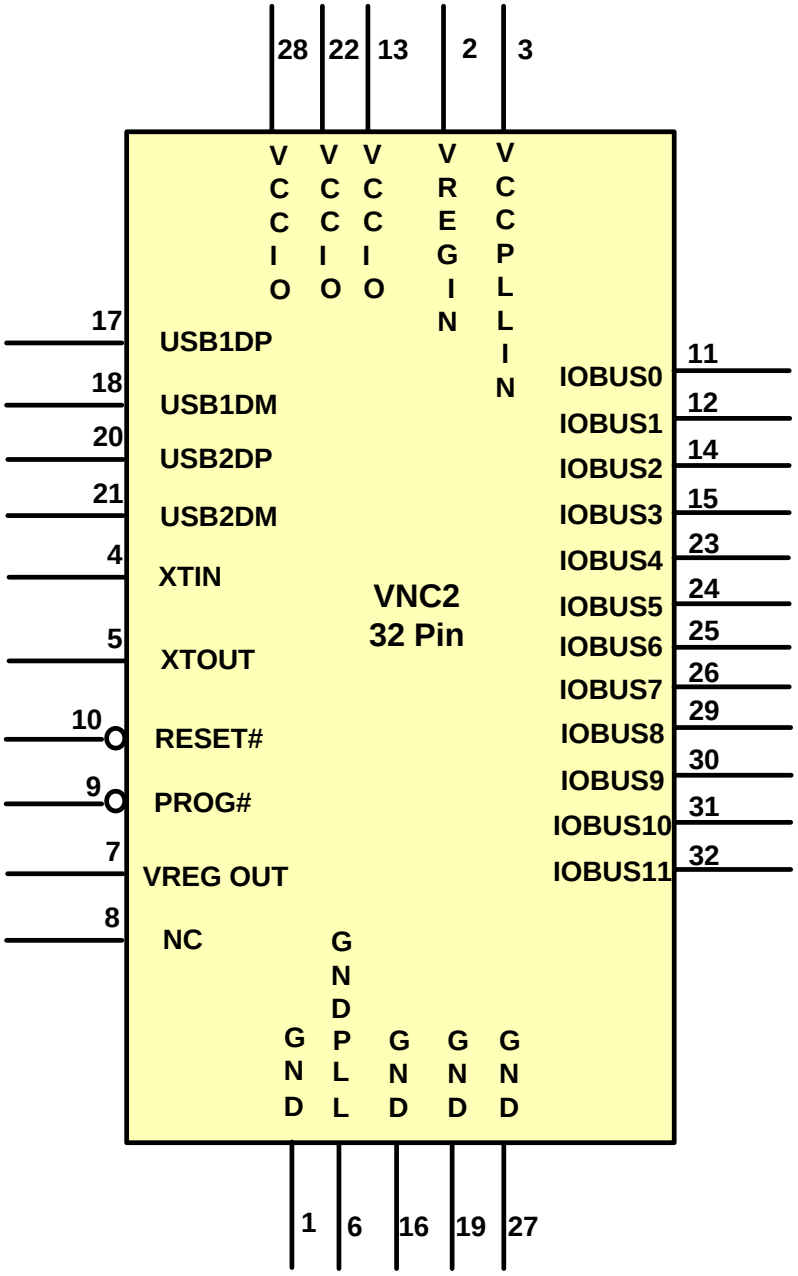

Figure 3.7 Schematic Symbol 32 Pin

3.8 VNC2 Schematic symbol 48 Pin

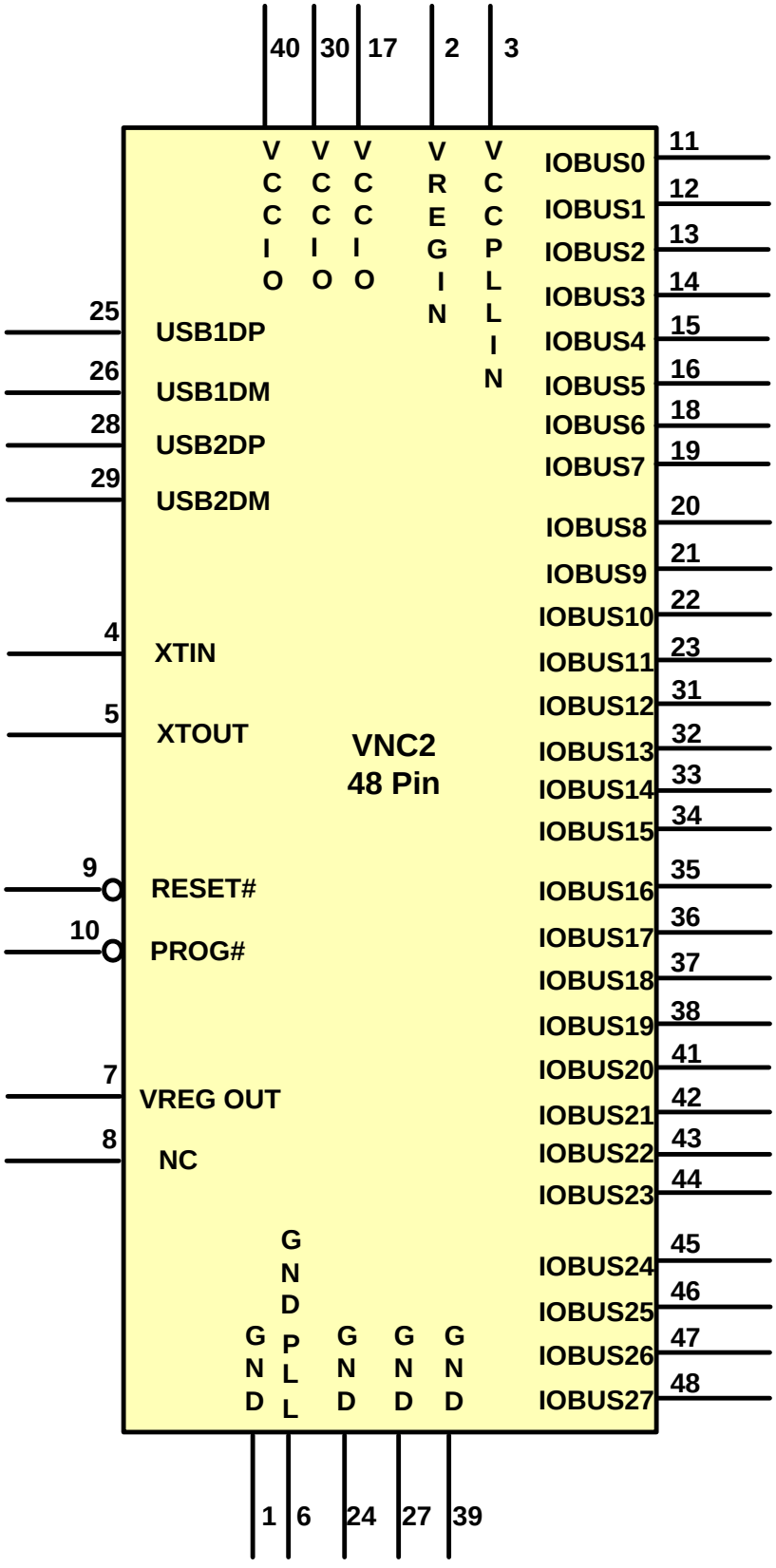

Figure 3.8 Schematic Symbol 48 Pin

3.9 VNC2 Schematic symbol 64 Pin

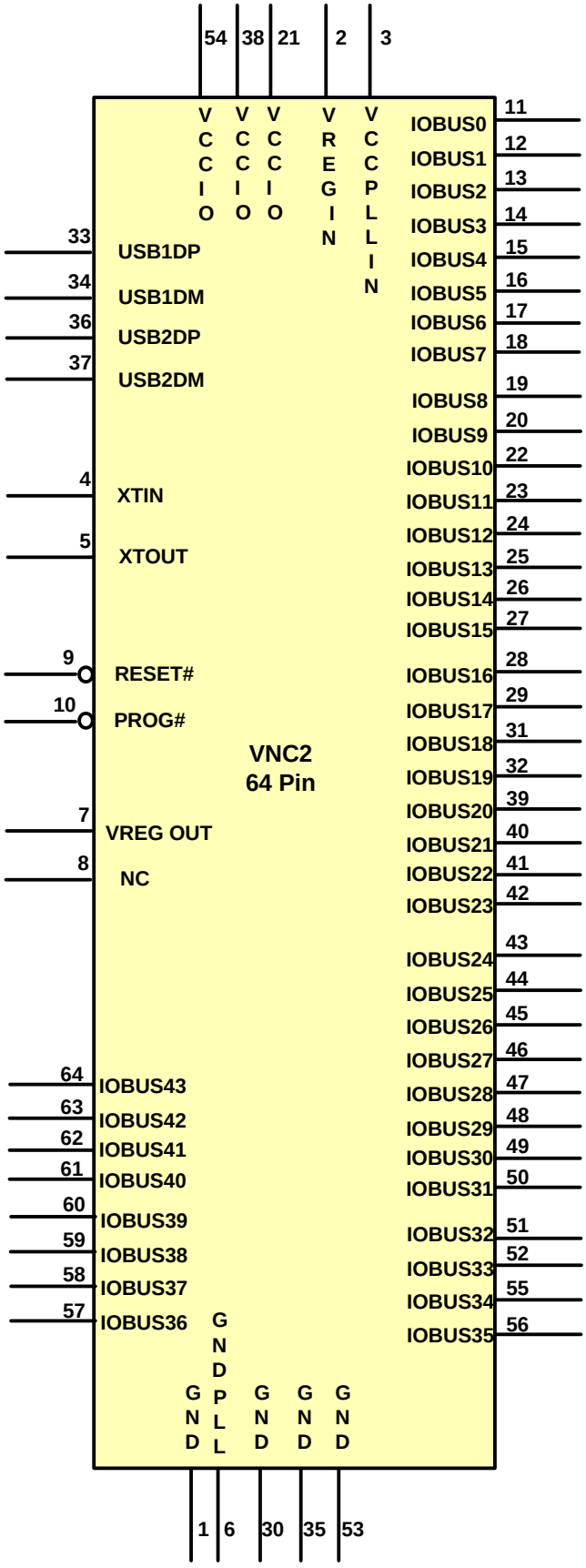

Figure 3.9 Schematic Symbol 64 Pin

3.10 Pin Configuration USB and Power

Pin No			Name	Type	Description
64 pin	48 pin	32 pin			
33	25	17	USB1DP	I/O	USB host/slave port 1 - USB Data Signal Plus with integrated pull-up/pull-down resistor.
34	26	18	USB1DM	I/O	USB host/slave port 1 - USB Data Signal Minus with integrated pull-up/pull-down resistor.
36	28	20	USB2DP	I/O	USB host/slave port 2 - USB Data Signal Plus with integrated pull-up/pull-down resistor.
37	29	21	USB2DM	I/O	USB host/slave port 2 - USB Data Signal Minus with integrated pull-up/pull-down resistor.

Table 3.1 USB Interface Group

Pin No			Name	Type	Description
64 pin	48 pin	32 pin			
1, 30, 35, 53	1, 24, 27, 39	1, 16, 19, 27	GND	PWR	Device ground supply pins.
2	2	2	3.3V VREGIN	PWR	+3.3V supply to the regulator.
3	3*	3	1.8V VCC PLL IN	PWR	+1.8V supply to the internal clock multiplier. This pin requires a 100nF decoupling capacitor. * 48 pin LQFP package only – This power input is internally connected to VREG_OUT. All other packages need this pin connected to a 1.8V power source. Most common applications will connect this to VREG_OUT.
6	6	6	GND PLL	PWR	Device analogue ground supply for internal clock multiplier.
7	7*	7	VREG OUT	Output	1.8V output from regulator to device core * N/C on 48 pin LQFP package only. All other packages will typically need to connect pins 7 and 3.
21, 38, 54	17, 30, 40	13, 22, 28	VCCIO	PWR	+3.3V supply to the input / output. Interface pins (IOBUS). Leaving the VCCIO unconnected will lead to unpredictable operation on the interface pins.

Table 3.2 Power and Ground

3.11 Miscellaneous Signals

Pin No			Name	Type	Description
64 pin	48 pin	32 pin			
4	4	4	XTIN	Input	Input to 12MHz Oscillator Cell. Connect 12MHz crystal across pins 4 and 5. If driven by an external source, reference to 1.8V VCC PLL IN.
5	5	5	XTOUT	Output	Output from 12MHz Oscillator Cell. Connect 12MHz crystal across pins 4 and 5. No connect if XTIN is driven by an external source.
8	8	8	NC	NC	No Connect (rev C) – (was TEST Rev A, B)
9	9	10	RESET#	Input	Can be used by an external device to reset VNC2.
10	10	9	PROG#	Input	Asserting PROG# on its own enables programming mode.

Table 3.3 Miscellaneous Signal Group

Note 1: # is used to indicate an active low signal.

Note 2: Pin 9 and 10 are 5V safe inputs

3.12 Pin Configuration Input / Output

VNC2 has multiple interfaces available for connecting to external devices. These are UART, FIFO, SPI slave, SPI master, GPIO and PWM. The Interface I/O Multiplexer is used to share the available I/O Pins between each peripheral.

VNC2 is configured with default settings for the I/O pins however they can be easily changed to suit the needs of a designer. This is explained in **Section 5 – I/O Multiplexer.** Default configuration for each package type is shown in **Table 3.4- Default I/O Configuration.** The signal names are also indicated for the VNC1L device as it is pin-compatible with the 48 pin LQFP VNC2 device.

Note: The default values of the pins listed in the following table are only available when the I/O Mux is enabled. A blank VNC2 chip defaults to all I/O pins as inputs.

Pin No			Name	64 Pin	48 Pin	32 PIN	Type	Description
64 Pin	48 Pin	32 Pin	(VINC1-L)	Default	Default	Default		
11	11	11	IOBUS0 (BDBUS0)	debug_if	debug_if	debug_if	I/O	GPIO
12	12	12	IOBUS1 (BDBUS1)	Input	pwm[1]	gpio[A1]	I/O	GPIO
13	13	14	IOBUS2 (BDBUS2)	Input	pwm[2]	gpio[A2]	I/O	GPIO
14	14	15	IOBUS3 (BDBUS3)	Input	pwm[3]	gpio[A3]	I/O	GPIO
15	15	23	IOBUS4 (BDBUS4)	fifo_data[0]	spi_s0_clk	uart_txd	I/O	GPIO
16	16	24	IOBUS5 (BDBUS5)	fifo_data[1]	spi_s0_mosi	uart_rxd	I/O	GPIO
17	18	25	IOBUS6 (BDBUS6)	fifo_data[2]	spi_s0_miso	uart_rts#	I/O	GPIO
18	19	26	IOBUS7 (BDBUS7)	fifo_data[3]	spi_s0_ss#	uart_cts#	I/O	GPIO
19	20	29	IOBUS8 (BCBUS0)	fifo_data[4]	spi_m_clk	spi_s0_clk	I/O	GPIO
20	21	30	IOBUS9 (BCBUS1)	fifo_data[5]	spi_m_mosi	spi_s0_mosi	I/O	GPIO
22	22	31	IOBUS10 (BCBUS2)	fifo_data[6]	spi_m_miso	spi_s0_miso	I/O	GPIO
23	23	32	IOBUS11 (BCBUS3)	fifo_data[7]	spi_m_ss_0#	spi_s0_ss#	I/O	GPIO
24	31	-	IOBUS12 (ADBUS0)	fifo_rxf#	uart_txd		I/O	GPIO
25	32	-	IOBUS13 (ADBUS1)	fifo_txe#	uart_rxd		I/O	GPIO
26	33	-	IOBUS14 (ADBUS2)	fifo_rd#	uart_rts#		I/O	GPIO

Pin No			Name	64 Pin	48 Pin	32 PIN	Type	Description
64 Pin	48 Pin	32 Pin	(VINC1-L)	Default	Default	Default		
27	34	-	IOBUS15 (ADBUS3)	fifo_wr#	uart_cts#		I/O	GPIO
28	35	-	IOBUS16 (ADBUS4)	fifo_oe#	uart_dtr#		I/O	GPIO
29	36	-	IOBUS17 (ADBUS5)	Input	uart_dsr#		I/O	GPIO
31	37	-	IOBUS18 (ADBUS6)	Input	uart_dcd#		I/O	GPIO
32	38	-	IOBUS19 (ADBUS7)	Input	uart_ri#		I/O	GPIO
39	41	-	IOBUS20 (ACBUS0)	uart_txd	uart_tx_active		I/O	GPIO
40	42	-	IOBUS21 (ACBUS1)	uart_rxd	gpio[A5]		I/O	GPIO
41	43	-	IOBUS22 (ACBUS2)	uart_rts#	gpio[A6]		I/O	GPIO
42	44	-	IOBUS23 (ACBUS3)	uart_cts#	gpio[A7]		I/O	GPIO
43	45	-	IOBUS24 (ACBUS4)	uart_dtr#	gpio[A0]		I/O	GPIO
44	46	-	IOBUS25 (ACBUS5)	uart_dsr#	gpio[A1]		I/O	GPIO
45	47	-	IOBUS26 (ACBUS6)	uart_dcd#	gpio[A2]		I/O	GPIO
46	48	-	IOBUS27 (ACBUS7)	uart_ri#	gpio[A3]		I/O	GPIO
47	-	-	IOBUS28	uart_tx_active			I/O	GPIO
48	-	-	IOBUS29	Input			I/O	GPIO
49	-	-	IOBUS30	Input			I/O	GPIO
50	-	-	IOBUS31	Input			I/O	GPIO
51	-	-	IOBUS32	spi_s0_clk			I/O	GPIO
52	-	-	IOBUS33	spi_s0_mosi			I/O	GPIO
55	-	-	IOBUS34	spi_s0_miso			I/O	GPIO
56	-	-	IOBUS35	spi_s0_ss#			I/O	GPIO
57	-	-	IOBUS36	spi_s1_clk			I/O	GPIO

Pin No			Name (VINC1-L)	64 Pin Default	48 Pin Default	32 PIN Default	Type	Description
64 Pin	48 Pin	32 Pin						
58	-	-	IOBUS37	spi_s1_mosi			I/O	GPIO
59	-	-	IOBUS38	spi_s1_miso			I/O	GPIO
60	-	-	IOBUS39	spi_s1_ss#			I/O	GPIO
61	-	-	IOBUS40	spi_m_clk			I/O	GPIO
62	-	-	IOBUS41	spi_m_mosi			I/O	GPIO
63	-	-	IOBUS42	spi_m_miso			I/O	GPIO
64	-	-	IOBUS43	spi_m_ss_0#			I/O	GPIO

Table 3.4 Default I/O Configuration

Note: All GPIO are 5V safe inputs

4 Function Description

VNC2 is the second of FTDIs Vinculum family of Embedded USB host controller integrated circuit devices. VNC2 can encapsulate certain USB device classes by handling the USB Host Interface and data transfer functions using the in-built EMCU and embedded Flash memory. When interfacing to mass storage devices, such as USB Flash drives, VNC2 transparently handles the FAT file structure using a simple to implement command set. VNC2 provides a cost effective solution for introducing USB host capability into products that previously did not have the hardware resources to do so.

VNC2 has an associated software development tool suite to allow users to create customised firmware.

4.1 Key Features

VNC2 is a programmable SoC device with a powerful embedded microprocessor core and dual USB interfaces, large RAM and Flash capacity and the ability to develop and customise firmware using the VNC2 Toolchain. VNC2 has an enhanced feature list over and above VNC1L; however the 48 pin LQFP package is backward compatible with the VNC1L.

4.2 Functional Block Descriptions

The following paragraphs describe each function within VNC2. Please refer to the block diagram shown in **Figure 2.1**

4.2.1 Embedded CPU

The processor core is based on FTDIs proprietary 16-bit embedded MCU architecture. The EMCU has a Harvard architecture with separate code and data space.

4.2.2 Flash Module

VNC2 has 256k bytes (128k x 16-bits) of embedded Flash (E-FLASH) memory. No special programming voltages are necessary for programming the on-board E-FLASH as these are provided internally on-chip.

4.2.3 Flash Programming Module

The purpose of the flash programmer module is to perform all necessary operations for programming the flash, from general usage to first power on sequencing. This block is responsible for handling device firmware upgrades which can be accessed by the debugger interface, a USB cable or Flash drive interface.

4.2.4 Input / Output Multiplexer Module

VNC2 peripheral interfaces are UART, SPI slave0, SPI slave1, SPI master, FIFO-Asynchronous, FIFO-Synchronous, GPIO, debug interface and PWM.

The I/O multiplexer allows the designer to select which peripherals are connected to the device I/O pins.

The selectable peripheral interfaces are only limited by the number of I/O pins available. All peripherals are available across the package range except synchronous FIFO mode which cannot be selected on 32 pin packages. The available configurable I/O pins per package are as follows:

- 32 pin package – 12 I/O pins
- 48 pin package – 28 I/O pins
- 64 pin package – 44 I/O pins

Table 4.1 lists the peripherals which can be multiplexed to I/O and the maximum number of pins required for each one. The designer can choose any mix of peripheral configurations as long as they are within the specific package I/O pin count. Depending on the design not all 9 UART pins need to be configured. Similarly the GIPO peripheral does not need all pins configured.

e.g. The 48 pin package has 28 I/O pins which could be configured as UART – 9 pins, SPI Master – 5 pins, FIFO Asynchronous – 12 pins and GPIO – 2 pins. This makes a total of 28 pins.

Please refer to **Section 5** for a detailed description of the I/O multiplexer.

Peripherals	Maximum pins required
UART	9
SPI Slave 0	4
SPI Slave 1	4
SPI Master	5
FIFO Asynchronous	12
FIFO Synchronous	14
GPIO	40
Debug	1
PWM	8

Table 4.1 - Peripheral Pin Requirements

4.2.5 Peripheral DMA Modules 0, 1, 2 & 3

The peripheral DMA has the capability to transfer data to and from an I/O device. The CPU can offload the transfer of data between the processor and the peripheral freeing the CPU to execute other instructions.

The DMA module collects or transmits data from memory to an I/O address space; it is also capable of copying data in memory and transferring it to another location.

The DMA is not accessible by the user as it automatically controlled by the CPU.

4.2.6 RAM Module

The RAM module consists of 16k bytes on-chip (4k x 32-bits) data memory. The RAM is byte addressable.

4.2.7 Peripheral Interface Modules

VNC2 has nine peripheral interface modules. Full descriptions of each module are described in **section 6.**

- Debugger Interface
- UART
- PWM
- FIFO
- SPI Master
- SPI Slave 0 & 1
- GPIO - General purpose I/O pins
- General purpose timers

4.2.8 USB Transceivers 0 and 1

Two USB transceiver cells provide the physical USB device interface supporting USB 1.1 and USB 2.0 standards. Low-speed and full-speed USB data rates are supported. Each output driver provides +3.3V level slew rate control signalling, whilst a differential receiver and two single ended receivers provide USB DATA IN, SE0 and USB Reset condition detection. These cells also include integrated internal USB pull-up or pull-down resistors as required for host or slave mode.

4.2.9 USB Host / Device Controllers

These blocks handle the parallel-to-serial and serial-to-parallel conversion of the USB physical layer. This includes bit stuffing, CRC generation, USB frame generation and protocol error checking. The Host / Device controller is autonomous and therefore requires limited load from the CPU.

4.2.10 12MHz Oscillator

The 12MHz Oscillator cell generates a 12MHz reference clock input to the Clock Multiplier PLL from an external 12MHz crystal. The external crystal is connected across Pin 4 – XTIN and Pin 5 – XTOUT in the configuration shown in **Figure 10.1.**

4.2.11 Power Saving Modes and Standby mode.

VNC2 can be set to operate in three frequencies allowing the user to select a slower speed to reduce power consumption. Three operating frequencies available are 12MHz, 24MHz and normal operation of 48MHz. These operating modes can be configured using the RTOS. Full details are available in the RTOS manual available from the FTDI website.

