VPU16551 Versatile PC-Card UART

Product Datasheet ES313

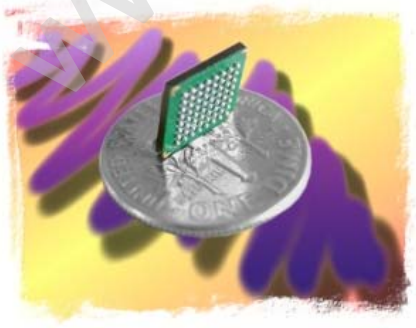
1.0 Key Features

- ***Built-in*** **512 byte reprogrammable E² for CIS *means true single chip* PCMCIA UART solution & *"live at power-up"***
- **5v or 3.3v operation**
- **Tiny *64-CBGA* package occupies 36% less PCB area than 48QFP[1] and 77% less than 100QFP, and is only 1.1mm thick (nom.)**

- **PC-Card 7.1 multi-function compliant device**
- **Up to 1MBaud operation using x8 baud rate setting in hardware…no special drivers needed**
- **16 Byte Deep TX and RX FIFOs**
- **Supports optional external FIFOs for ultra high rate streaming[2]**
- **For PCMCIA or Compact Flash cards**
- **16550 compatible UART function**
- **Uses standard drivers from Windows operating systems including CE / PocketPC[2]**
- **Ultra-low 3.3v quiescent power (max. 150uA)**
- **Output for UART activity LED**
- **Parallel port mode**
- **Integrated pull up / down resistors**

The VPU16551 is a PC-Card UART device that is designed to operate without any extra glue-logic. It requires just an external crystal, a few passive components and a suitable transceiver to form a complete serial port device compatible with industry standard "COM" ports. It can also operate at up to 1MBaud.

Optionally, the addition of external FIFOs allows the UART to add a high-speed parallel data streaming port, without affecting the UARTs normal operation. This is perfect for applications that require a serial "control" channel and a high rate data channel for throughputs in excess of 4Mbit/sec[2].



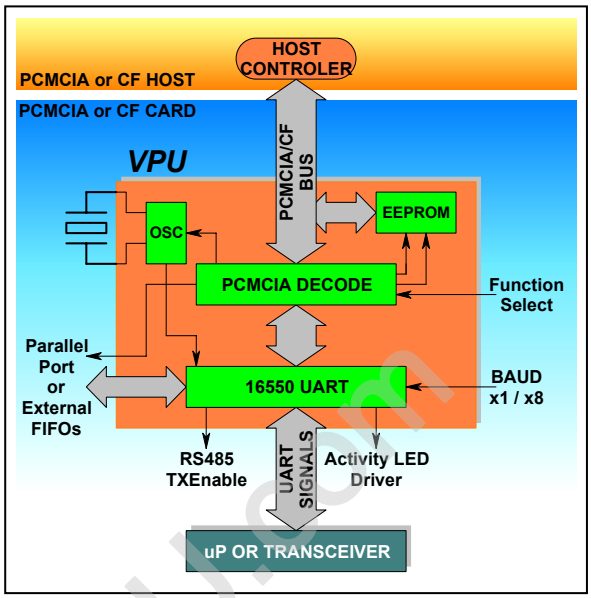


Figure 1-1 VPU16551 Block Diagram

The VPU16551 is ideally suited to the creation of PC-Cards or CF cards that require standard "COM" port type functionality such as wireless devices or instrumentation products.

The following applications are ideal for the VPU16551:

- **Bluetooth Wireless card**
- **Classic modem card**
- **Serial port card**
- **RF data link card**
- **GSM / GPRS interface card**
- **GPS card**
- **Proprietary interface card**

A clear advantage to using the VPU16551 is the Operating System level compatibility that the 16551 architecture brings. Together with its extra features, the VPU16551 greatly simplifies PC-Card / CF Card design. Unlike competing solutions, the VPU16551 does not need extra drivers (for standard operation) and so will "just work" under all Windows operating systems[2].

A reference kit is available for the VPU16551, please contact sales for further information.

Ordering Information

Part Number	Description
VPU16551-CBGAX64	64 ball CBGA, 0-70°C

[1] 64-CBGA Package is 8mm x 8mm. 48QFP requires approximately a 10mm x 10mm PCB area, 100QFP requires 16.5mm x16.5mm.