When a particular peripheral is not used, it is powered down internally thus saving power.

Standby mode is available under firmware control, this mode puts the VNC2 in a state with no clocks running or system blocks powered. The device will wake up out of this mode by toggling any of the following signals: USB0/1 DP or DM, SPI slave 0 select (spi_s0_ss#), SPI slave 1select (spi_s1_ss#) or UART ring indicator (uart_ri#).

5 I/O Multiplexer

FTDI devices typically have multiple interfaces available to communicate with external devices. VNC2 has UART, SPI slave0, SPI slave1, SPI master, FIFO, GPIO, and PWM peripherals. The available packages for VNC2 provide any of these interfaces to be active on the available pins through the use of an I/O Multiplexer. **Table 5.1** lists the signals available for each peripheral. **Table 5.2 to 12** explain the use of the I/O multiplexer.

Multiplexers are used to connect the VNC2 peripherals to the external IOBUS pins. This enables the designer to select which IOBUS pins he wishes to map a particular peripheral to. Peripheral signals are allocated to one of four groups, which connect to the I/O multiplexer. Each I/O peripheral signal can connect to one out of every four external IOBUS pins. The IOBUS pin that a peripheral signal can connect to is dictated by the peripheral signal's group. For example, if a peripheral signal is allocated to group 0 then it can connect to IOBUS0, IOBUS4, IOBUS8, and IOBUS12 and so on. If a peripheral signal is allocated to group 1 then it can connect to IOBUS1, IOBUS5, IOBUS9, and IOBUS13 and so on. Figure 5.1 details the I/O multiplexer concept, where, for example, a white peripheral signal can connect to any white IOBUS pin; a green peripheral signal can connect to a green IOBUS pin. Figure 5.2, Figure 5.3 and Figure 5.4 give examples of connecting peripheral signals to differing IOBUS pins.

The IO Multiplexer also provides the following features:

- Ability to configure an I/O pad as an input, output or bidirectional pad.
- At power on reset, all pins are set as inputs by default. Whenever the I/O Mux is enabled the pins are configured as their default values listed Table 6 within section **3.12**.

Note: It is recommended not to reassign the debug interface signal (debug_if) from its default setting of IOBUS0 (Pin 11 on all packages). This assumes that the debug pin is required in the application design, if not; pin 11 can be assigned to any other group 0 signal.

An application (IOMUX) within the RTOS is available to aid with pin configuration, **Section 5.2** has more details.

Further details of the IO Multiplexer are available within Application Note AN_139 Vinculum-II IO Mux Explained.

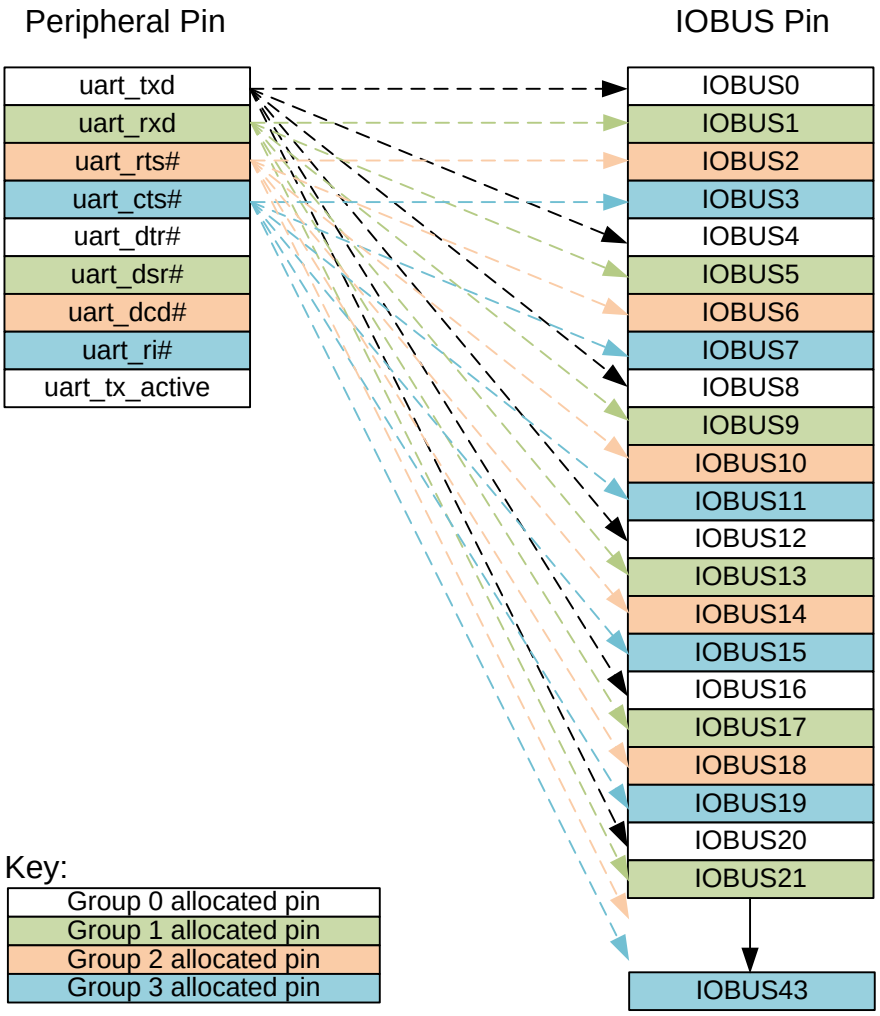

Figure 5.1 IOBUS to Group Relationship-64 Pin

Figure 5.2 details the UART, SPI slave0 and SPI master connecting to IOBUS pins:

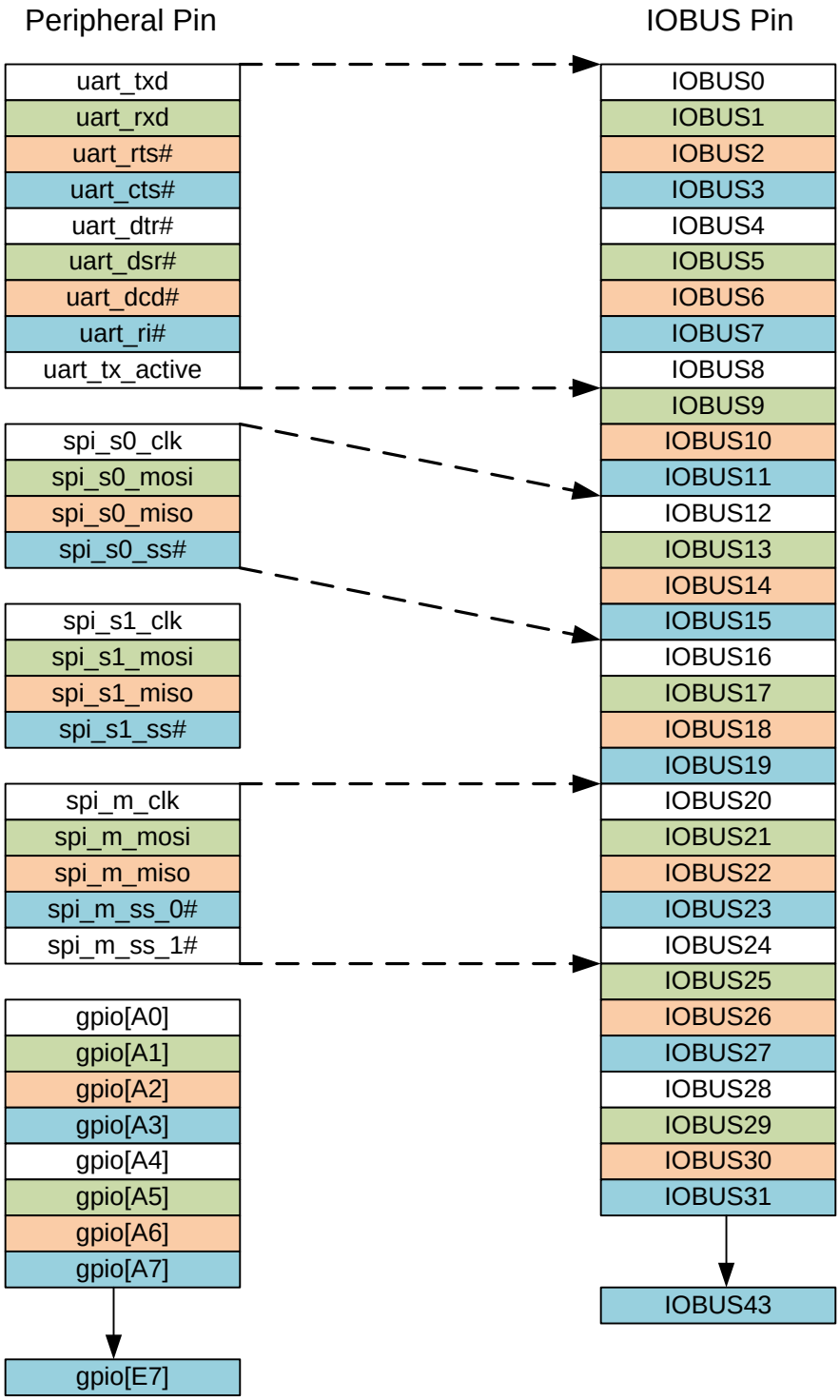

Figure 5.2 IOBUS to UART, SPI slave0 and SPI master example

Figure 5.3 expands upon Figure 5.2 by moving the UART, SPI slave0 and SPI master signals to different IOBUS positions. The purpose of this diagram to highlight peripherals connected to differing IOBUS positions.

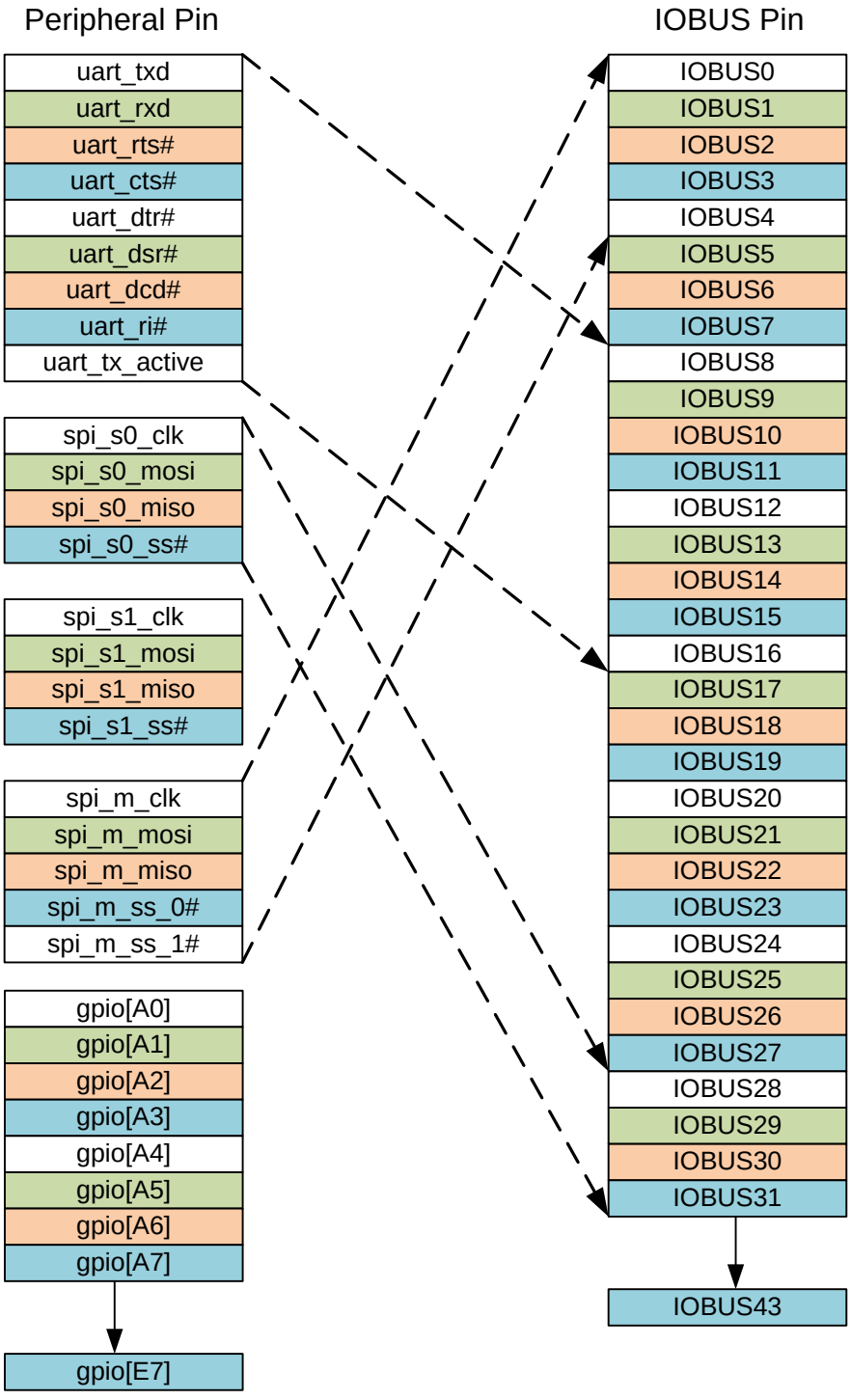

Figure 5.3 IOBUS to UART, SPI slave0 and SPI master second example

With reference to Figure 5.3, it can be seen that IOBUS9-11 and IOBUS16-19 were unused. Figure 5.4 expands upon the previous two figures to detail a fully occupied IOBUS, up to and including IOBUS19. The gaps at IOBUS9-11 have been filed with 3 GPIO pins, the gaps at IOBUS16-19 have been filled with the second SPI slave and a further 3 IOBUS pins (17-19) have been allocated to 3 GPIO pins. Note that GPIO pins A0 and A4 are unused as a sufficient gap wasn't available.

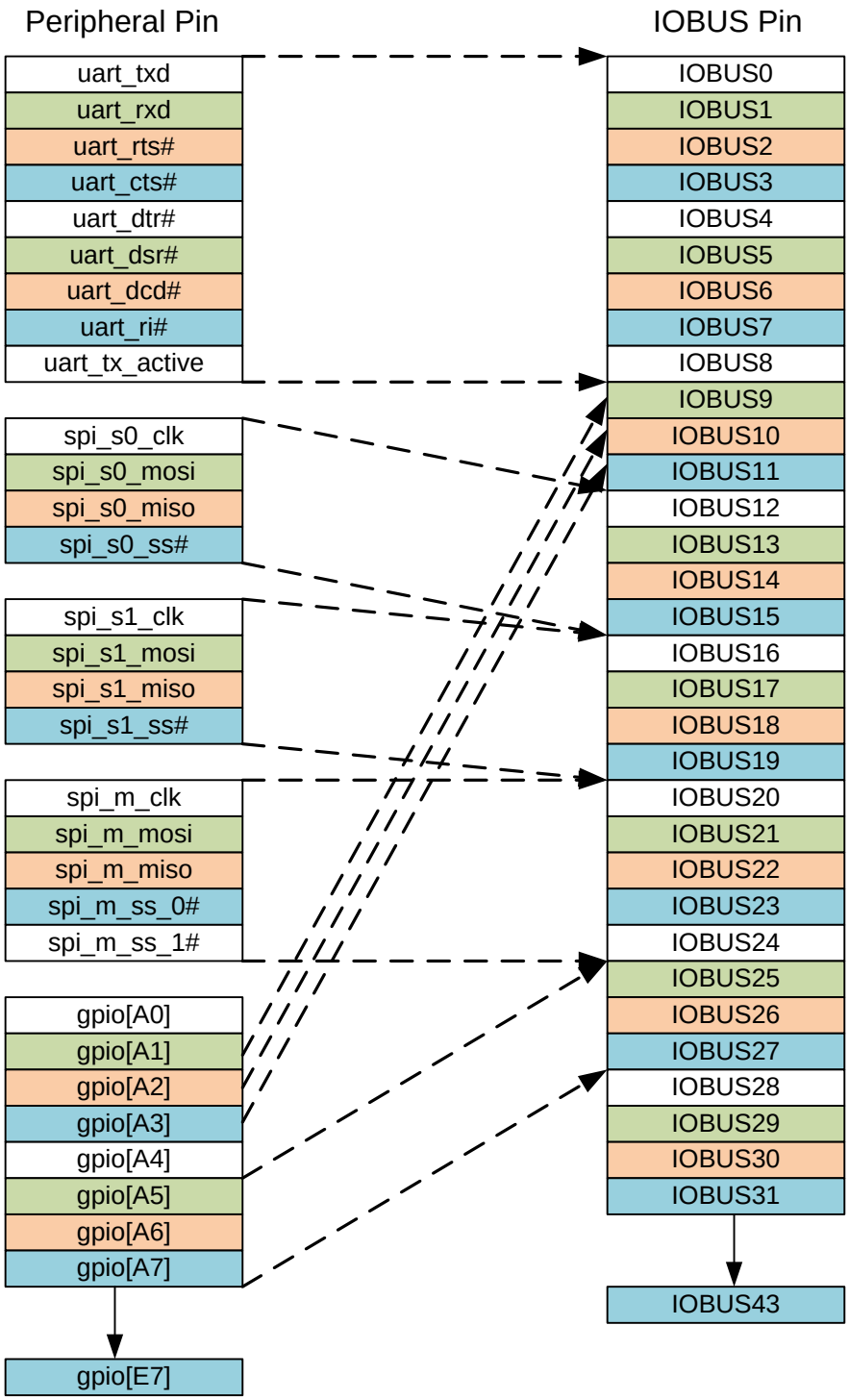

Figure 5.4 IOBUS to UART, SPI slave0 and SPI master third example

5.1 I/O Peripherals Signal Names

Peripheral	Signal Name	Outputs	Inputs	Description
Debugger	debug_if	1	1	debugger interface
UART	uart_txd	1	0	Transmit asynchronous data output
	uart_rts#	1	0	Request to send control output
	uart_dtr#	1	0	Data acknowledge (data terminal ready control) output
	uart_tx_active	1	0	Enable transmit data for RS485 designs
	uart_rxd	0	1	Receive asynchronous data input
	uart_cts#	0	1	Clear to send control input
	uart_dsr#	0	1	Data request (data set ready control) input
	uart_ri#	0	1	Ring indicator control input
	uart_dcd#	0	1	Data carrier detect control input
FIFO	fifo_data	8	8	FIFO data bus
	fifo_txe#	1	0	When high, do not write data into the FIFO. When low, data can be written into the FIFO by strobing WR high, then low.
	fifo_rxf#	1	0	When high, do not read data from the FIFO. When low, there is data available in the FIFO which can be read by strobing RD# low, then high.
	fifo_wr#	0	1	Writes the data byte on the D0...D7 pins into the transmit FIFO buffer when WR goes from high to low.
	fifo_rd#	0	1	Enables the current FIFO data byte on D0...D7 when low. Fetches the next FIFO data byte (if available) from the receive FIFO buffer when RD# goes from high to low
	fifo_oe#	0	1	FIFO output enable – synchronous FIFO only
	fifo_clkout	0	1	FIFO clock out – synchronous FIFO only
GPIO	gpio	40	40	General purpose I/O
SPI Slave 0	spi_s0_clk	0	1	SPI clock input – slave 0
	spi_s0_ss#	0	1	SPI chip select input – slave 0
	spi_s0_mosi	1	1	SPI master out serial in – slave 0
	spi_s0_miso	1	0	SPI master in slave out – slave 0
SPI Slave 1	spi_s1_clk	0	1	SPI clock input – slave 1
	spi_s1_ss#	0	1	SPI chip select input – slave 1
	spi_s1_mosi	1	1	Master out slave in – slave 1
	spi_s1_miso	1	0	Master in slave out – slave 1
SPI Master	spi_m_clk	1	0	SPI clock input – master
	spi_m_mosi	1	1	Master out slave in - master
	spi_m_miso	0	1	Master in slave out - master
	spi_m_ss_0#	1	0	Active low slave select 0 from master to slave 0
	spi_m_ss_1#	1	0	Active low slave select 1 from master to slave 1
PWM	pwm	8	0	Pulse width modulation

Table 5.1 I/O Peripherals Signal Names

Note: # is used to indicate an active low signal.

5.2 I/O Multiplexer Configuration

The VNC2 I/O Multiplexer allows signals to be routed to different pins on the device. To simplify the routing of signals, the VNC2 RTOS provides an utility (IOMux) to configure the I/O Multiplexer as the designer requires. The IOMux is fully integrated into the VNC2 IDE (Integrated development Environment) which is available to download: Vinculum-II Toolchain. A screenshot of the IOMux utility is shown in figure 5.5 below.

The IOMux utility user guide is available to download: VINCULUM-II IO Mux Configuration Utility User Guide

The following tables provide a lookup guide to determine what signals are available and the list of pins that can be used:

- **Table 5.2 Group 0**
- **Table 5.3 Group 1**
- **Table 5.4 Group 2**
- **Table 5.5 Group 3**

Each VNC2 has a default state of IOBUS signals following a hard reset. The number of I/O pins available is determined by the package size:

- Package 32pin (LQFP & QFN)- Twelve I/O pins – IOBUS0 to IOBUS11
- Package 48pin (LQFP & QFN)- Twenty eight I/O pins – IOBUS0 to IOBUS27
- Package 64pin (LQFP & QFN)- Forty-four I/O pins – IOBUS0 to IOBUS43

Section **3.12** shows the default signal settings for all three package sizes.

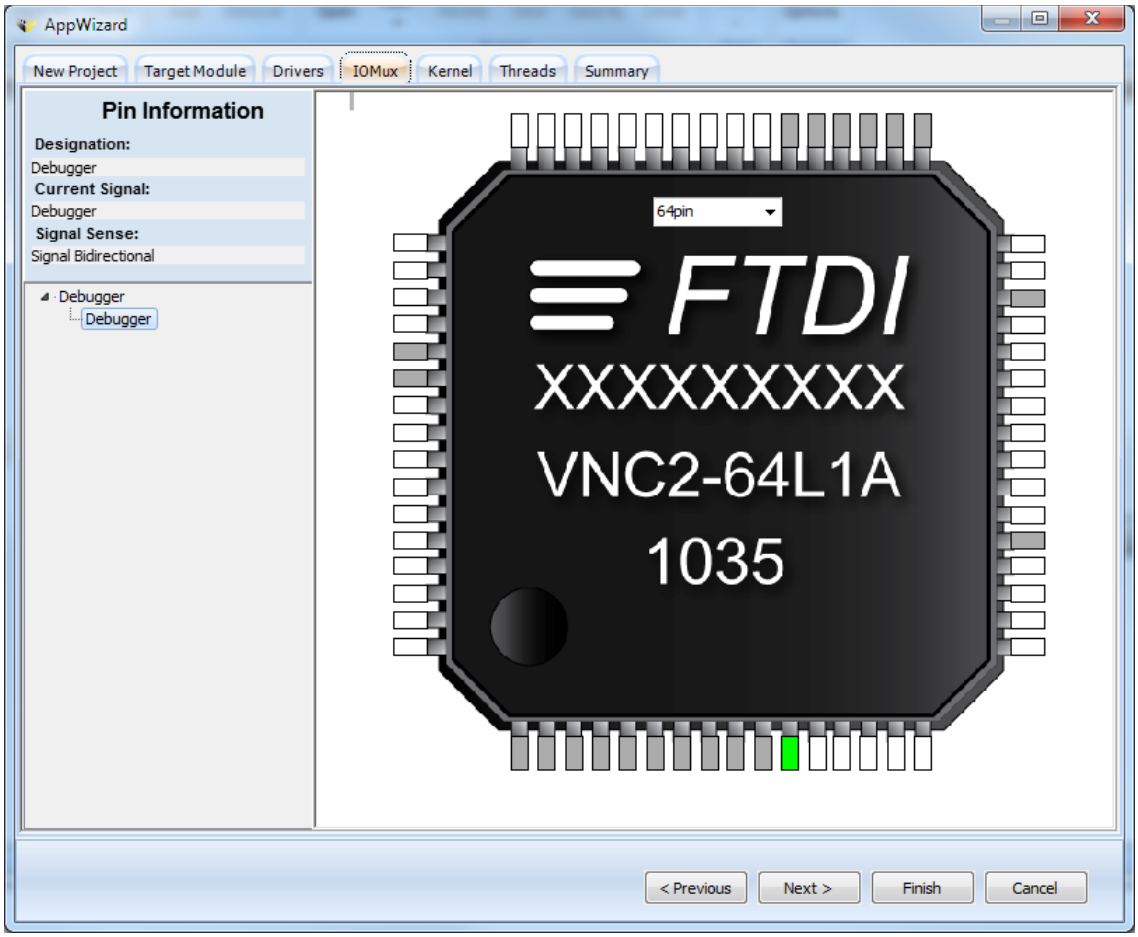

Figure 5.5 VNC2 Toolchain App Wizard showing IOMux Configuration

5.3 I/O Mux Group 0

Available Input signals	Available output signals	64 Pin Package Available pins	48 Pin Package Available pins	32 Pin Package Available pins
debug_if fifo_data[0] fifo_data[4] fifo_oe# spi_s0_clk spi_s1_clk gpio[A0] gpio[A4] gpio[B0] gpio[B4] gpio[C0] gpio[C4] gpio[D0] gpio[D4] gpio[E0] gpio[E4]	debug_if uart_txd uart_dtr# uart_tx_active fifo_data[0] fifo_data[4] fifo_rxf# pwm[0] pwm[4] spi_m_clk spi_m_ss_1# gpio[A0] gpio[A4] gpio[B0] gpio[B4] gpio[C0] gpio[C4] gpio[D0] gpio[D4] gpio[E0] gpio[E4]	11, 15, 19, 24, 28, 39, 43, 47, 51, 57, 61	11, 15, 20, 31, 35, 41, 45	11, 23 29

Table 5.2 Group 0

Table 5.2 - Input and output signals that are available for all the IOBUS pins that are in group 0. For example if using the 48 pin package device this would allow pins 11, 15, 20, 31, 35, 41 and 45 to be configured as either an input signal (listed in the first column) or a output signal (listed in the second column).

5.4 I/O Mux Group 1

Available Input signals	Available output signals	64 Pin Package Available pins	48 Pin Package Available pins	32 Pin Package Available pins
uart_rxd uart_dsr# fifo_data[1] fifo_data[5] spi_s0_mosi spi_s1_mosi gpio[A1] gpio[A5] gpio[B1] gpio[B5] gpio[C1] gpio[C5] gpio[D1] gpio[D5] gpio[E1] gpio[E5]	fifo_data[1] fifo_data[5] fifo_txe# pwm[1] pwm[5] spi_s0_mosi spi_s1_mosi spi_m_mosi fifo_clkout gpio[A1] gpio[A5] gpio[B1] gpio[B5] gpio[C1] gpio[C5] gpio[D1] gpio[D5] gpio[E1] gpio[E5]	12, 16, 20, 25, 29, 40, 44, 48, 52, 58, 62	12,16, 21, 32, 36, 42, 46	12, 24, 30

Table 5.3 Group 1

Table 5.3 - Input and output signals that are available for all the IOBUS pins that are in group 1. For example if using the 64 pin package device this would allow pins 12, 16, 20, 25, 29, 40, 44, 48, 52, 58 and 62 to be configured as either an input signal (listed in the first column) or a output signal (listed in the second column).

5.5 I/O Mux Group 2

Available Input signals	Available output signals	64 Pin Package Available pins	48 Pin Package Available pins	32 Pin Package Available pins
uart_dcd# fifo_data[2] fifo_data[6] fifo_rd# spi_m_miso gpio[A2] gpio[A6] gpio[B2] gpio[B6] gpio[C2] gpio[C6] gpio[D2] gpio[D6] gpio[E2] gpio[E6]	uart_rts# fifo_data[2] fifo_data[6] pwm[2] pwm[6] spi_s0_miso spi_s1_miso gpio[A2] gpio[A6] gpio[B2] gpio[B6] gpio[C2] gpio[C6] gpio[D2] gpio[D6] gpio[E2] gpio[E6]	13, 17, 22, 26, 31, 41, 45, 49, 55, 59, 63	13, 18, 22, 33, 37, 43, 47	14, 25, 31

Table 5.4 Group 2

Table 5.4 - Input and output signals that are available for all the IOBUS pins that are in group 2. For example if using the 32 pin package device this would allow pins 14, 25 and 31 to be configured as either an input signal (listed in the first column) or a output signal (listed in the second column).

5.6 I/O Mux Group 3

Available Input signals	Available output signals	64 Pin Package Available pins	48 Pin Package Available pins	32 Pin Package Available pins
uart_cts#				
uart_ri#	fifo_data[3]			
fifo_data[3]	fifo_data[7]			
fifo_data[7]	pwm[3]			
fifo_wr#	pwm[7]			
spi_s0_ss#	spi_m_ss_0#	14, 18, 23, 27, 32, 42, 46, 50, 56, 60, 64	14, 19, 23, 34, 38, 44, 48	15, 26, 32
spi_s1_ss#	gpio[A3]			
gpio[A3]	gpio[A7]			
gpio[A7]	gpio[B3]			
gpio[B3]	gpio[B7]			
gpio[B7]	gpio[C3]			
gpio[C3]	gpio[C7]			
gpio[C7]	gpio[D3]			
gpio[D3]	gpio[D7]			
gpio[D7]	gpio[E3]			
gpio[E3]	gpio[E7]			
gpio[E7]				

Table 5.5 Group 3

Table 5.5 - Input and output signals that are available for all the IOBUS pins that are in group 3. For example if you using the 48 pin package device this would allow pins 14, 19, 23, 34, 38, 44 and 48 to be configured as either an input signal (listed in the first column) or a output signal (listed in the second column).

5.7 I/O Mux Interface Configuration Example

This example shows how to set a UART interface on the VNC2 64 pin package. The UART is made up of two output signals (uart_txd and uart_rts#) and two input signals (uart_rxd and uart_cts#). For PCB design it is best to have the four pins of the UART interface adjacent to each other. This can be achieved easily since the four signals are members of each different groups. **Figure 5.1** clearly shows that the four groups are adjacent to each other. So the four adjacent pins can be used for the UART interface as long as they are selected one from each of the four groups. Tables 9, 10, 11 & 12 can now be used to select where the UART interface can be placed. **Figure 5.6** shows the four UART signal selected on pins 11, 12, 13 & 14 however they could have been selected on any of the other four pins highlighted in blue dashed lines.

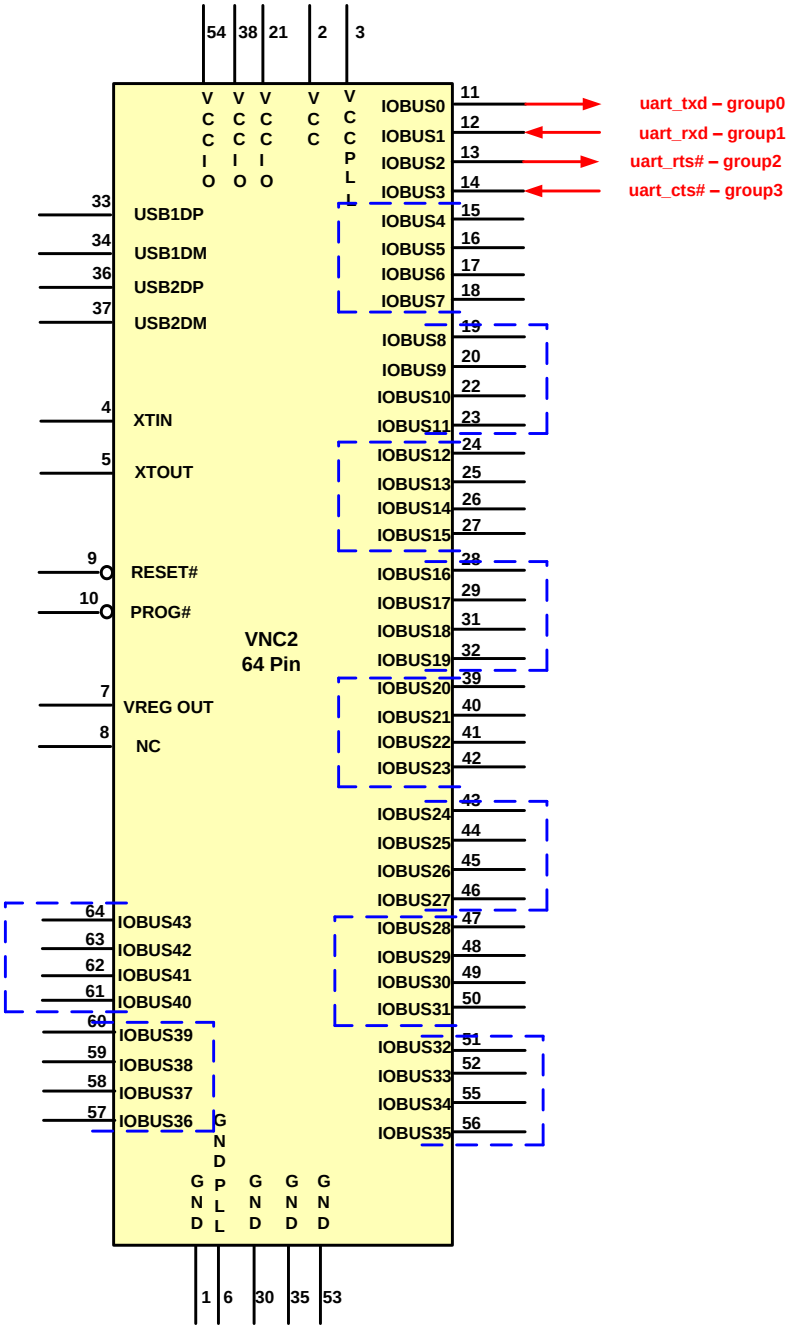

Figure 5.6 UART Example 64 pin

6 Peripheral Interfaces

In addition to the two USB Host and Slave blocks, VNC2 contains the following peripheral interfaces:

- Universal Asynchronous Receiver Transmitter (UART)
- Two Serial Peripheral Interface (SPI) slaves
- SPI Master
- Debugger Interface
- Parallel FIFO Interface (245 mode and synchronous FIFO mode)
- General Purpose Timers
- Eight Pulse Width Modulation blocks (PWM)
- General Purpose Input Output (GPIO)

The following sections describe each peripheral in detail.

6.1 UART Interface

When the data and control bus are configured in UART mode, the interface implements a standard asynchronous serial UART port with flow control, for example RS232/422/485. The UART can support baud rates from 183 baud to 6 Mbaud. The maximum UART speed is determined by the CPU speed/8.The CPU can be run at three frequencies, therefore the following maximum rates apply:

CPU Frequency	Maximum UART Speed
48 MHz	6 Mbaud
24 MHz	3 Mbaud
12 MHz	1.5 Mbaud

Data transfer uses NRZ (Non-Return to Zero) data format consisting of 1 start bit, 7 or 8 data bits, an optional parity bit, and one or two stop bits. When transmitting the data bits, the least significant bit is transmitted first. Transmit and receive waveforms are illustrated in **Figure 6.1** and **Figure 6.2**:

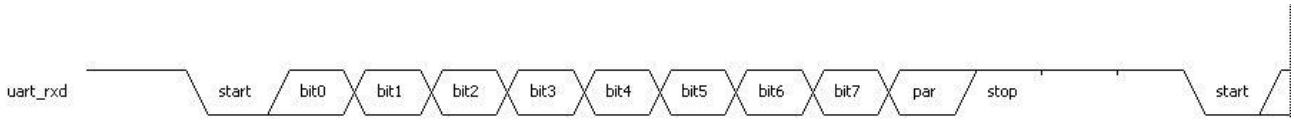

Figure 6.1 UART Receive Waveform

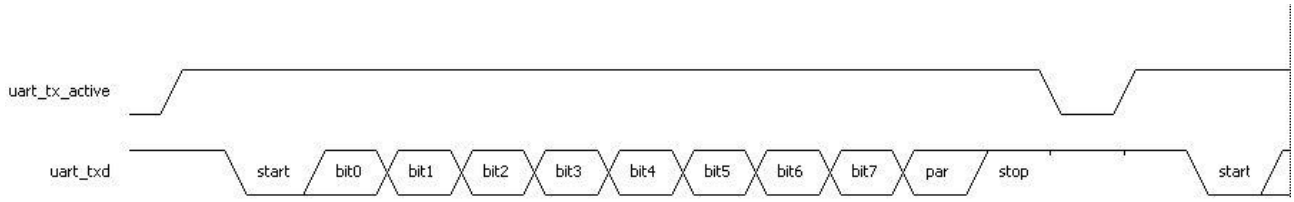

Figure 6.2 UART Transmit Waveform

Baud rate (default =9600 baud), flow control settings (default = RTS/CTS), number of data bits (default=8), parity (default is no parity) and number of stop bits (default=1) are all configurable using the firmware command interface. Please refer to http://www.ftdichip.com.

uart_tx_active is transmit enable, this output may be used in RS485 designs to control the transmit of the line driver.

6.1.1 UART Mode Signal Descriptions

64 Pin Package Available pins	48 Pin Package Available pins	32 Pin Package Available pins	Name	Type	Description
11, 15, 19, 24, 28, 39, 43, 47, 51, 57, 61	11, 15, 20, 31, 35, 41, 45	11, 23 29	uart_txd	Output	Transmit asynchronous data output
12, 16, 20, 25, 29, 40, 44, 48, 52, 58, 62	12,16, 21, 32, 36, 42, 46	12, 24, 30	uart_rxd	Input	Receive asynchronous data input
13, 17, 22, 26, 31, 41, 45, 49, 55, 59, 63	13, 18, 22, 33, 37, 43, 47	14, 25, 31	uart_rts#	Output	Request to send control output
14, 18, 23, 27, 32, 42, 46, 50, 56, 60, 64	14, 19, 23, 34, 38, 44, 48	15, 26, 32	uart_cts#	Input	Clear to send control input
11, 15, 19, 24, 28, 39, 43, 47, 51, 57, 61	11, 15, 20, 31, 35, 41, 45	11, 23 29	uart_dtr#	Output	Data acknowledge (data terminal ready control) output
13, 17, 22, 26, 31, 41, 45, 49, 55, 59, 63	13, 18, 22, 33, 37, 43, 47	14, 25, 31	uart_dcd#	Input	Data carrier detect control input

64 Pin Package Available pins	48 Pin Package Available pins	32 Pin Package Available pins	Name	Type	Description
14, 18, 23, 27, 32, 42, 46, 50, 56, 60, 64	14, 19, 23, 34, 38, 44, 48	15, 26, 32	uart_ri#	Input	Ring indicator is used to wake VNC2 depending on firmware
11, 15, 19, 24, 28, 39, 43, 47, 51, 57, 61	11, 15, 20, 31, 35, 41, 45	11, 23 29	uart_tx_active	Output	Enable transmit data for RS485 designs. This signal may be used to signal that a transmit operation is in progress. The uart_tx_active signal will be set high one bit-time before data is transmitted and return low one bit time after the last bit of a data frame has been transmitted.

Table 6.1 Data and Control Bus Signal Mode Options – UART Interface

The UART signals can be programmed to a choice of I/O pins depending on the package size. **Table 6.1** details the available pins for each of the UART signals. Further details on the configuration of input and output signals are available in **Section 5 - I/O Multiplexer**.

6.2 Serial Peripheral Interface – SPI Modes

The Serial Peripheral Interface Bus is an industry standard communications interface. Devices communicate in Master / Slave mode, with the Master initiating the data transfer.

VNC2 has one master module and two slave modules. Each SPI slave module has four signals – clock, slave select, MOSI (master out – slave in) and MISO (master in – slave out). The SPI Master has the same four signals as the slave modules but with one additional signal because it requires a slave select for the second slave module. **Table 6.2** lists how the signals are named in each module.

The SPI Master clock can operate up to one half of the CPU system clock depending on what power mode the device is set to:

- Normal power mode 48Mhz would set the SPI maximum clock to 24Mhz
- Low power mode 24Mhz would set the SPI maximum clock to 12Mhz
- Lowest power mode 12Mhz would set the SPI maximum clock to 6hMz

Module	Signal Name	Type	Description
SPI Slave 0	spi_s0_clk	Input	Clock input – slave 0
	spi_s0_ss#	Input	Active low chip select input – slave 0
	spi_s0_mosi	Input	Master out serial in – slave 0
	spi_s0_miso	Output	Master in slave out – slave 0
SPI Slave 1	spi_s1_clk	Input	Clock input – slave 1
	spi_s1_ss#	Input	Active low chip select input – slave 1
	spi_s1_mosi	Input	Master out slave in – slave 1
	spi_s1_miso	Output	Master in slave out – slave 1
SPI Master	spi_m_clk	Output	Clock output – master
	spi_m_mosi	Output	Master out slave in - master
	spi_m_miso	Input	Master in slave out - master
	spi_m_ss_0#	Output	Active low slave select 0 from master to slave 0
	spi_m_ss_1#	Output	Active low slave select 1 from master to slave 1

Table 6.2 SPI Signal Names

The SPI slave protocol by default does not support any form of handshaking. FTDI have added extra modes to support handshaking, faster throughput of data and reduced pin count. There are 5 modes (Table 15) of operation in the VNC2 SPI Slave.

- Full Duplex – **Section 0**
- Half Duplex, 4 pin - **Section 6.3.3**
- Half Duplex, 3 pin - **Section 6.3.4**
- Unmanaged - **Section 6.3.5**
- VNC1L legacy mode – **Section 6.3.6**

Mode	Pins	Word Size	Handshaking	Speed	Comments
VNC1L	4	12	Yes	Read 66% Write 66%	Legacy mode
Full Duplex	4	8	Yes	Read 50% Write 100%	
Half Duplex 4 pin	4	8	Yes	Read 100% Write 100%	MOSI becomes bi-directional
Half Duplex 3 pin	3	8	Yes	Read 50% Write 50%	MOSI becomes bi-directional
Unmanaged	4	8	No	Read 100% Write 100%	

Table 6.3 - SPI Slave Speeds

VNC2 SPI Master is described in **Section 6.4.1 SPI Master Signal Descriptions.**

Table 6.5 shows the SPI master signals and the available pins that they can be mapped to depending on the package size. Further details on the configuration of input and output signals are available in **Section 5 - I/O Multiplexer**.

6.2.1 SPI Clock Phase Modes

SPI interface has 4 unique modes of clock phase (CPHA) and clock polarity (CPOL), known as Mode 0, Mode 1, Mode 2 and Mode 3. **Table 6.4** summarizes these modes and available interface and **Figure 6.3** is the function timing diagram.

For CPOL = 0, the base (inactive) level of SCLK is 0.

In this mode:

- When CPHA = 0, data is clocked in on the rising edge of SCLK, and data is clocked out on the falling edge of SCLK.
- When CPHA = 1, data is clocked in on the falling edge of SCLK, and data is clocked out on the rising edge of SCLK

For CPOL =1, the base (inactive) level of SCLK is 1.

In this mode:

- When CPHA = 0, data v in on the falling edge of SCLK, and data is clocked out on the rising edge of SCLK
- When CPHA =1, data is clocked in on the rising edge of SCLK, and data is clocked out on the falling edge of SCLK.

Mode	CPOL	CPHA	Full Duplex	Half Duplex 4 pin	Half Duplex 3 pin	Unmanaged	VNC1L Legacy
0	0	0	N	N	N	Y	N
1	0	1	Y	Y	Y	Y	N
2	1	0	N	N	N	Y	N
3	1	1	Y	Y	Y	Y	N

Table 6.4 - Clock Phase/Polarity Modes

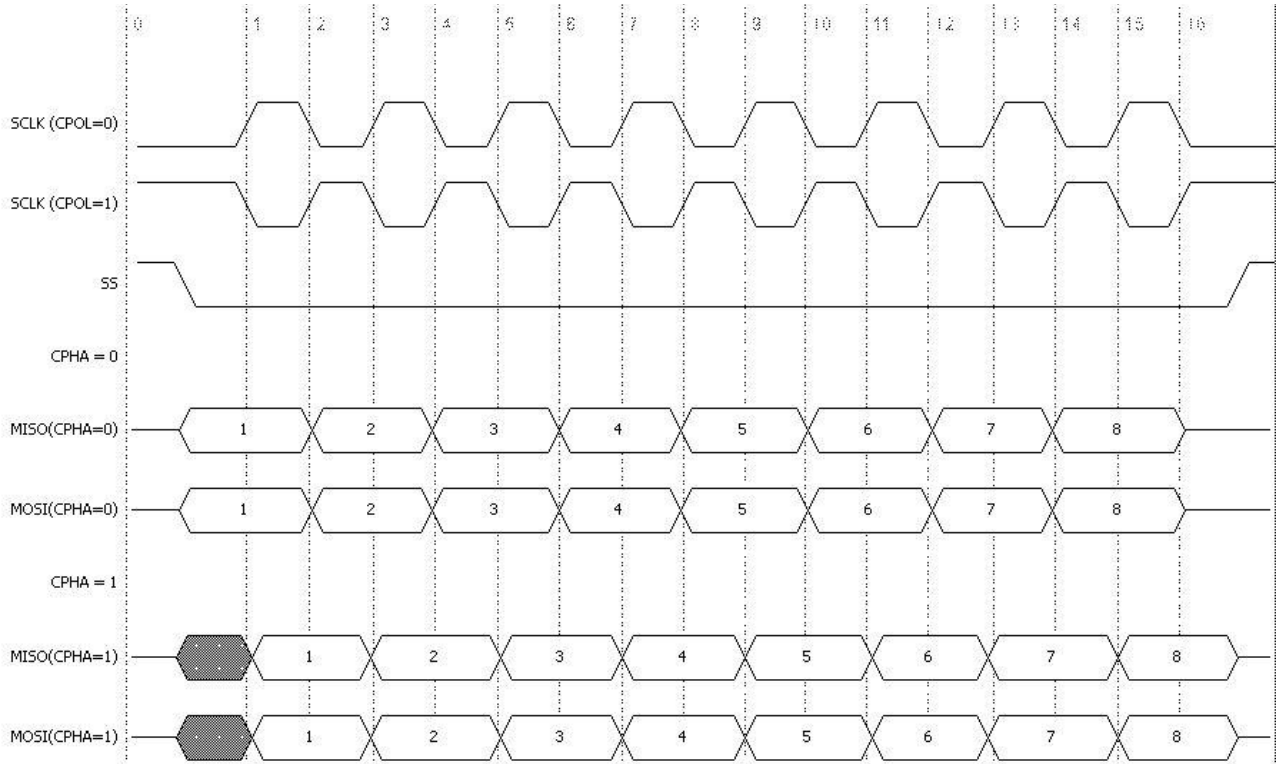

Figure 6.3 - SPI CPOL CPHA Function

6.3 Serial Peripheral Interface – Slave

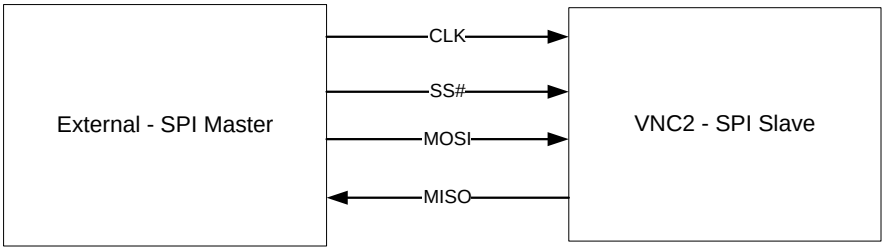

Figure 6.4 SPI Slave block diagram

VNC2 has two SPI Slave modules both of which use four wire interfaces: MOSI, MISO, CLK and SS#. Their main purpose is to send data from main memory to the attached SPI master, and / or receive data and send it to main memory. The SPI Slave is controlled by the internal CPU using internal memory mapped I/O registers. It operates from the main system clock, although sampling of input data and transmission of output data is controlled by the SPI clock (CLK). An SPI transfer can only be initiated by the SPI Master and begins with the slave select signal being asserted. This is followed by a data byte being clocked out with the master supplying CLK. The master always supplies the first byte, which is called a command byte. After this the desired number of data bytes are transferred before the transaction is terminated by the master de-asserting slave select. An SPI Master is able to abort a transfer at any time by de-asserting its SS# output. This will cause the Slave to end its current transfer and return to idle state.

6.3.1 SPI Slave Signal Descriptions

64 Pin Package Available pins	48 Pin Package Available pins	32 Pin Package Available pins	Name	Type	Description
11, 15, 19, 24, 28, 39, 43, 47, 51, 57, 61	11, 15, 20, 31, 35, 41, 45	11, 23 29	spi_s0_clk spi_s1_clk	Input	Slave clock input
12, 16, 20, 25, 29, 40, 44, 48, 52, 58, 62	12,16, 21, 32, 36, 42, 46	12, 24, 30	spi_s0_mosi spi_s1_mosi	Input	Mater Out Slave In Synchronous data from master to slave
13, 17, 22, 26, 31, 41, 45, 49, 55, 59, 63	13, 18, 22, 33, 37, 43, 47	14, 25, 31	spi_s0_miso spi_s1_miso	Output	Master In Slave Out Synchronous data from slave to master
14, 18, 23, 27, 32, 42, 46, 50, 56, 60, 64	14, 19, 23, 34, 38, 44, 48	15, 26, 32	spi_s0_ss# spi_s1_ss#	Input	Slave chip select

Table 6.5 Data and Control Bus Signal Mode Options - SPI Slave Interface

6.3.2 Full Duplex

In full duplex mode, the SPI slave sends data on MISO line at the same time as it receives data on MOSI. During the command phase this data is always the slave status byte. For a write command, write data can be streamed out of MOSI and status can be sent during each write phase from slave to master. As long as the slave status indicates that it can receive more data, the master can continue to stream further write bytes. **Figure 6.5** is an example of this.

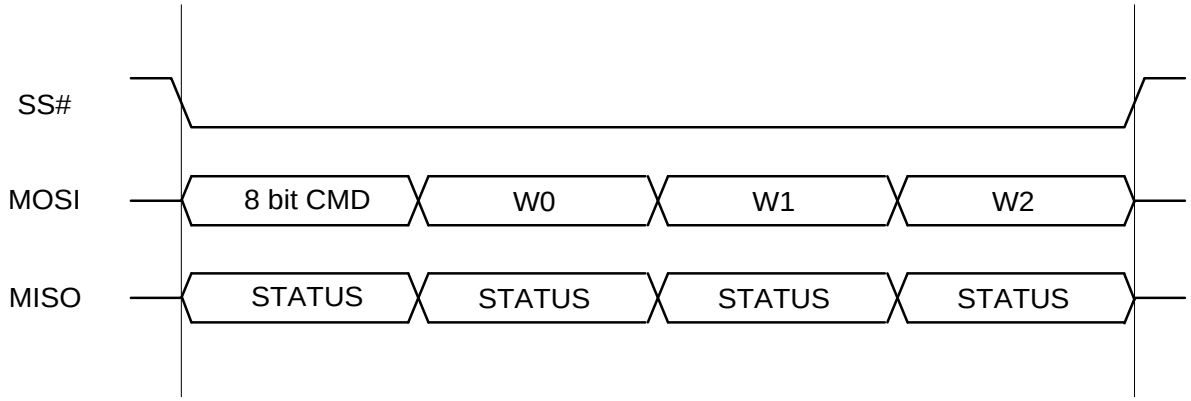

Figure 6.5 Full Duplex Data Master Write

When the master is performing a data read, the data and status both need to share the same pin (MISO). In this case the master and slave will exchange command and status bytes, followed by the slave sending its data. If the Master keeps SS# active the Slave will send a further status byte after the data followed by another data byte. This continues until the Master indicates the end of the communications by raising SS#. **Figure 6.6** is an example of this.

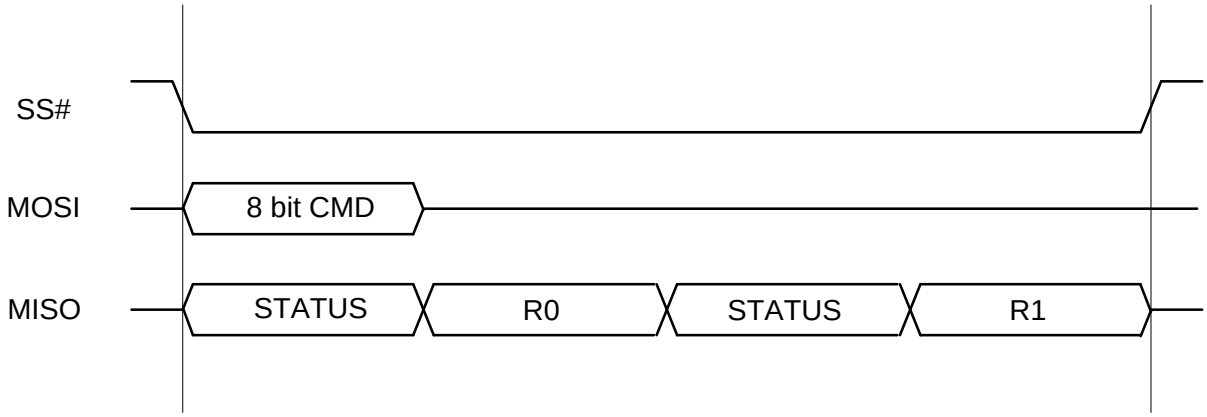

Figure 6.6 Full Duplex Data Master Read

The command and status formats for this mode can be seen in **Figure 6.7** below with a description of each field in **Table 6.6**:

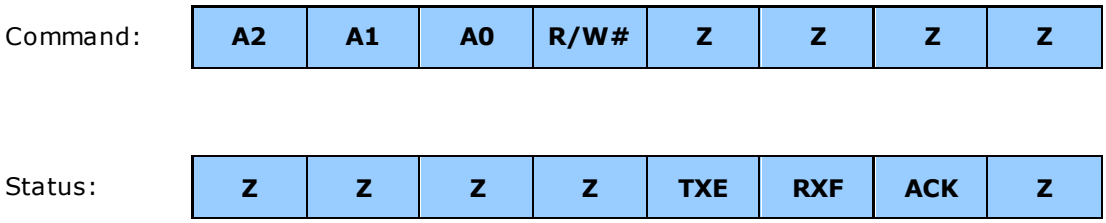

Figure 6.7 SPI Command and Status Structure

Field	Description
A2:A0	Address of slave being used in a multi-slave environment. This would typically be used in the scenario where a shared data bus is used.
R/W#	Set to '1' for a read and '0' for a write.
Z	Tri-stated.
TXE	Transmit Empty. When '1' the Slave transmit buffer has no new data to transmit. When '0' the Slave transmit buffer does have new data.
RXF	Receive Full. When '1' the Slave receive buffer has new data which has not been read yet. When '0' the Slave receive buffer is empty and can be safely written to.
ACK	Set to '1' when a Slave has correctly decoded its address.

Table 6.6 SPI Command and Status Fields

6.3.3 Half Duplex, 4 pin

In half duplex mode, the MOSI signal is shared for both Master to Slave and Slave to Master communications. When using 4 pins, the MISO signal carries the status bits. The Master initiates data write transfer, this by asserting SS# and then sending out a command byte. This has the same format as that shown in **Figure 6.7.** The Slave sends status during this command phase and if this indicates that the Slave can accept data the Master will follow this up with a byte of write data. If the status continues to indicate that more data can be written, a whole stream of data can be written following one single command. The operation completes when the Master raises SS# again. **Figure 6.8** is an example of this.

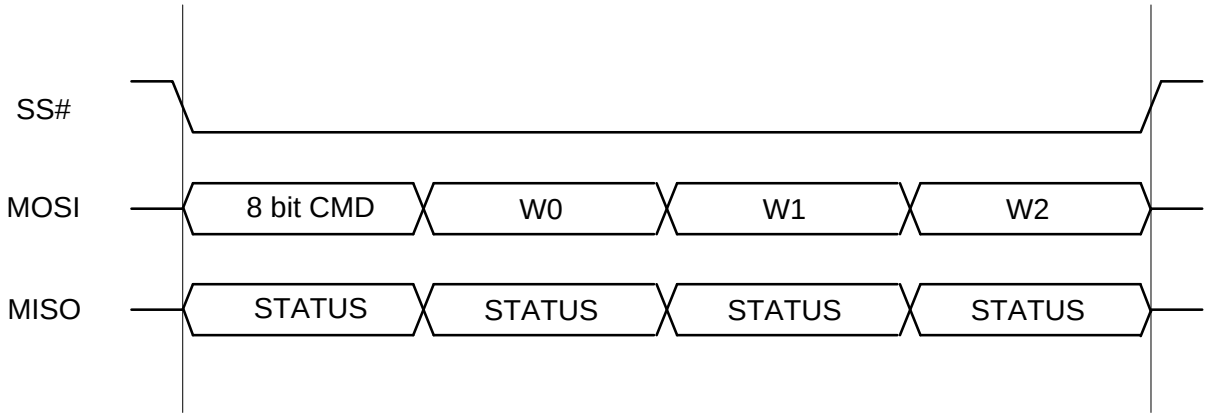

Figure 6.8 Half Duplex Data Master Write

Data reads are similar, apart from the MOSI pin changing from Slave input to Slave output after the command phase. **Figure 6.9** is an example. In this diagram, the Master drives the command while the Slave returns with status. Then the MOSI buffers are turned round and a stream of read data is sent from the Slave to the Master on the MOSI signal.

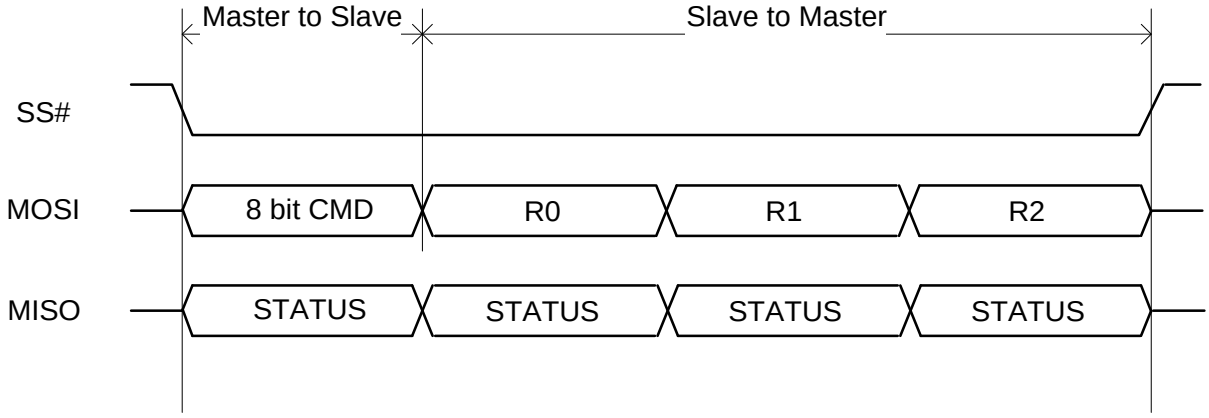

Figure 6.9 Half Duplex Data Master Read

6.3.4 Half Duplex, 3 pin

The 3 pin half duplex mode eliminates the MISO pin from the protocol. This means that status bytes need to be sent on the MOSI pin. Again the Master initiates a transfer by asserting SS# and sending out a command byte. The Slave sends status back to the Master. If a write has been requested and the status indicates that the Slave can accept data, MOSI should be changed to an output again and data will be sent from Master to Slave.

Following this data, the Slave will send a further status byte if SS# remains active. If the status indicates that more data can be written, the next data byte can be sent to the Slave and this process continues until SS# is de-asserted. **Figure 6.10** is an example of this:

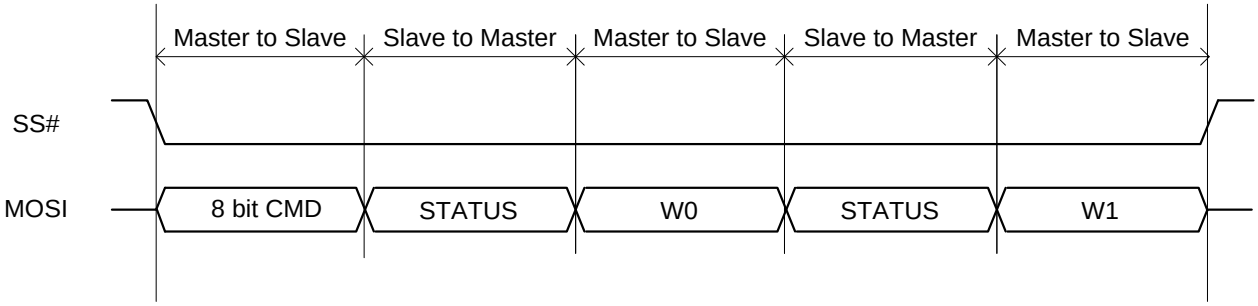

Figure 6.10 Half Duplex 3-pin Data Master Write

Data reads are similar expect that after the command byte all data transfer is from Slave to Master. **Figure 6.11** is an example of this:

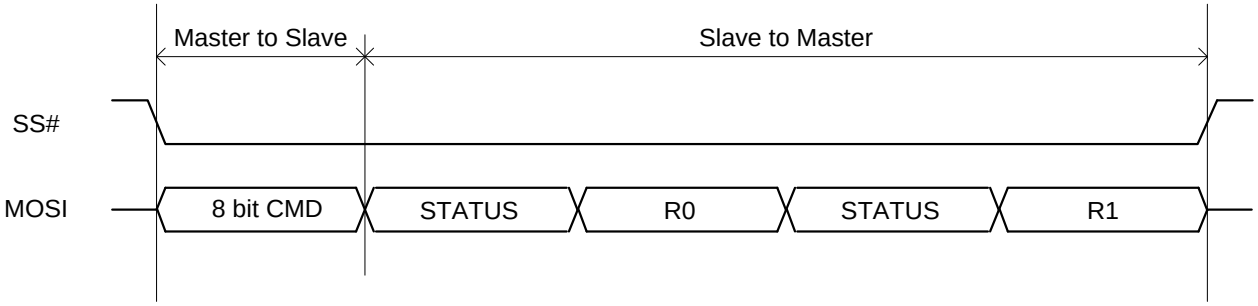

Figure 6.11 Half Duplex 3-pin Data Master Read

6.3.5 Unmanaged Mode

The VNC2 SPI Slave also supports an unmanaged SPI mode. This is a simple data exchange between Master and Slave. It operates in the standard 4 pin mode (SS#, CLK, MOSI and MISO) with all transfers controlled by the SPI Master.

When the CPU wants to send data out of the SPI Slave it writes this into the spi_slave_data_tx register. This will then be moved into the transfer shift register to wait for the SPI Master to request it. The SPI Master will at some point assert SS# and start clocking data on MOSI with SCK. As this is shifted into the transfer shift register, the SPI Slave will also be shifting data in the opposite direction on MISO. At the end of the transfer the SPI Slave copies the received data from the shift register to spi_slave_data_rx as seen in **Figure 6.12.**

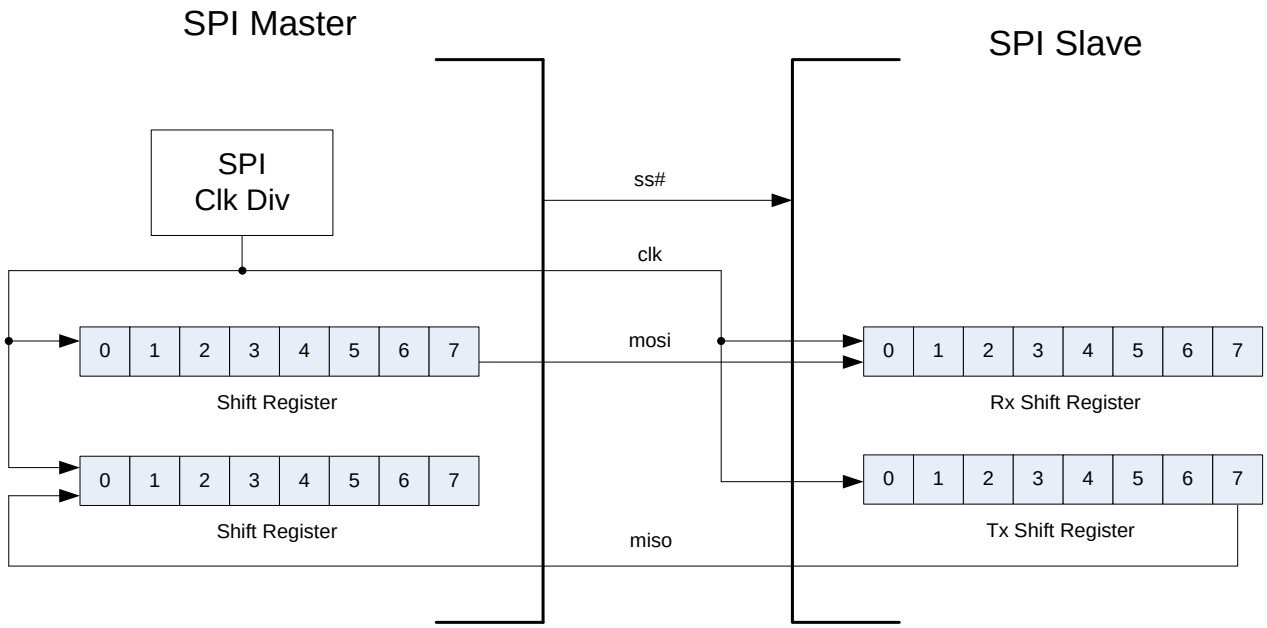

Figure 6.12 Unmanaged Mode Transfer Diagram

6.3.6 VNC1L Legacy Interface

VNC2 SPI is compatible with the SPI slave of VNC1L. This is a custom protocol using 4 wires and will be explained here.

The Master asserts the slave select, but in this case it is an active high signal. Following this, a 3 bit command is sent on the MOSI pin (see **Figure 6.15** for command structure). This has instructions on whether a read or write is requested and if data or status is to be sent. For a data write, 8 bits of data are sent on MOSI followed by a status bit being returned on MISO. If this bit is '0' it means the data write was successful. If it is '1' it means that internal buffer was full and the write should be repeated. Finally, the slave select is de-asserted. See **Figure 6.13** for an example of this.

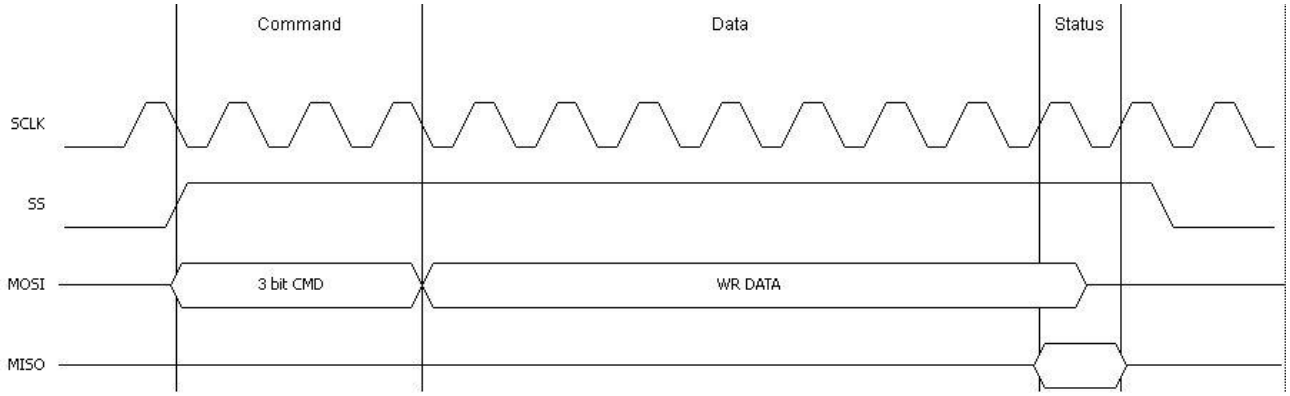

Figure 6.13 VNC1L Mode Data Write

Data reads are similar, with the data from Slave to Master coming on the MISO pin. If the status bit is '0' it means the data byte sent is new data that has not been read before. If it is '1' it means that it is old data. See **Figure 6.14** for an example.

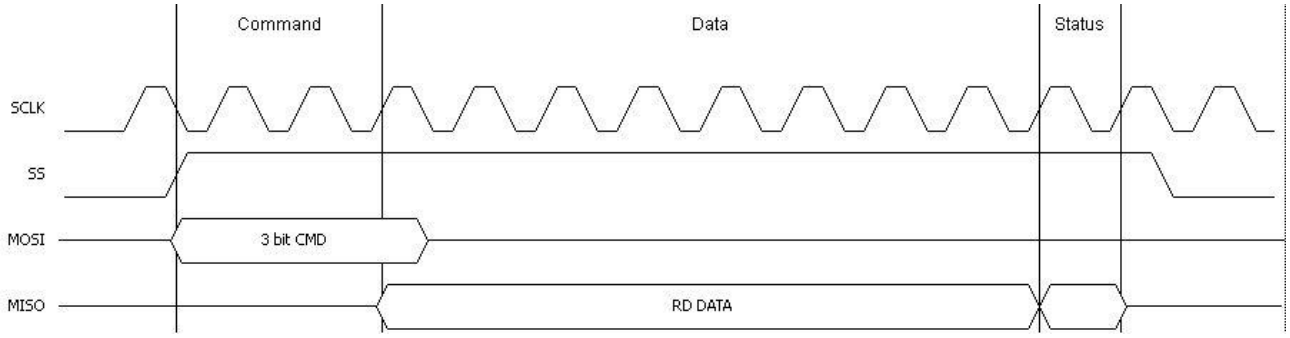

Figure 6.14 VNC1L Mode Data Read

The command and status formats for this mode can be seen in **Figure 6.15** below with a description of each field in **Table 6.7.**

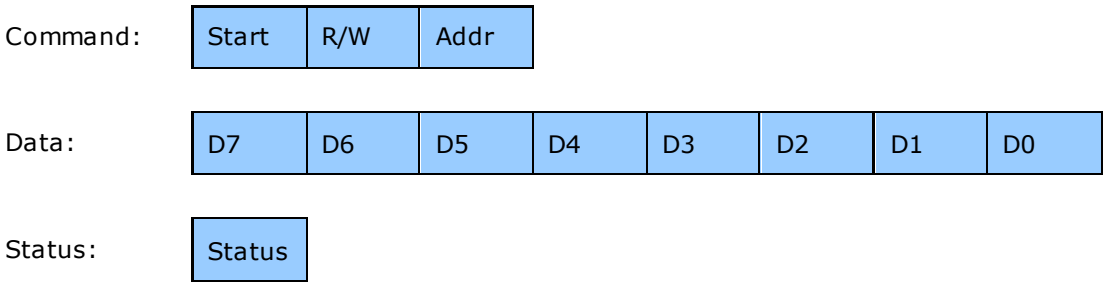

Figure 6.15 VNC1L Compatible SPI Command and Status Structure

Field	Description
Start	Driven to '1'.
R/W	If set to '1', the SPI Master wishes to read from the slave. If set to '0', the SPI Master wishes to write to the slave.
Addr	If set to '1', a read operation will return the status byte in the data phase. A write will have no effect. If set to '0', a read or a write will operate on the data register.
D7:D0	Data.
Status	When '0' this means a read or write was successful. When '1' it means a read contains old data, or a write did not work and needs retried.

Table 6.7 SPI Command and Status Fields

6.3.6.1 SPI Setup Bit Encoding

The VNC1L compatible SPI interface differs from most other implementations in that it uses a 12 clock sequence to transfer a single byte of data. In addition to a 'Start' state, the SPI master must send two setup bits which indicate data direction and target address. The encoding of the setup bits is shown in **Table 6.8**. A single data byte is transmitted in each SPI transaction, with the most significant bit transmitted first.

After each transaction VNC2 returns a single status bit. This indicates if a Data Write was successful or a Data Read was valid.

Direction (R/W)	Target Address	Operation	Meaning
1	0	Data Read	Retrieve byte from Transmit Buffer
1	1	Status Read	Read SPI Interface Status
0	0	Data Write	Add byte to Receive Buffer
0	1	N/A	N/A

Table 6.8 SPI Setup Bit Encoding

The VNC2 SPI interface uses 4 signal lines: SCLK, SS, MOSI and MISO. The signals MOSI, MISO and SS are always clocked on the rising edge of the SCLK signal.

SS signal must be raised high for the duration of the entire transaction. For data transactions, the SS must be released for at least one clock cycle after a transaction has completed. It is not necessary to release SS between Status Read operations.

The 'Start' state of MOSI and SS high on the rising edge of SCLK initiates the transfer. The transfer finishes after 13 clock cycles, and the next transfer starts when MOSI is high during the rising edge of CLK.

The following **Figure 6.16** and
Table 6.9 give details of the bus timing requirements.

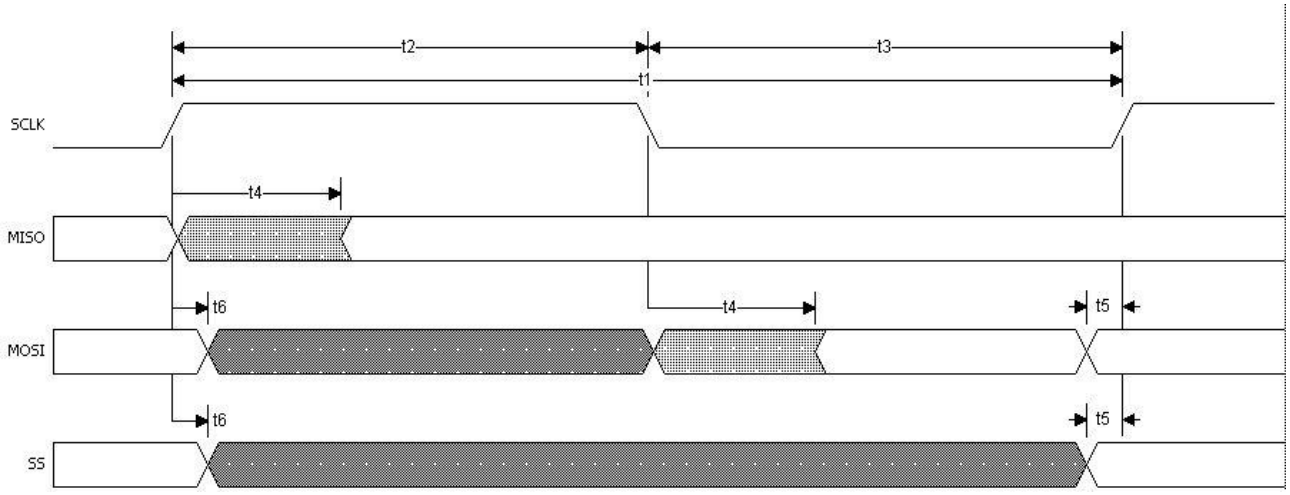

Figure 6.16 SPI Slave Mode Timing

Time	Description	Minimum	Typical	Maximum	Unit
T1	SCLK period	79.37	83.33		ns
T2	SCLK high period	39.68	41.67	39.68	ns
T3	SCLK low period	39.68	41.67	39.68	ns
T4	SCLK driving edge to MISO/MOSI	0.5		14	ns
T5	MISO/SS setup time to sample SCLK edge			3	ns
T6	MISO/SS hold time from sample SCLK edge	3			ns

Table 6.9 SPI Slave Data Timing

6.3.6.2 SPI Master Data Read Transaction in VNC1L legacy mode

The SPI master must periodically poll for new data in VNC2 Transmit Buffer. It is recommended that this is done first before sending any command.

The Start and Setup sequence is sent to VNC2 by the SPI master, see **Figure 6.17.**

The VNC2 clocks out data from its Transmit Buffer on subsequent rising edge clock cycles provided by the SPI master. This is followed by a status bit generated by VNC2. The Data Read status bit is defined in **Table 6.10**.

If the status bit indicates New Data then the byte received is valid. If it indicates Old Data then the Transmit Buffer in VNC2 is empty and the byte of data received in the current transaction should be disregarded.

Status Bit	Meaning	
0	New Data	Data in current transaction is valid data. Byte removed from Transmit Buffer.
1	Old Data	This same data has been read in a previous read cycle. Repeat the read cycle until New Data is received.

Table 6.10 SPI Master Data Read Status Bit

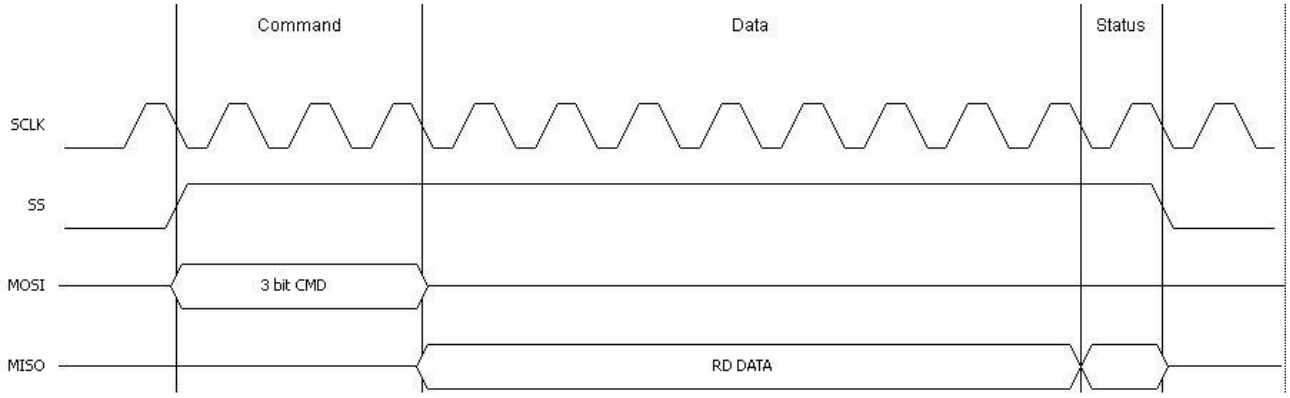

Figure 6.17 SPI Master Data Read (VNC2 Slave Mode)

The status bit is only valid until the next rising edge of SCLK after the last data bit.

During the Data Read operation the SS signal must not be de-asserted.

The transfer completes after 12 clock cycles and the next transfer can begin when MOSI and SS are high during the rising edge of SCLK.

6.3.6.3 SPI Master Data Write Transaction in VNC1L legacy mode

During an SPI master Data Write operation the Start and Setup sequence is sent by the SPI master to VNC2, see **Figure 6.18**. This is followed by the SPI master transmitting each bit of the data to be written to VNC2. The VNC2 then responds with a status bit on MISO on the rising edge of the next clock cycle.

The SPI master must read the status bit at the end of each write transaction to determine if the data was written successfully to VNC2 Receive Buffer. The Data Write status bit is defined in **Table 6.11**.The status bit is only valid until the next rising edge of SCLK after the last data bit.

If the status bit indicates Accept then the byte transmitted has been added to VNC2 Receive Buffer. If it shows Reject then the Receive Buffer is full and the byte of data transmitted in the current transaction should be re-transmitted by the SPI master to VNC2.

Any application should poll VNC2 Receive Buffer by retrying the Data Write operation until the data is accepted.

Status Bit	Meaning	
0	Accept	Data from the current transaction was accepted and added to the Receive Buffer
1	Reject	Write data was not accepted. Retry the same write cycle.

Table 6.11 SPI Master Data Write Status Bit

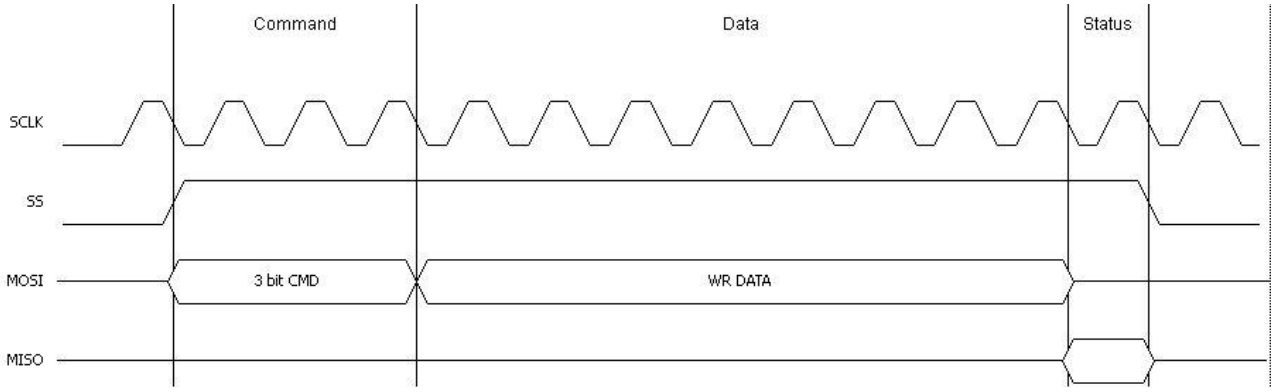

Figure 6.18 SPI Slave Mode Data Write

6.3.6.4 SPI Master Status Read Transaction in VNC1L legacy mode

The VNC2 has a status byte which determines the state of the Receive and Transmit Buffers. The SPI master must poll VNC2 and read the status byte.

The Start and Setup sequence is sent to VNC2 by the SPI master, see **Figure 6.19**. The VNC2 clocks out its status byte on subsequent rising edge clock cycles from the SPI master. This is followed by a status bit generated by VNC2 (also on the MISO) which will always be zero (indicating new data).

The meaning of the bits within the status byte sent by VNC2 during a Status Read operation is described in **Table 6.12**. The result of the Status Read transaction is only valid during the transaction itself. Data read and data write transactions must still check the status bit during a Data Read or Data Write cycle regardless of the result of a Status Read operation.

Bit	Description	Description
0	RXF#	Receive Buffer Full
1	TXE#	Transmit Buffer Empty
2	-	Not used
3	-	Not used
4	RXF IRQEn	Receive Buffer Full Interrupt Enable
5	TXE IRQEn	Transmit Buffer Empty Interrupt Enable
6	-	Not used
7	-	Not used

Table 6.12 SPI Status Read Byte – bit descriptions

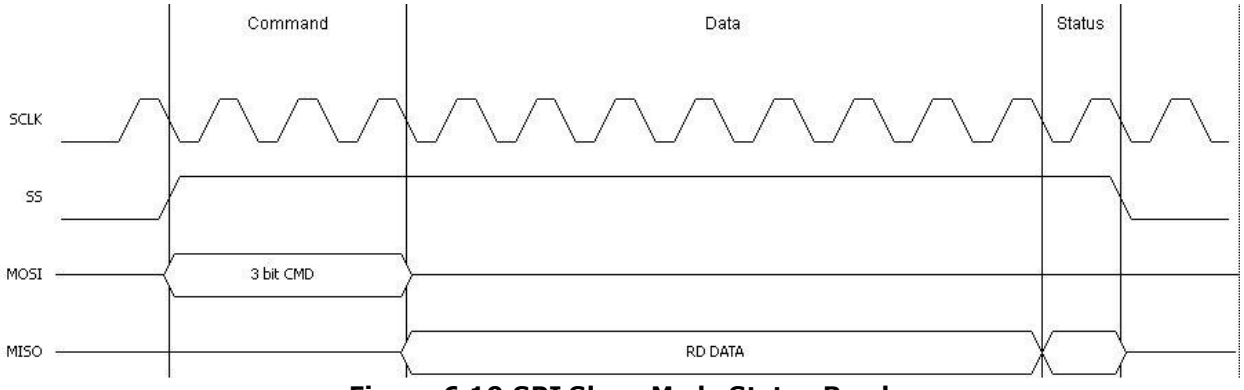

Figure 6.19 SPI Slave Mode Status Read

6.4 Serial Peripheral Interface – SPI Master

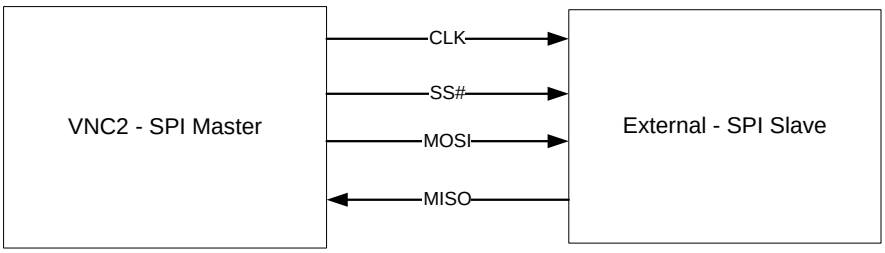

Figure 6.20 SPI Master block diagram

The SPI Master interface is used to interface to applications such as SD Cards. The SPI Master provides the following features:

- Synchronous serial data link.
- Full and half duplex data transmission.
- Serial clock with programmable frequency, polarity and phase.
- One slave select output.
- Programmable delay between negative edge of slave select and start of transfer.
- SD Card interface.
- An interface that's compatible with the VLSI VS1033 SCI mode used for VMUSIC capability

The SPI Master only clocks in and out data that the VNC2 CPU sets up in its register space. The VNC2 CPU interprets the data words that are to be sent and received.

6.4.1 SPI Master Signal Descriptions.

Table 6.13 shows the SPI master signals and the available pins that they can be mapped to depending on the package size. Further details on the configuration of input and output signals are available in **Section 5 - I/O Multiplexer**.

64 Pin Package Available pins	48 Pin Package Available pins	32 Pin Package Available pins	Name	Type	Description
11, 15, 19, 24, 28, 39, 43, 47, 51, 57, 61	11, 15, 20, 31, 35, 41, 45	11, 23 29	spi_m_clk	Output	SPI master clock input

64 Pin Package Available pins	48 Pin Package Available pins	32 Pin Package Available pins	Name	Type	Description
12, 16, 20, 25, 29, 40, 44, 48, 52, 58, 62	12,16, 21, 32, 36, 42, 46	12, 24, 30	spi_m_mosi	Output	Master Out Slave In Synchronous data from master to slave
13, 17, 22, 26, 31, 41, 45, 49, 55, 59, 63	13, 18, 22, 33, 37, 43, 47	14, 25, 31	spi_m_miso	Input	Master In Slave Out Synchronous data from slave to master
14, 18, 23, 27, 32, 42, 46, 50, 56, 60, 64	14, 19, 23, 34, 38, 44, 48	15, 26, 32	spi_m_ss_0#	Output	Active low slave select 0 from master to slave 0
11, 15, 19, 24, 28, 39, 43, 47, 51, 57, 61	11, 15, 20, 31, 35, 41, 45	11, 23 29	spi_m_ss_1#	Output	Active low slave select 1 from master to slave 1

Table 6.13 SPI Master Signal Names

The main purpose of the SPI Master block is to transfer data between an external SPI interface and the VNC2. It does this under the control of the CPU and DMA engine via the on chip I/O bus.

An SPI master interface transfer can only be initiated by the SPI Master and begins with the slave select signal being asserted. This is followed by a data byte being clocked out with the master supplying SCLK. The master always supplies the first byte, which is called a command byte. After this the desired number of data bytes are transferred before the transaction is terminated by the master de-asserting slave select.

The SPI Master will transmit on MOSI as well as receive on MISO during every data stage. At the end of each byte spi_tx_done and spi_rx_full_int are set. **Figure 6.21 Typical SPI Master Timing** and **Table 6.14 SPI Master** Timing show an example of this.

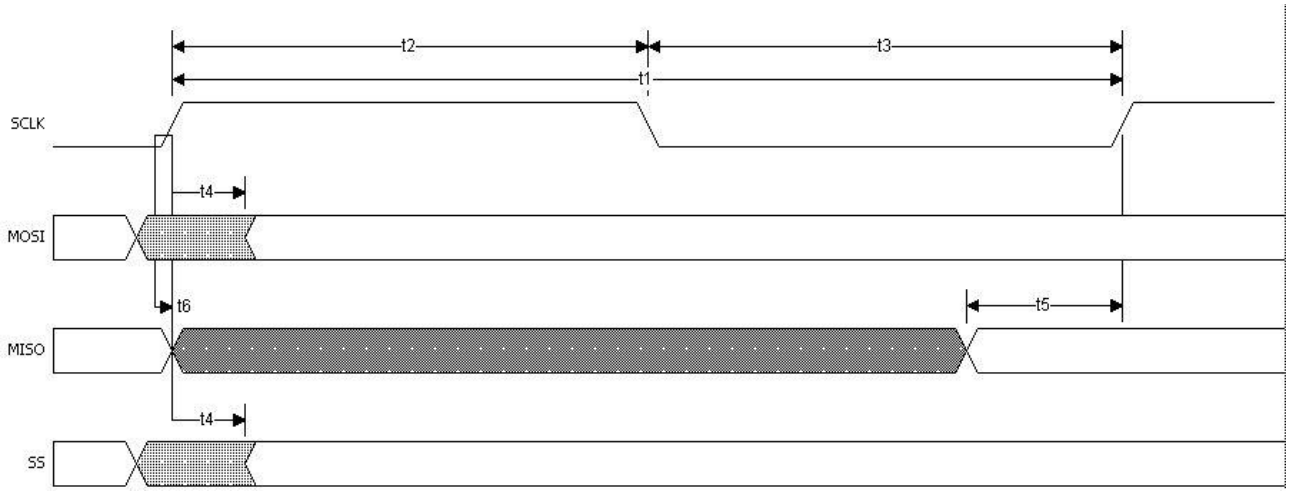

Figure 6.21 Typical SPI Master Timing

Time	Description	Minimum	Typical	Maximum	Unit
t1	SCLK period	39.68	41.67		ns
t2	SCLK high period	19.84	20.84	21.93	ns
t3	SCLK low period	19.84	20.84	21.93	ns
t4	SCLK driving edge to MOSI/SS	-1.5		3	ns
t5	MISO setup time to sample SCLK edge			6.5	ns
t6	MISO hold time from sample SCLK edge	0			ns

Table 6.14 SPI Master Timing

6.5 Debugger Interface

The purpose of the debugger interface is to provide the Integrated Development Environment (IDE) with the following capabilities:

- Flash Erase, Write and Program.
- Application debug - application code can have breakpoints, be single stepped and can be halted.
- Detailed internal debug - memory read/write access.

The single wire interface has the following features:

- Half Duplex Operation
- 1Mbps speed
- 1 start bit
- 1 stop bit
- 8 data bits
- Pull up

Further information of the Debugger Interface is available in an Application Note AN_138 Vinculum-II Debug Interface Description.

6.5.1 Debugger Interface Signal description

64 Pin Package Available pins	48 Pin Package Available pins	32 Pin Package Available pins	Name	Type	Description
11, 15, 19, 24, 28, 39, 43, 47, 51, 57, 61	11, 15, 20, 31, 35, 41, 45	11, 23 29	debug_if	Input/ Output	Debugger Interface

Table 6.15 Debugger Signal Name

6.6 Parallel FIFO – Asynchronous Mode

Parallel FIFO Asynchronous mode known as '245', is functionally the same as the one that is present in VNC1L has an eight bit data bus, individual read and write strobes and two hardware flow control signals.

6.6.1 FIFO Signal Descriptions

The Parallel FIFO interface signals are described in **Table 6.16** They can be programmed to a choice of I/O pins depending on the package size. Further details on the configuration of input and output signals are available in **Section 5 - I/O Multiplexer**.

64 Pin Package Available pins	48 Pin Package Available pins	32 Pin Package Available pins	Name	Type	Description
11, 15, 19, 24, 28, 39, 43, 47, 51, 57, 61	11, 15, 20, 31, 35, 41, 45	11, 23 29	fifo_data[0]	I/O	FIFO Data Bus Bit 0
12, 16, 20, 25, 29, 40, 44, 48, 52, 58, 62	12,16, 21, 32, 36, 42, 46	12, 24, 30	fifo_data[1]	I/O	FIFO Data Bus Bit 1
13, 17, 22, 26, 31, 41, 45, 49, 55, 59, 63	13, 18, 22, 33, 37, 43, 47	14, 25, 31	fifo_data[2]	I/O	FIFO Data Bus Bit 2
14, 18, 23, 27, 32, 42, 46, 50, 56, 60, 64	14, 19, 23, 34, 38, 44, 48	15, 26, 32	fifo_data[3]	I/O	FIFO Data Bus Bit 3

64 Pin Package Available pins	48 Pin Package Available pins	32 Pin Package Available pins	Name	Type	Description
11, 15, 19, 24, 28, 39, 43, 47, 51, 57, 61	11, 15, 20, 31, 35, 41, 45	11, 23 29	fifo_data[4]	I/O	FIFO Data Bus Bit 4
12, 16, 20, 25, 29, 40, 44, 48, 52, 58, 62	12,16, 21, 32, 36, 42, 46	12, 24, 30	fifo_data[5]	I/O	FIFO Data Bus Bit 5
13, 17, 22, 26, 31, 41, 45, 49, 55, 59, 63	13, 18, 22, 33, 37, 43, 47	14, 25, 31	fifo_data[6]	I/O	FIFO Data Bus Bit 6
14, 18, 23, 27, 32, 42, 46, 50, 56, 60, 64	14, 19, 23, 34, 38, 44, 48	15, 26, 32	fifo_data[7]	I/O	FIFO Data Bus Bit 7
11, 15, 19, 24, 28, 39, 43, 47, 51, 57, 61	11, 15, 20, 31, 35, 41, 45	11, 23 29	fifo_rxf#	Output	When high, do not read data from the FIFO. When low, there is data available in the FIFO which can be read by strobing fifo_rd# low, then high.

64 Pin Package Available pins	48 Pin Package Available pins	32 Pin Package Available pins	Name	Type	Description
12, 16, 20, 25, 29, 40, 44, 48, 52, 58, 62	12,16, 21, 32, 36, 42, 46	12, 24, 30	fifo_txe#	Output	When high, do not write data into the FIFO. When low, data can be written into the FIFO by strobing fifo_wr# high, then low.
13, 17, 22, 26, 31, 41, 45, 49, 55, 59, 63	13, 18, 22, 33, 37, 43, 47	14, 25, 31	fifo_rd#	Input	Enables the current FIFO data byte on D0...D7 when low. Fetches the next FIFO data byte (if available) from the receive FIFO buffer when fifo_rd# goes from high to low
14, 18, 23, 27, 32, 42, 46, 50, 56, 60, 64	14, 19, 23, 34, 38, 44, 48	15, 26, 32	fifo_wr#	Input	Writes the data byte on the D0...D7 pins into the transmit FIFO buffer when fifo_wr# goes from high to low.

Table 6.16 Data and Control Bus Signal Mode Options - Parallel FIFO Interface

6.6.2 Read / Write Transaction Asynchronous FIFO Mode

When in Asynchronous FIFO interface mode, the timing of read and write operations on the FIFO interface are shown in **Figure 6.22** and **Table 6.17.**

In asynchronous mode an external device can control data transfer driving FIFO_WR# and FIFO_RD# inputs. In contrast to synchronous mode, in asynchronous mode the 245 FIFO module generates the output enable EN# signal. EN# signal is effectively the read signal RD#.

Current byte is available to be read when FIFO_RD# goes low. When FIFO_RD# goes high, FIFO_RXF# output will also go high. It will only become low again when there is another byte to read.

When FIFO_WR# goes low FIFO_TXE# flag will always go high. FIFO_TXE# goes low again only when there is still space for data to be written in to the module.

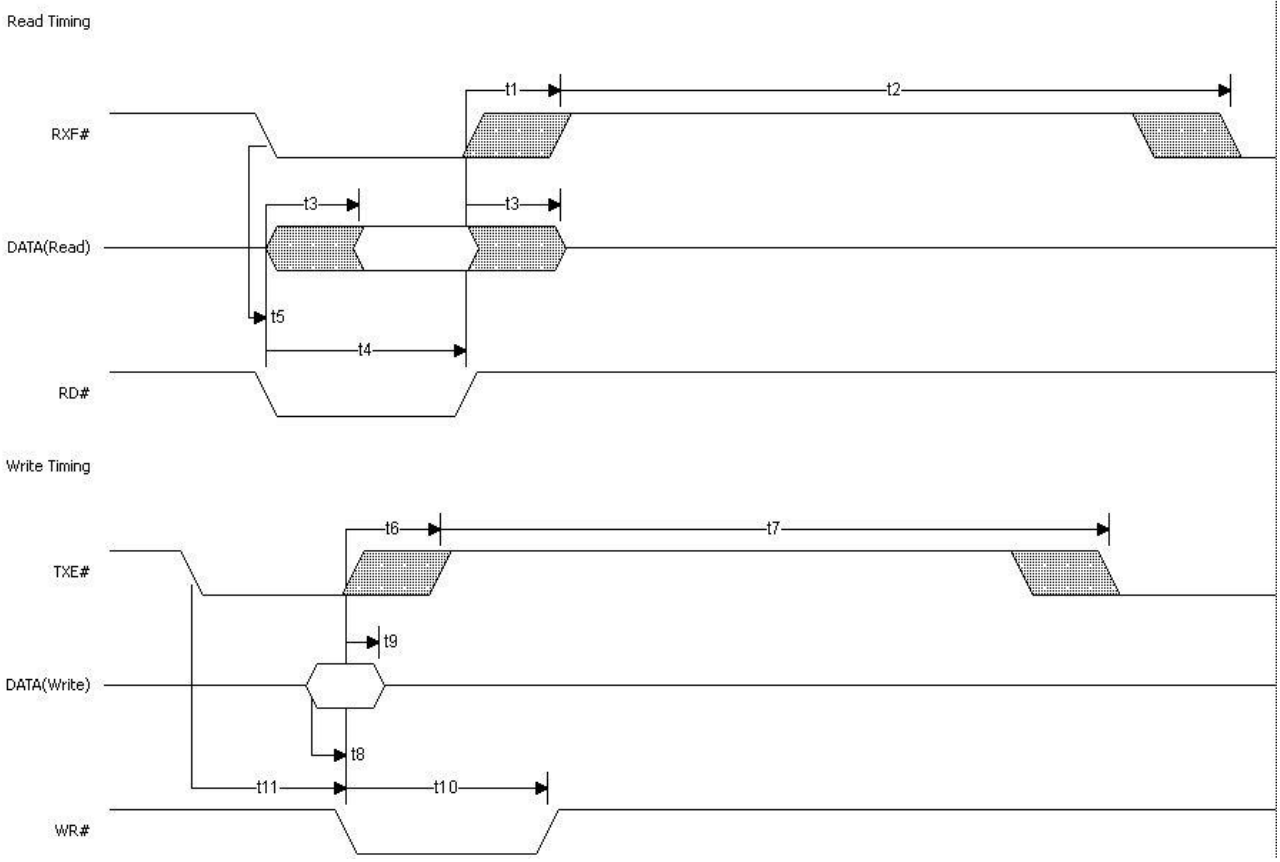

Figure 6.22 Asynchronous FIFO mode Read / Write Cycle

Time	Description	Minimum	Maximum	Unit
t1	RD# inactive to RXF#	1	14	ns
t2	RXF# inactive after RD# cycle	100		ns
t3	RD# to DATA	1	14	ns
t4	RD# active pulse width	30		ns
t5	RD# active after RXF#	0		ns
t6	WR# active to TXE# inactive	1	14	ns
t7	TXE# inactive after WR# cycle	100		ns
t8	DATA to TXE# active setup time	5		ns
t9	DATA hold time after WR# inactive	5		ns
t10	WR# active pulse width	30		ns
t11	WR# active after TXE#	0		ns

Table 6.17 Asynchronous FIFO mode Read / Write Timing

6.7 Parallel FIFO – Synchronous Mode

The Parallel FIFO Synchronous mode has an eight bit data bus, individual read and write strobes, two hardware flow control signals, an output enable and a clock out.

The synchronous FIFO mode uses the parallel FIFO interface signals detailed in **Table 6.16** and an additional two signals detailed in **Table 6.18**.

This mode is not available on the 32 pin packages.

64 Pin Package Available pins	48 Pin Package Available pins	32 Pin Package Available pins	Name	Type	Description
11, 15, 19, 24, 28, 39, 43, 47, 51, 57, 61	11, 15, 20, 31, 35, 41, 45	11, 23 29	fifo_oe#	I/O	FIFO Output enable
12, 16, 20, 25, 29, 40, 44, 48, 52, 58, 62	12,16, 21, 32, 36, 42, 46	12, 24, 30	fifo_clkout	I/O	FIFO Clock out

Table 6.18 Synchronous FIFO control signals

6.7.1 Read / Write Transaction Synchronous FIFO Mode

When in Synchronous FIFO interface mode, the timing of read and write operations on the FIFO interface are shown in **Figure 6.23 Synchronous FIFO mode Read / Write Cycle** and **Table 6.19 Synchronous FIFO mode Read / Write Timing**

In synchronous mode data can be transmitted to and from the FIFO module on each clock edge. An external device synchronises to the CLKOUT output and it also has access to the output enable OE# input to control data flow. An external device should drive output enable OE# low before pulling RD# line down.

When bursts of data are to be read from the module RD# should be kept low. RXF# remains low when there is still data to be read. Similarly when bursts of data are to be written to the module WR# should be kept low. TXE# remains low when there is still space available for the data to be written.

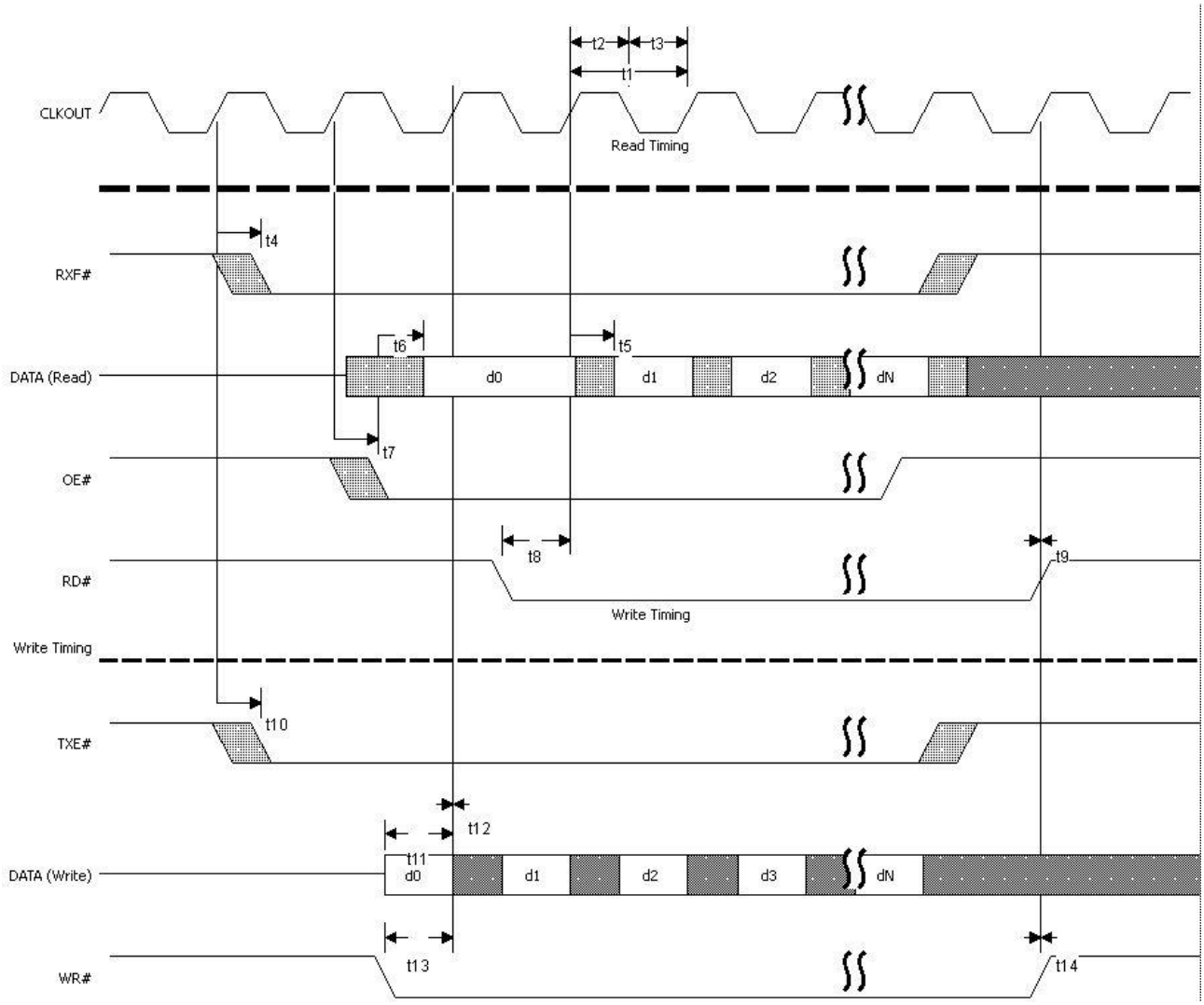

Figure 6.23 Synchronous FIFO mode Read / Write Cycle

Time	Description	Minimum	Typical	Maximum	Unit
t1	CLKOUT period		20.83		ns
t2	CLKOUT high period	9.38	10.42	11.46	ns
t3	CLKOUT low period	9.38	10.42	11.46	ns
t4	CLKOUT to RXF#	1		7.83	ns
t5	CLKOUT to read DATA valid	1		7.83	ns
t6	OE# to read DATA valid	1		7.83	ns
t7	CLKOUT to OE#	1		7.83	ns
t8	RD# setup time	12			ns
t9	RD# hold time	0			ns
t10	CLKOUT TO TXE#	1		7.83	ns
t11	Write DATA setup time	12			ns
t12	Write DATA hold time	0			ns
t13	WR# setup time	12			ns
t14	WR# hold time	0			ns

Table 6.19 Synchronous FIFO mode Read / Write Timing

6.8 General Purpose Timers

In VNC2 there are 4 General Purpose Timers available. Three are available to the designer and one is reserved for the RTOS.

The timers have the following features:

- 16 bit
- Count down
- One shot and auto-reload
- enable
- Interrupt on zero

6.9 Pulse Width Modulation

VNC2 provides 8 Pulse Width Modulation (PWM) outputs. These can be used to generate PWM signals which can be used to control motors, DC/DC converters, AC/DC supplies, etc. Further information is available in an Application Note AN_140 - Vinculum-II PWM Example.

The features of the PWM module are as follows:

- 8 PWM outputs
- A trigger input
- 8-bit prescaler
- 16-bit counter

- Generation of up to 4-pulse signal with controlled output enable and configurable initial state
- Interrupt

A single PWM cycle can have up to 4 pulses (8 edges). The PWM block uses a 16-bit counter to determine the period of a single PWM cycle. This counter counts system clocks which can also be divided by an optional 8-bit prescaler. The PWM drivers allow the user to select when PWM output toggles. These values correspond to the values of 16-bit counter. For example, on the timing diagram below - **Figure 6.24**, the 16-bit counter counts to 23 and pwm_out[0] output toggles when the counter's current value is equal to 7, 8, 12, 14, 15, 16, 19 and 22.

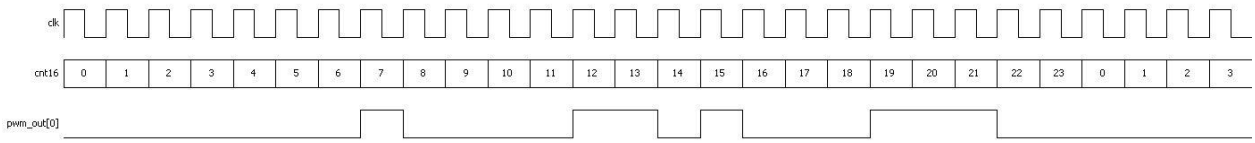

Figure 6.24 PWM – Timing Diagram

The user can also select the initial state of each of the PWM outputs (HI or LOW). PWM outputs can also be enabled continuously or a cycle can be repeated 1..255 times. The PWM cycle can be started by the PWM driver or externally using a trigger input.

6.10 General Purpose Input Output

VNC2 provides up to 40 configurable Input/output pins depending on the package. The Input/output pins are connected to Ports A through E. These ports are controlled by the VNC2 CPU. All ports are configurable to be either inputs or outputs and allow level or edge driven interrupts to be generated.

To simplify the use of the 40 available GPIO signals, they have been grouped into 5 "ports", identified as A, B, C, D and E. Each port is 1 byte wide and the RTOS drivers will allow each port to be individually accessed.

Each GPIO signal is mapped on to a bit of the port value. For example, gpio[A0] is the least significant bit of the value read from or written to GPIO port A. Similarly, gpio[A7] is the most significant bit of the value read from or written to GPIO port A (see **Figure 6.25 GPIO Port Groups**)

Each pin can be individually configured as input or output. GPIO port A supports an interrupt that can be used to detect a state change of any of its 8 pins. Port B features a more sophisticated set of 4 configurable interrupts that can be associated with individual pins and supports several conditions such as positive edge, negative edge, high or low.

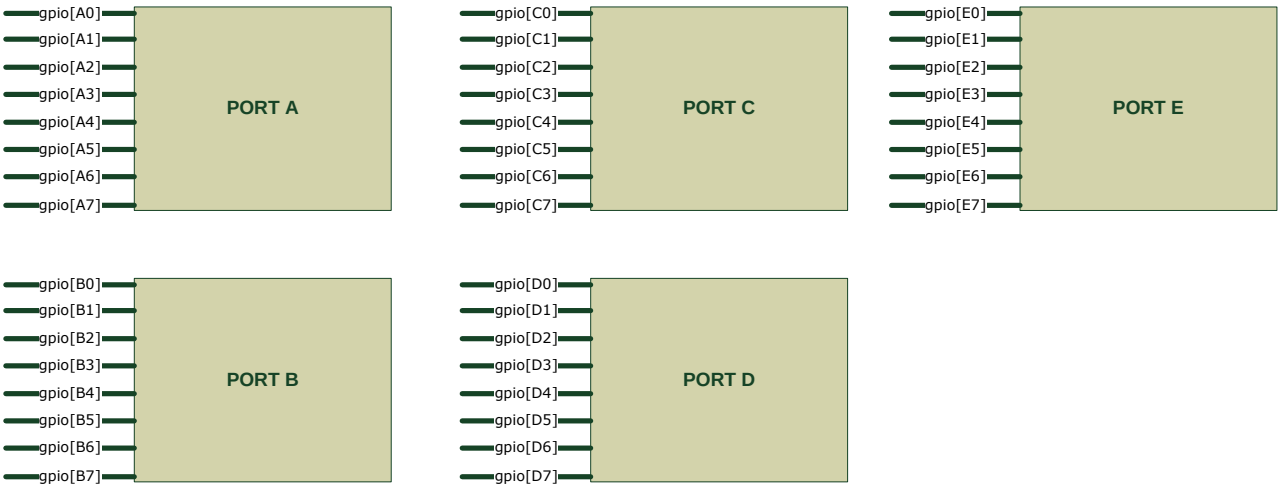

Figure 6.25 GPIO Port Groups

7 USB Interfaces

VNC2 has two USB 1.1 and USB 2.0 compliant interfaces available either as a USB host or slave device capable of supporting 1.5Mb/s (Low Speed) and 12Mb/s (full Speed) transactions. The USB specification defines 4 transfer types that are all supported by VNC2:

- Interrupt transfer: Used for legacy devices where the device is periodically polled to see if the device has data to transfer e.g. Mouse, Keyboard. VNC2 interrupt transfers are valid for low- and full-speed transactions.
- Bulk transfer: Used for transferring large blocks of data that have no periodic or transfer rate requirement e.g. USB to RS232 (FT232R device), memory sticks. VNC2 bulk transfers are only valid for full-speed transactions.
- Isochronous transfer: Used for transferring data that requires a constant delivery rate e.g. web cam, wireless modem. VNC2 isochronous transfers are only valid for full-speed transactions.
- Control transfer: Used to transfer specific requests to all types USB devices (most commonly used during device configuration). VNC2 control transfers are valid for low- and full-speed transfers.

USB 2.0 - 480Mb/s (High Speed) transactions are not supported as the power requirements are deemed excessive for VNC2 target applications. VNC2 configured to Full speed is supported.

VNC2 has two main USB modes of operation: host mode or client (or Slave) mode. As a client, VNC2 is able to connect to a PC and act as a USB peripheral. At the same time as being a client the second USB interface is also able to act as a host and connect to a second USB device using two separate ports (i.e. Port 0 – Host Port 1- Client). Each USB interface can be either a host or a client. It is not possible to change from host to client or client to host "on-the-fly". The following diagrams in figure 7.1 give examples of possible modes of operation:

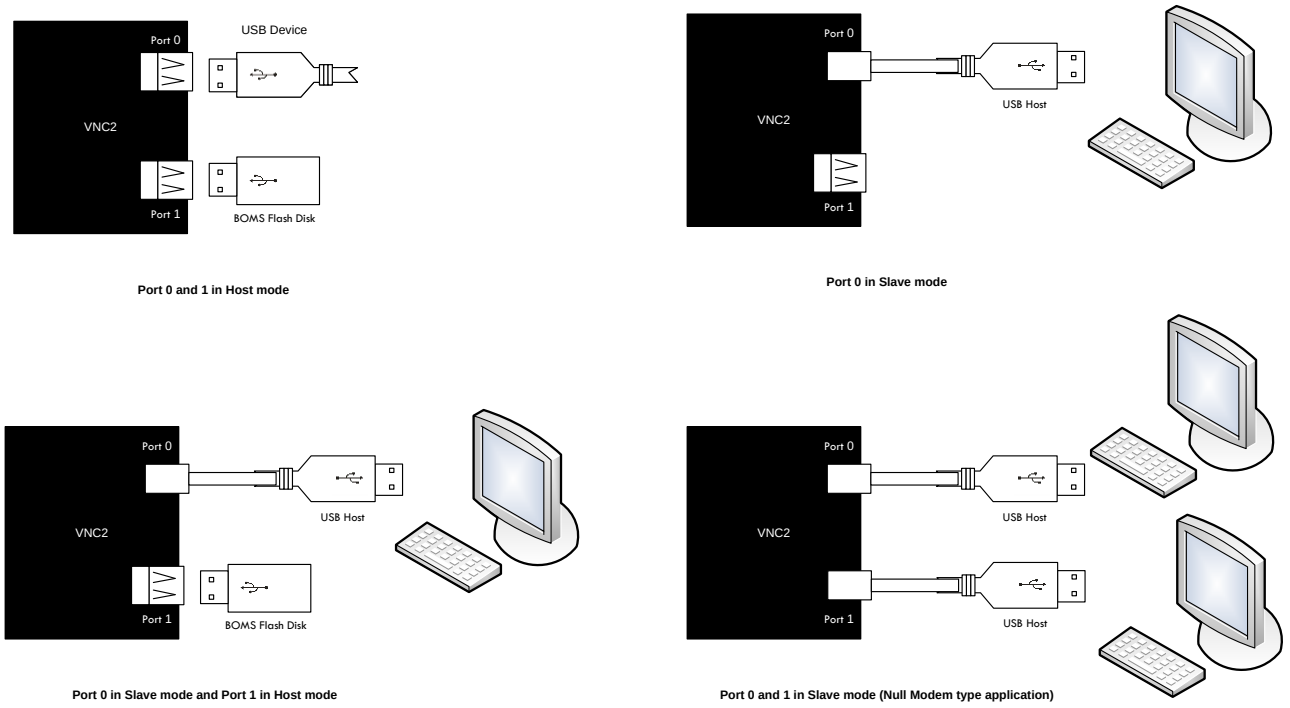

Figure 7.1 USB Modes

8 Firmware

VNC2 firmware model has evolved considerably since VINC1L. For reasons of code maintainability, performance, stability and ease of use from the point of view of the customer, VNC2 has a modular firmware model.

VNC2 firmware can be separated into 4 categories:

- VNC2 real-time operating system (RTOS).
- VNC2 device drivers.
- User applications – Toolchain.
- Precompiled Firmware.

8.1 RTOS

The VNC2 RTOS (VOS) is a pre-emptive priority-based multi-tasking operating system. VOS has been developed by FTDI and is available to customers for use in their own VNC2 based systems free of charge. VOS is supplied as linkable object files.

A full explanation and how to use VOS is available in a separate application note which can be downloaded from the FTDI website.

8.2 Device drivers

To facilitate communication between user applications and the VNC2 hardware peripherals FTDI provides device drivers which operate with VOS. In addition to the hardware device drivers, FTDI provides function drivers (available from the FTDI website) which build upon the basic hardware device driver functionality for a specific purpose. For example, drivers for standard USB device classes may be created which build upon the USB host hardware driver to implement a BOMS class, CDC, printer class or even a specific vendor class device driver.

8.3 Firmware – Software Development Toolchain

The VNC2 provides customers with the opportunity to customise the firmware and perform useful tasks without an external MCU. A Firmware application note is available to download from the FTDI website, this give further details and operating instructions. The VNC2 Software Development Toolchain consists of the following components:

- **Compiler**

  The compiler will take high-level source code and compile it into object code or direct to programmable code.

- **Linker**

  The linker will take object code and libraries and link the code to produce either libraries or programmable code. It is designed to be as hardware independent as possible to allow reuse in future hardware devices.

- **Debugger**

  The debugger allows a programmer to test code on the hardware platform using a special communication channel to the CPU. It is also used to debug code – run, stop, single step, breakpoints etc.

- **IDE**

  All compiler, simulator and debugger functions are integrated into a single application for programmers. It provides a specialised text editor which is used generally used to develop application code, debugging and simulation.

8.4 Precompiled Firmware

VNC2 can be programmed with various pre-compiled firmware profiles to allow a designer to easily change the functionality of the chip.

The following pre-compiled ROM files are currently available:

- **V2DAP firmware:** USB Host for single Flash Disk and general purpose USB peripherals. Selectable UART, FIFO or SPI interface command monitor. Offers a migration path from VNC1L designs with VDAP firmware.
- **V2DPS firmware:** USB Host for single Flash Disk and general purpose USB peripherals and USB peripheral emulating a FT232 on a Host computer. Offers a migration path for VNC1L designs with VDPS.
- **V2F2F firmware:** USB Host for two Flash Disks with file copy functions. Offers a migration path for VNC1L designs with VF2F firmware.
- **CDC Modem Sample Application:** Demonstrates connection of a CDC device to USB Port 1 by establishing a link between the CDC device and the UART of the VNC2.
- **USBHost FT232 UART Echo Sample Application:** Demonstrates emulation of a FTDI FT232 device on USB Port 1. Data is looped back.

Designers are advised to refer to the FTDI website for the most current details on available Firmware.

9 Device Characteristics and Ratings

9.1 Absolute Maximum Ratings

The absolute maximum ratings for VNC2 are shown in **Table 9.1**. These are in accordance with the Absolute Maximum Rating System (IEC 60134). Exceeding these may cause permanent damage to the device.

Parameter	Value	Unit
Storage Temperature	-65 to +150	°C
Floor Life (Out of Bag) At Factory Ambient (30°C / 60% Relative Humidity)	168 (IPC/JEDEC J-STD-033A MSL Level 3 Compliant)*	Hours
Ambient Temperature (Power Applied)	-40 to +85	°C
Vcc Supply Voltage	0 to +3.63	V
VCC_IO	0 to +3.63	V
VCC_PLL_IN	0 to + 1.98	V
DC Input Voltage - USBDP and USBDM	-0.5 to +(Vcc +0.5)	V
DC Input Voltage - XTIN	-0.5 to +((1.8V VCC PLL IN) +0.5)	V
DC Input Voltage - High Impedance Bidirectional	-0.5 to +5.00	V
DC Input Voltage - All other Inputs	-0.5 to +(Vcc +0.5)	V
DC Output Current - Outputs	Default 4 **	mA
DC Output Current - Low Impedance Bidirectional	Default 4 **	mA

Table 9.1 Absolute Maximum Ratings

* If devices are stored out of the packaging beyond this time limit the devices should be baked before use. The devices should be ramped up to a temperature of 125°C and baked for up to 17 hours.

** The drive strength of the output stage may be configured for either 4mA, 8mA, 12mA or 16mA depending on the register setting controlled within the firmware. The default is 4mA.

9.2 DC Characteristics

DC Characteristics (Ambient Temperature -40°C to +125°C)

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Vcc1	VCC Operating Supply Voltage	1.62	1.8	1.98	V	
Vcc2	VCCIO Operating Supply Voltage	2.97	3.3	3.63	V	
VCC_PLL	VCC_PLL Operating Supply Voltage	1.62	1.8	1.98	V	
Icc1	Operating Supply Current 48MHz		25		mA	Normal Operation
Icc2	Operating Supply Current 24MHz		16		mA	Low Power Mode
Icc3	Operating Supply Current 12MHz		8		mA	Lowest Power Mode
Icc4	Operating Supply Current		128		µA	USB Suspend

Table 9.2 Operating Voltage and Current

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	2.4			V	I source = 8mA
Vol	Output Voltage Low			0.4	V	I sink = 8mA
Vin	Input Switching Threshold		1.5		V	

Table 9.3 I/O Pin Characteristics

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
UVoh	I/O Pins Static Output (High)	2.8			V	
UVol	I/O Pins Static Output (Low)			0.3	V	
UVse	Single Ended Rx Threshold	0.8		2.0	V	
UCom	Differential Common Mode	0.8		2.5	V	
UVdif	Differential Input Sensitivity	0.2			V	
UDrvZ	Driver Output Impedance	3	6	9	Ohms	

Table 9.4 USB I/O Pin (USBDP, USBDM) Characteristics

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
VCCK	Power supply of internal core cells and I/O to core interface	1.62	1.8	1.98	V	1.8V power supply
VCC18IO	Power supply of 1.8V OSC pad	1.62	1.8	1.98	V	1.8V power supply
T_J	Operating junction temperature	-40	25	125	°C	
I_{in}	Input leackage current	-10	±1	10	µA	I_{in} = VCC18IO or 0V
I_{oz}	Tri-state output leakage current	-10	±1	10	µA	

Table 9.5 Crystal Oscillator 1.8 Volts DC Characteristics

9.3 ESD and Latch-up Specifications

Description	Specification
Human Body Mode (HBM)	TBD
Machine mode (MM)	TBD
Charged Device Mode (CDM)	TBD
Latch-up	> ± 200mA

Table 9.6 ESD and Latch-up Specifications

10 Application Examples

10.1 Example VNC2 Schematic (MCU – UART Interface)

VNC2 can be configured to communicate with a microcontroller using a UART interface. An example of this is shown in **Figure 10.1**.

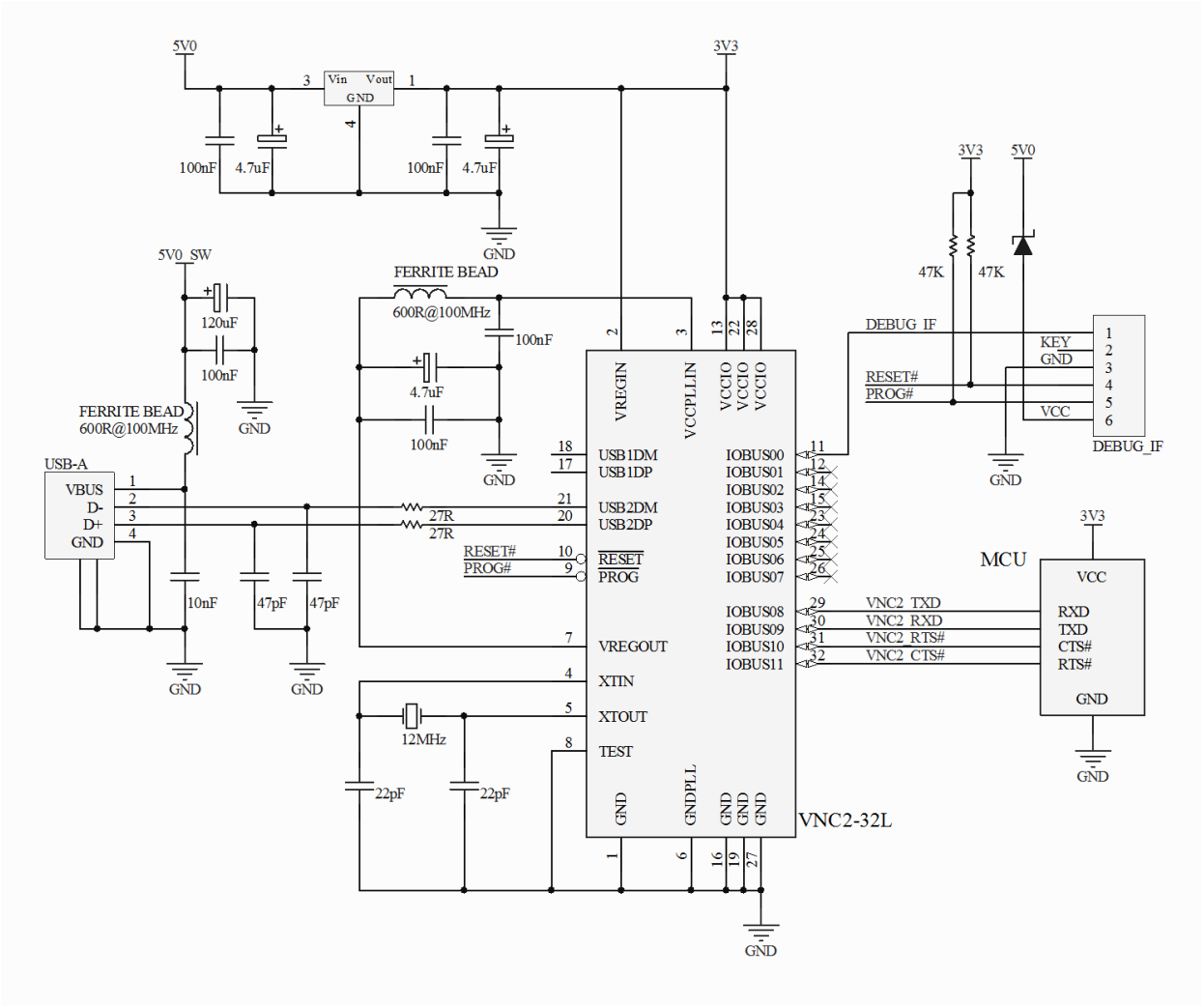

Figure 10.1 VNC2 Schematic (MCU – UART Interface)

NOTES: This sample circuit is not intended to be a complete design. It shows the minimum connections for a basic VNC2 circuit. The value of the capacitors connected to the crystal will depend on the requirements of the crystal. The 5V0_SW power signal assumes proper switching and over-current detection conform to the USB-IF specifications for a USB host port. The 120uF capacitor should be a low-ESR type. The value of the ferrite beads may need adjusted for EMI compatibility. Input and output capacitors for the 3.3V regulator should be chosen according to the datasheet of the selected part. The VNC2 outputs connect to the MCU inputs and MCU outputs to VNC2 inputs (TXD to RXD and RTS# to CTS# in each direction).

NOTE for VNC2-48L1B ONLY: With the 48-pin LQFP package, pin 7 is not connected (VREGOUT). The regulator output has an internal connection to VCCPLLIN to accommodate a migration path from VNC1L designs. In this case, pin 3 (VCCPLLIN) requires only one 100nF capacitor to ground. All other packages require the external circuitry shown to connect VREGOUT to VCCPLLIN.

11 Package Parameters

VNC2 is available in six RoHS Compliant packages, three QFN packages (64QFN, 48QFN & 32QFN) and three LQFP packages (64LQFP, 48LQFP & 32LQFP). All packages are lead (Pb) free and use a 'green' compound. The packages are fully compliant with European Union directive 2002/95/EC.

The mechanical drawings of all six packages are shown in sections **11.2** to **11.7**– all dimensions are in millimetres.

The solder reflow profile for all packages can be viewed in Section **11.8**.

11.1 VNC2 Package Markings

An example of the markings on each package is shown in **Figure 11.1**. The FTDI part number is too long for the 32 QFN package so in this case the last two digits are wrapped down onto the date code line as shown in **Figure 11.2**.

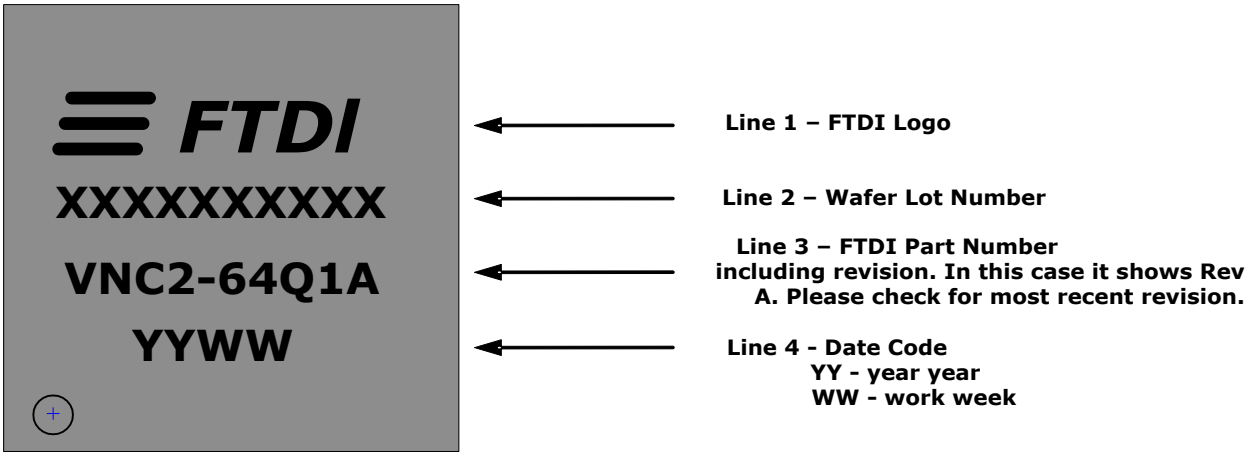

Figure 11.1 Package Markings

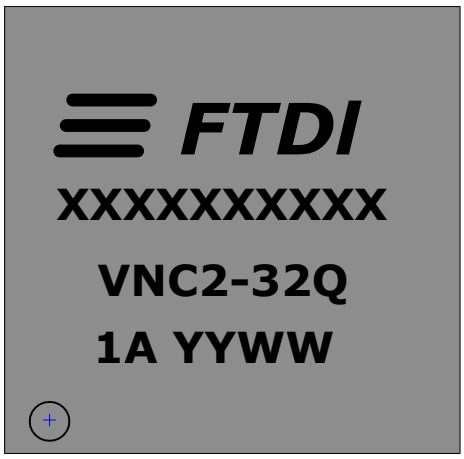

Figure 11.2 Markings – 32 QFN

The last letter of the FTDI part number is the silicon revision number. This may change from A to B to C, etc.,. Please check the part number for the most recent revision.

11.2 VNC2, LQFP-32 Package Dimensions

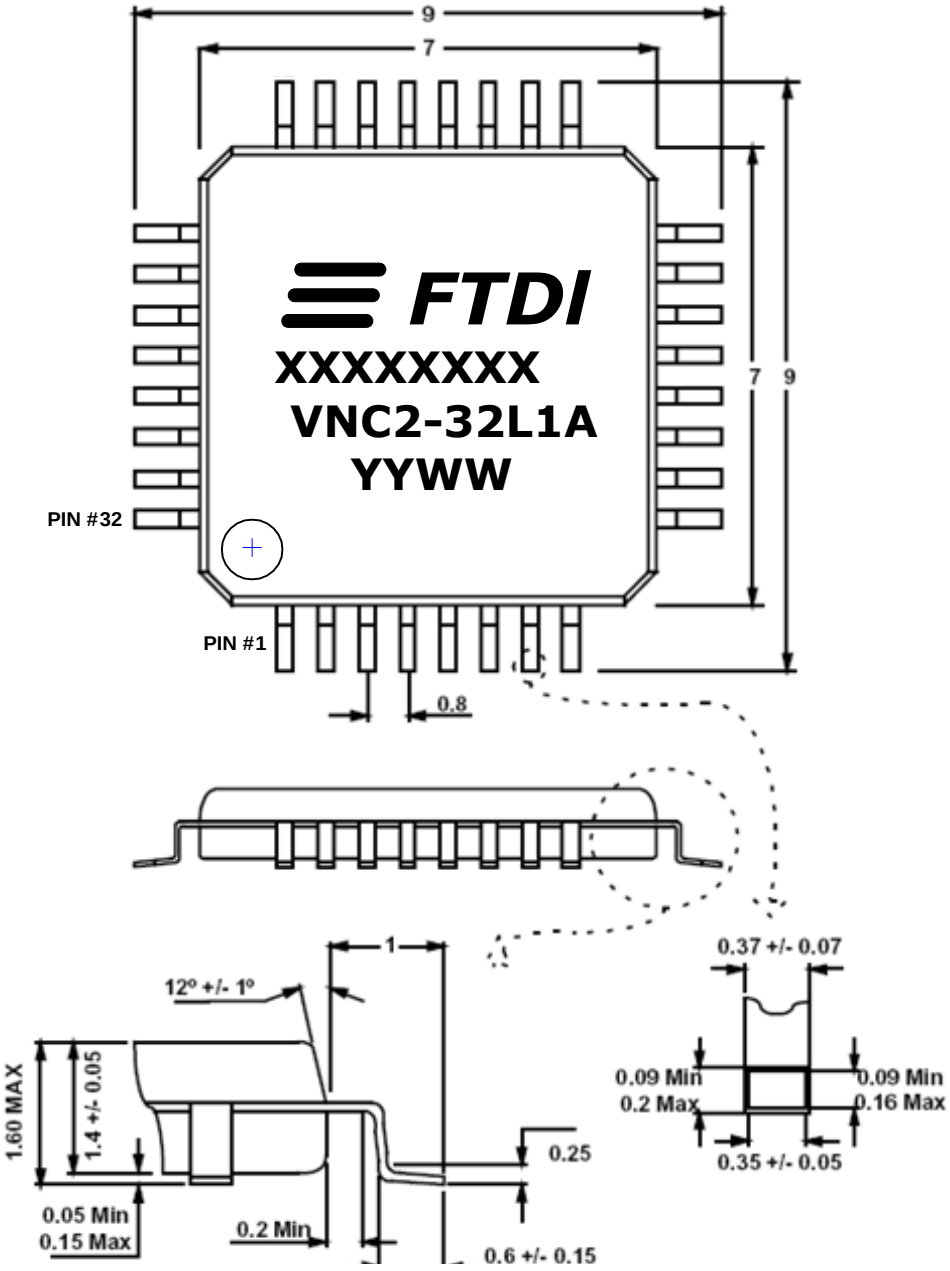

Figure 11.3 LQFP-32 Package Dimensions

11.3 VNC2, QFN-32 Package Dimensions

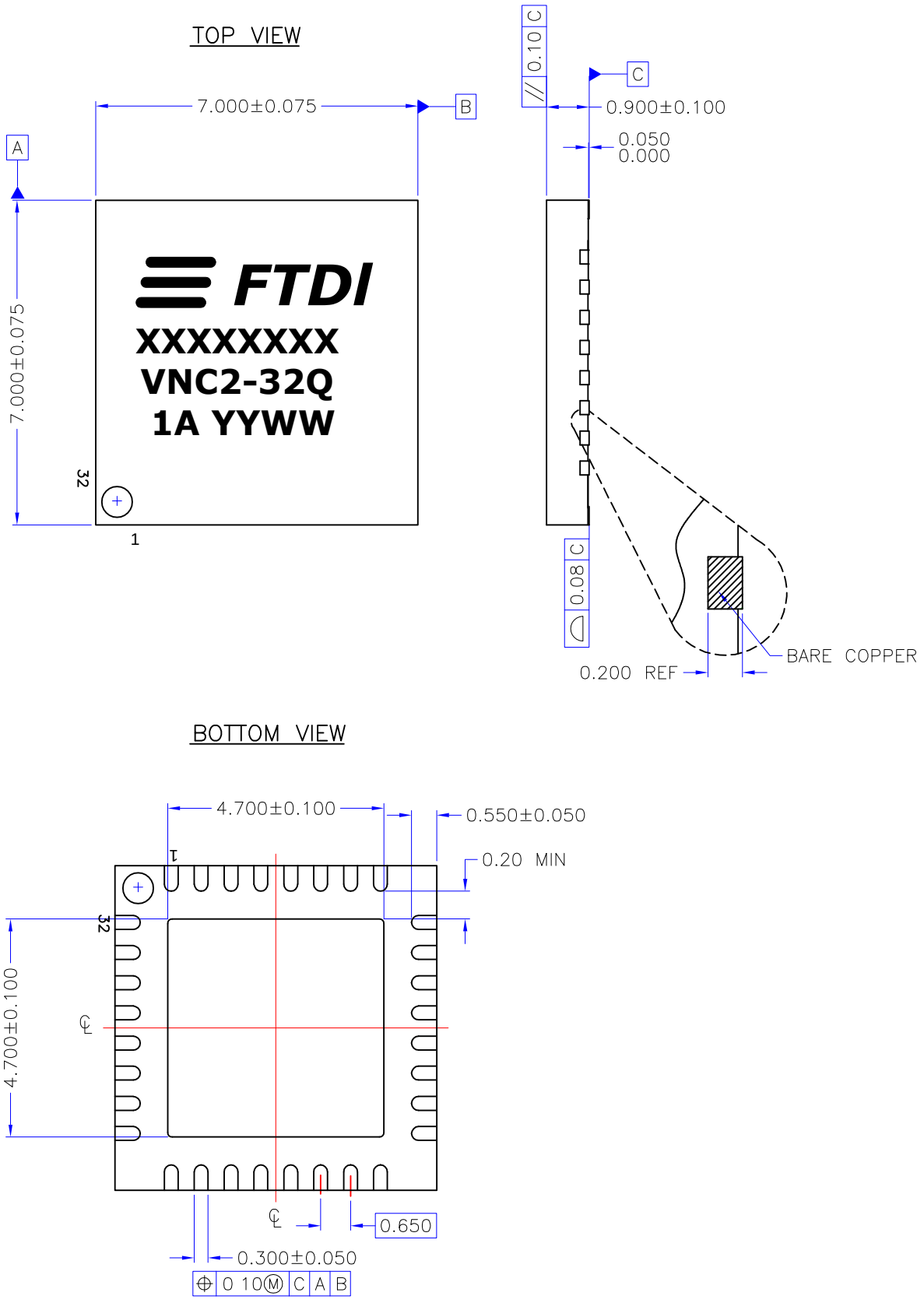

Figure 11.4 QFN-32 Package Dimensions

11.4 VNC2, LQFP-48 Package Dimensions

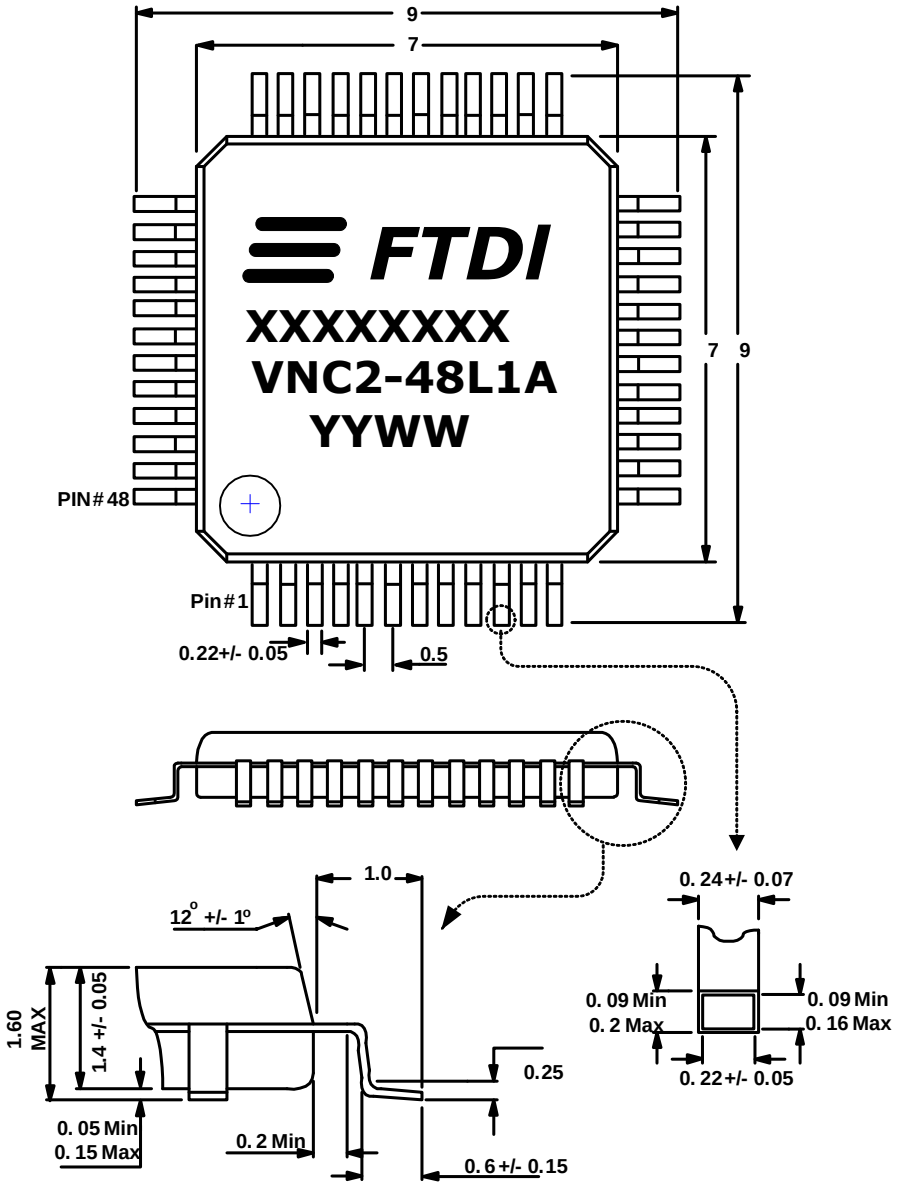

Figure 11.5 LQFP-48 Package Dimensions

11.5 VNC2, QFN-48 Package Dimensions

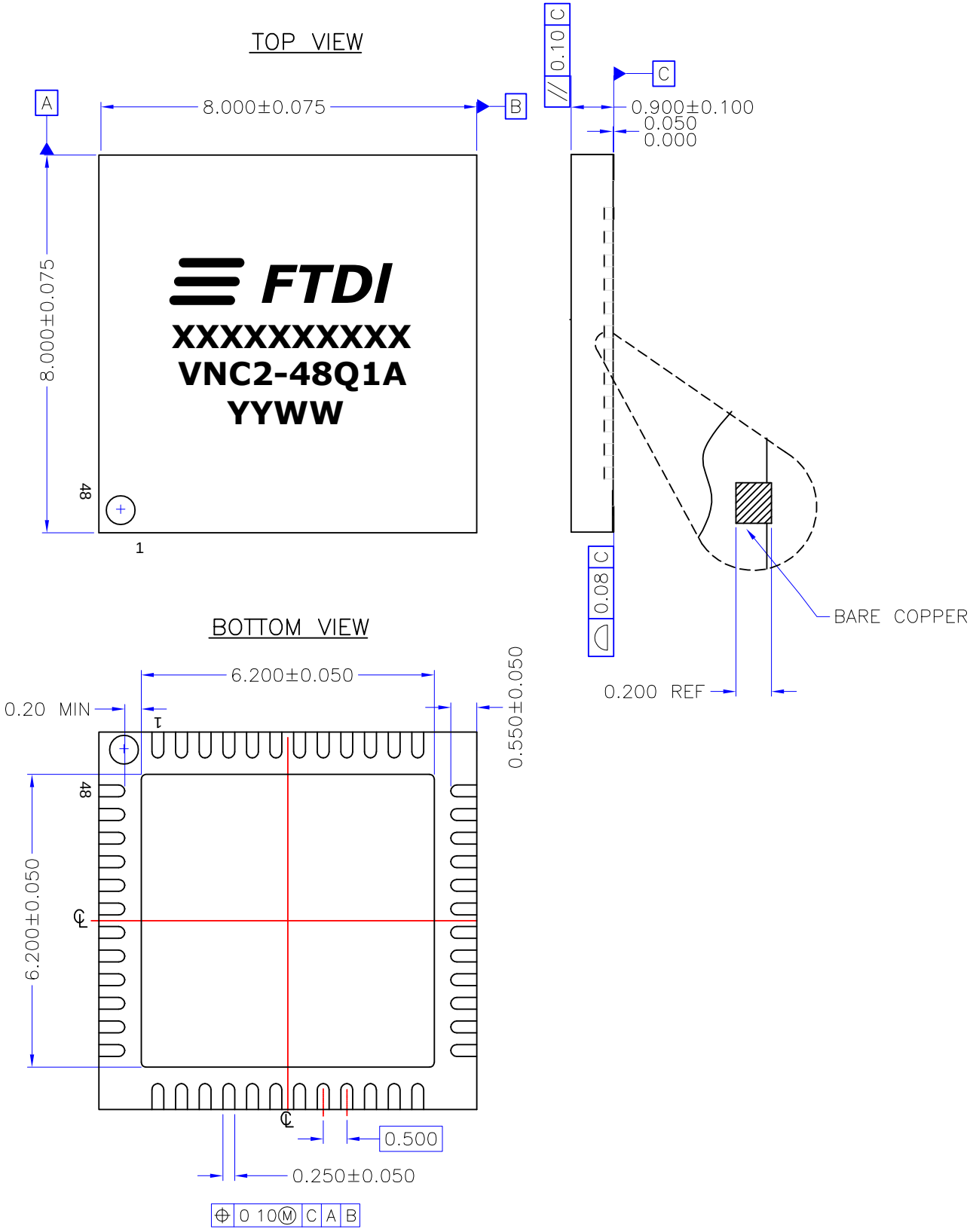

Figure 11.2 QFN-48 Package Dimensions

11.6 VNC2, LQFP-64 Package Dimensions

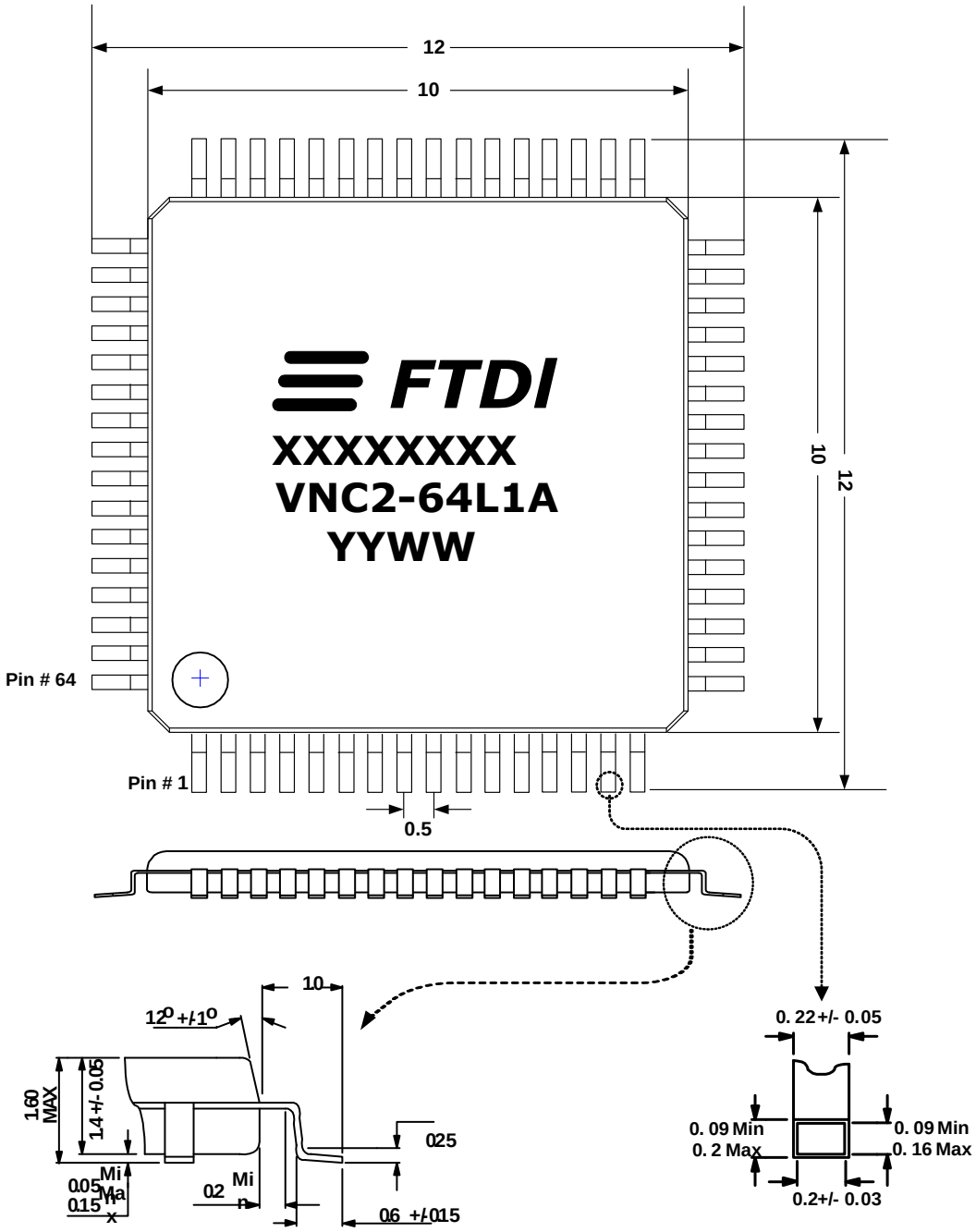

Figure 11.6 64 pin LQFP Package Details

11.7 VNC2, QFN-64 Package Dimensions

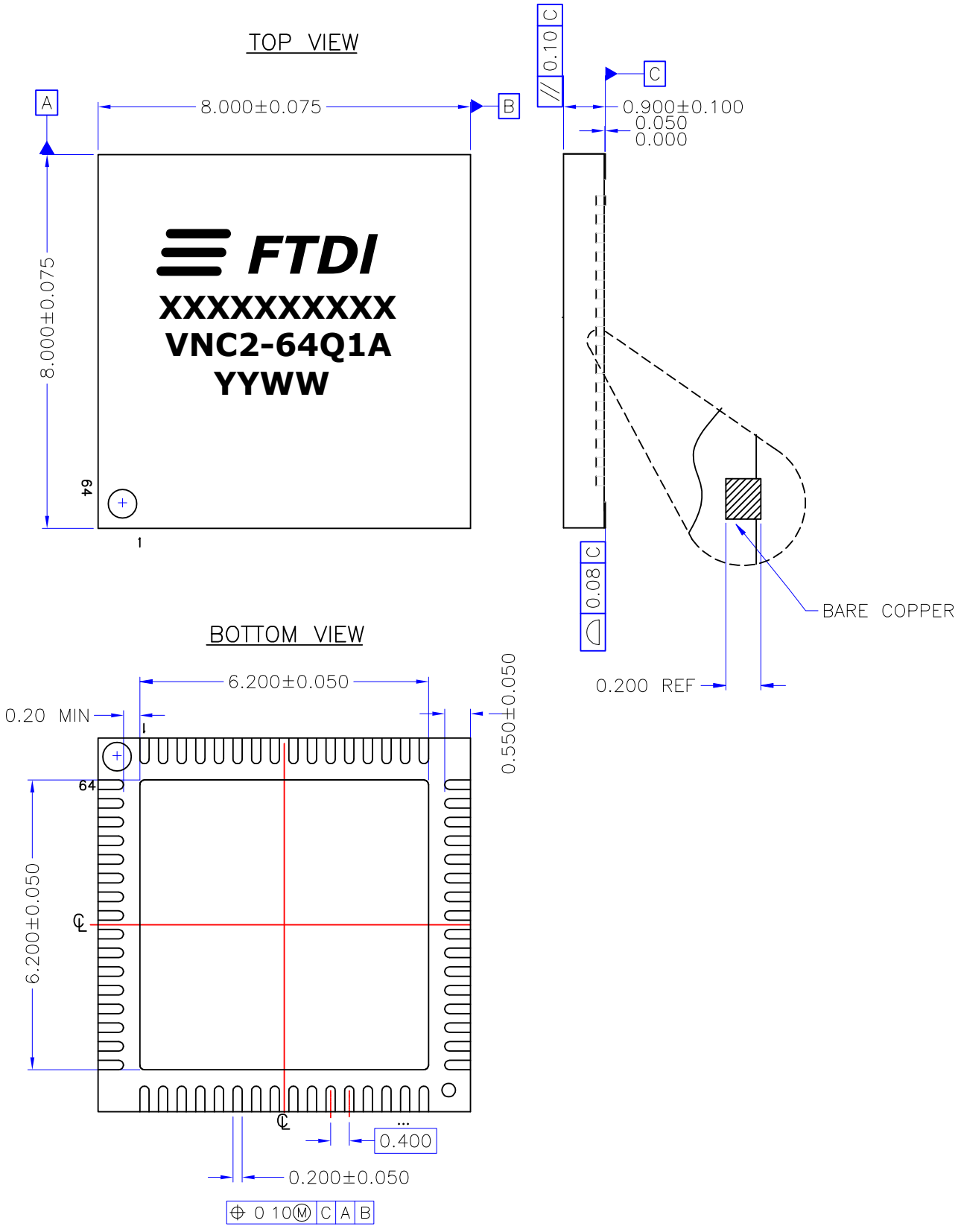

Figure 11.7 64 pin QFN Package Details

11.8 Solder Reflow Profile

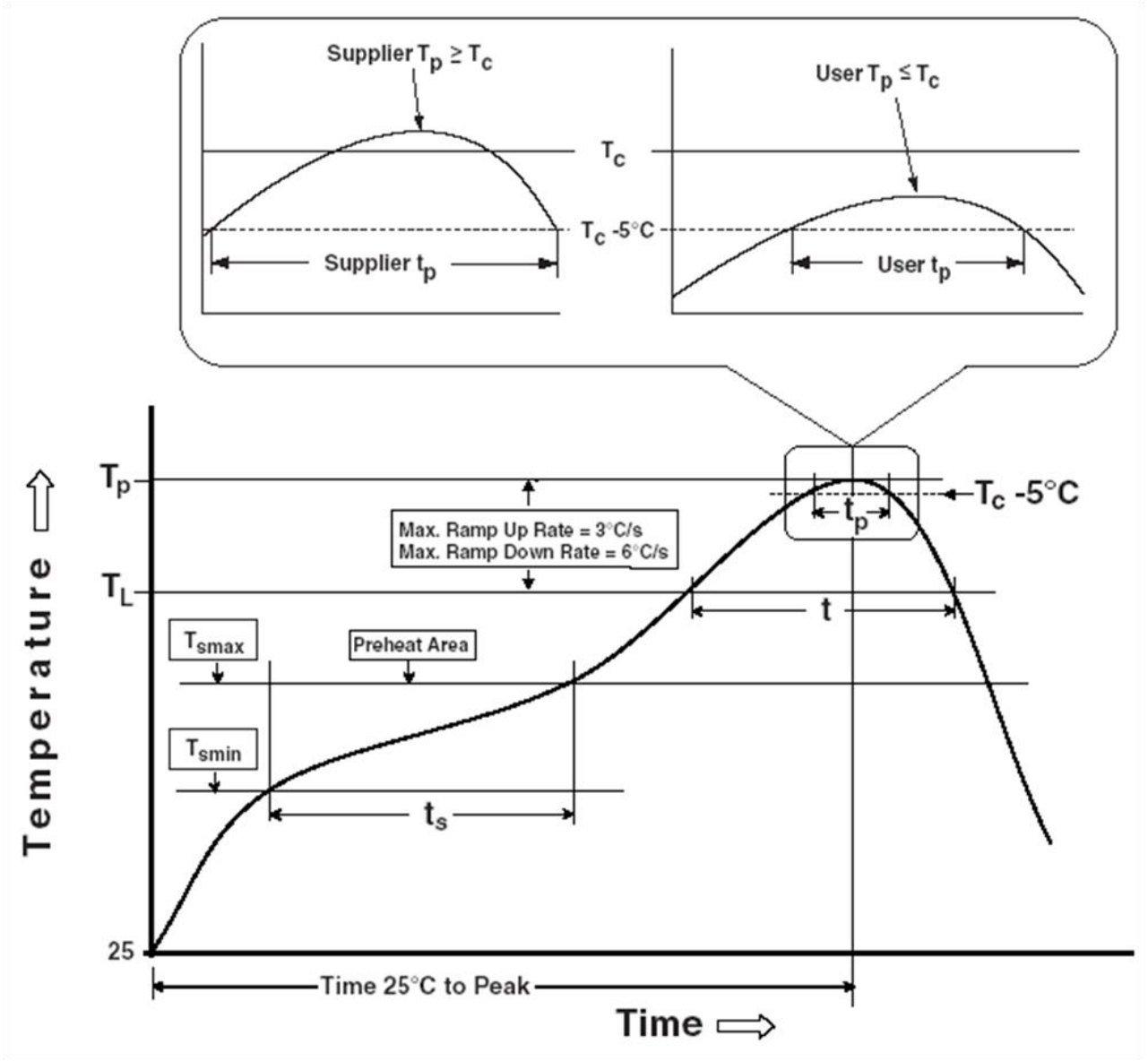

Figure 11.8 All packages Reflow Solder Profile

Profile Feature	Pb Free Solder Process (green material)	SnPb Eutectic and Pb free (non green material) Solder Process
Average Ramp Up Rate (T_s to T_p)	3°C / second Max.	3°C / Second Max.
Preheat: - Temperature Min (T_s Min.) - Temperature Max (T_s Max.) - Time (t_s Min to t_s Max)	 150°C 200°C 60 to 120 seconds	 100°C 150°C 60 to 120 seconds
Time Maintained Above Critical Temperature T_L: - Temperature (T_L) - Time (t_L)	 217°C 60 to 150 seconds	 183°C 60 to 150 seconds
Peak Temperature (T_p)	260°C	see **Table 11.2**
Time within 5°C of actual Peak Temperature (t_p)	30 to 40 seconds	20 to 40 seconds
Ramp Down Rate	6°C / second Max.	6°C / second Max.
Time for T= 25°C to Peak Temperature, T_p	8 minutes Max.	6 minutes Max.

Table 11.1 Reflow Profile Parameter Values

SnPb Eutectic and Pb free (non green material)		
Package Thickness	Volume mm3 < 350	Volume mm3 >=350
< 2.5 mm	235 +5/-0 °C	220 +5/-0 °C
≥ 2.5 mm	220 +5/-0 °C	220 +5/-0 °C
Pb Free (green material) = 260 +5/-0 °C		

Table 11.2 Package Reflow Peak Temperature

12 Contact Information

Head Office – Glasgow, UK

Future Technology Devices International Limited
Unit 1, 2 Seaward Place, Centurion Business Park
Glasgow G41 1HH
United Kingdom
Tel: +44 (0) 141 429 2777
Fax: +44 (0) 141 429 2758

E-mail (Sales) sales1@ftdichip.com
E-mail (Support) support1@ftdichip.com
E-mail (General Enquiries) admin1@ftdichip.com

Branch Office – Tigard, Oregon, USA

Future Technology Devices International Limited (USA)
7130 SW Fir Loop
Tigard, OR 97223-8160
USA
Tel: +1 (503) 547 0988
Fax: +1 (503) 547 0987

E-Mail (Sales) us.sales@ftdichip.com
E-Mail (Support) us.support@ftdichip.com
E-Mail (General Enquiries) us.admin@ftdichip.com

Branch Office – Taipei, Taiwan

Future Technology Devices International Limited (Taiwan)
2F, No. 516, Sec. 1, NeiHu Road
Taipei 114
Taiwan , R.O.C.
Tel: +886 (0) 2 8797 1330
Fax: +886 (0) 2 8751 9737

E-mail (Sales) tw.sales1@ftdichip.com
E-mail (Support) tw.support1@ftdichip.com
E-mail (General Enquiries) tw.admin1@ftdichip.com

Branch Office – Shanghai, China

Future Technology Devices International Limited (China)
Room 1103, No. 666 West Huaihai Road,
Shanghai, 200052
China
Tel: +86 21 62351596
Fax: +86 21 62351595

E-mail (Sales) cn.sales@ftdichip.com
E-mail (Support) cn.support@ftdichip.com
E-mail (General Enquiries) cn.admin@ftdichip.com

Web Site

http://ftdichip.com

Distributor and Sales Representatives

Please visit the Sales Network page of the FTDI Web site for the contact details of our distributor(s) and sales representative(s) in your country.

Appendix A – References

Application, Technical Notes,Toolchain download and precompiled romfile links

The following VNC2 documents and the full Vinculum-II Toolchain software suite can be downloaded by clicking on the appropriate links below:

Technical note TN_108	Vinculum Chipset Feature Comparison
Technical note TN_118	Vinculum-II Errata Technical Note
Application note AN_118	Migrating Vinculum Designs From VNC1L to VNC2-48L1A
Application note AN_137	Vinculum-II IO Cell Description
Application note AN_138	Vinculum-II Debug Interface Description
Application note AN_139	Vinculum-II IO Mux Explained
Application note AN_140	Vinculum-II PWM Example
Application note AN_142	Vinculum-II Toolchain Getting Started Guide
Application note AN_144	Vinculum-II IO Mux Configuration Utility User Guide
Application note AN_145	Vinculum-II Toolchain Installation Guide
Application note AN_151	Vinculum-II User Guide
VNC2 FTDI Web Page	Vinculum-II Web Page

The following application notes provide pre-compiled example rom files and complete source code to allow users to get started:

- Application note AN_182 : VNC2 UART to FT232 HostBridge
- Application note AN_183 : VNC2 UART to CDC Modem Bridge
- Application note AN_185 : VNC2 UART to USB HID Class HostBridge
- Application note AN_186 : an SPI Slave to USB Memory bridge
- Application note AN_187 : VNC2 UART to USB Memory Bridge
- Application note AN_192 : an SPI Master to a USB HID class host bridge
- Application note AN_193 : an SPI Master to a USB HID class host bridge
- Application note AN_194 : VNC2 UART to HID Class device bridge
- Application note AN_195 : an SPI Master to UART bridge
- Application note AN_199 : VNC2 SPI Slave to HID Class device bridge

Application note AN_203 : Loading VNC2 ROM files Using V2PROG Utility

For the most up to date pre-compiled rom files, please refer to the following FTDI webpage
http://www.ftdichip.com/Firmware/Precompiled.htm

Acronyms and Abbreviations

Terms	Description
USB	Universal Serial Bus
FIFO	First In First Out
SPI	Serial Peripheral Interface
PWM	Pulse Width Modulation
GPIO	General Purpose Input Output
I/O	Input / Output
VNC1L	Vinculum-I
VNC2	Vinculum-II
DMA	Direct Memory Access
IDE	Integrated Development Environment
BOMS	Bulk Only Mass Storage
UART	Universal Asynchronous Receiver/Transmitter
SIE	Serial Interface Engine
CPU	Central Processing Unit
SoC	System-on-a-chip
FAT	File Allocation Table
RTOS	Real Time Operating System
VOS	Vinculum Operating System
OSI	Open System Interconnection
MOSI	Master Out Slave In
MISO	Master In Slave Out
SE0	Single Ended Zero
EMCU	Embedded Micro Central Processing Unit
FPGA	Field Programmable Gate Array

Appendix B – List of Figures and Tables

List of Tables

List of Figures

Appendix C – Revision History

Document Title: Vinculum-II Embedded Dual USB Host Controller IC Datasheet
Document Reference No.: FT_000138
Clearance No.: FTDI# 143
Document Folder: Vinculum-II
Document Feedback: Send Feedback

Revision	Changes	Date
Prelim	Data sheet released as "Preliminary – Subject to change" before product launch	2010-02-01
1.0	Initial Release	2010-02-26
1.1	Changed gpio signal names, fixed minor typographical errors, added crystal characteristic information, added ESD table	2010-09-09
1.2	Revised part numbers to "Rev B" in section 1.2, added notes to sections 3.12 and 5 – default pin assignments	2010-10-07
1.3	Added USB transfer/transaction combinations	2011-04-19
1.31	Table 37 (now 9.6) modified	2011-05-16
1.32	Table 33 (now 9.2) modified	2011-05-17
1.4	Renumbered and reformatted figures and tables, Added life/safety notice, Added comments regarding 48LQFP package in table 3.2, Replaced Figure 5.5 with new screen shot, Updated list of pre-compiled firmware in section 8.4, Replaced schematic and added notes in Section 10, Updated header, Updated revision history and contact information pages	2011-10-17
1.5	Added notes (5V safe inputs) in section 3.11, 3.12 and first page. Added links to pre-compiled rom files.	2012-03-14
1.6	XTIN is referenced to 1.8V VCC PLL IN - Tables 3.3 and 9.1 Pin 8 updated from TEST to NC for rev C	2016-07-07
1.7	Removed revision code in table1.1	2016-07-21