[2] EXTERNAL FIFO mode and some enhanced features must use Elan's proprietary driver

Contents

Disclaimer

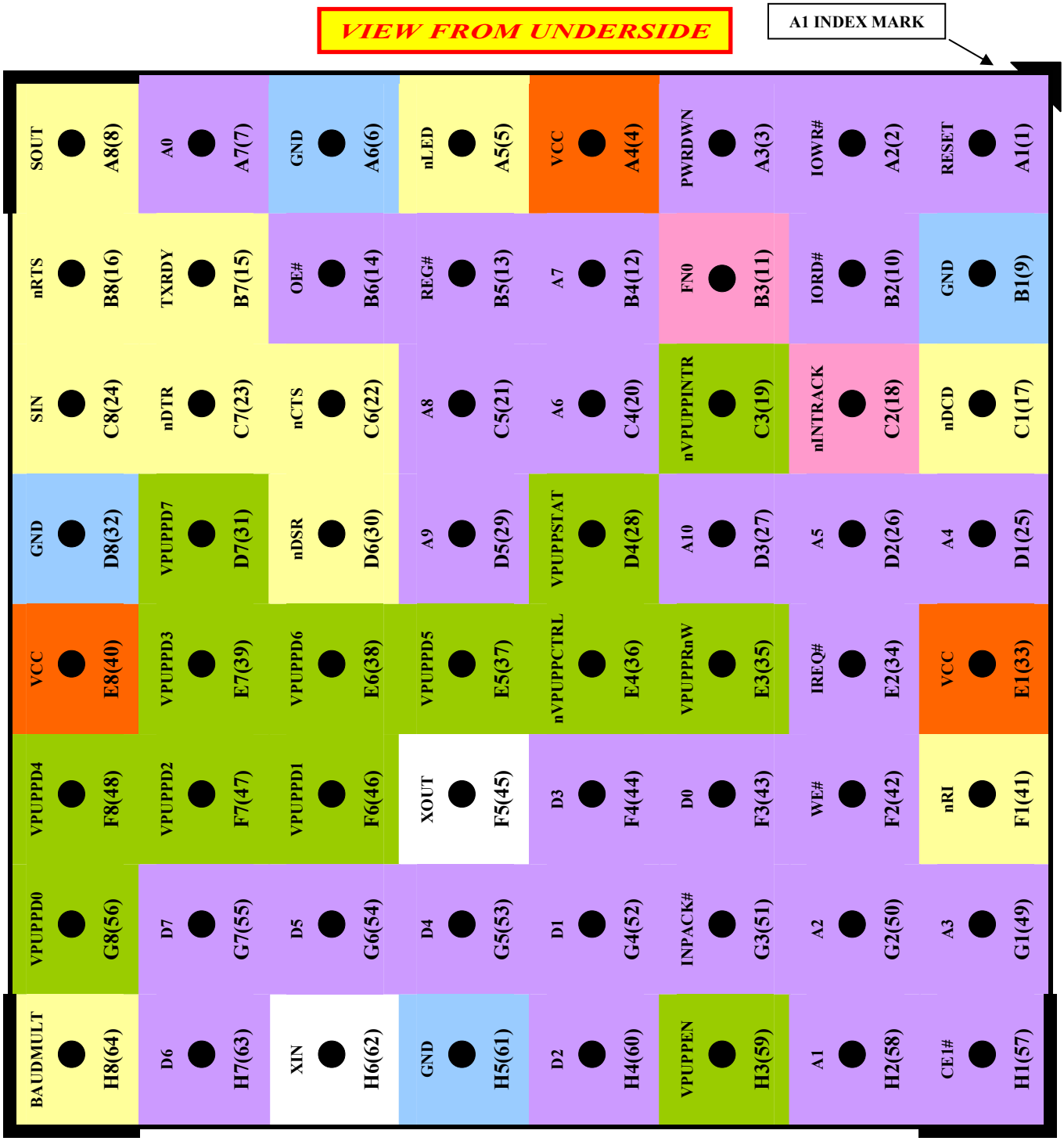
This document has been carefully prepared and checked. No responsibility can be assumed for inaccuracies. Elan reserves the right to make changes without prior notice to any products herein to improve functionality, reliability or other design aspects. Elan does not assume any liability for loses arising out of the use of any product described herein; neither does it convey any licence under its patent rights or the rights of others. Elan does not guarantee the compatibility or fitness for purpose of any product listed herein. Elan products are not authorised for use as components in life support services or systems. Elan should be informed of any such intended use to determine suitability of the products.

Source code supplied with Elan products is provided "as-is" with no warranty, express or implied, as to its quality or fitness for a particular purpose. Elan assumes no liability for any direct or indirect losses arising from use of the supplied code.

DATA SHEET REVISION HISTORY

Revision	Changes
01	First formal issue for VPU16551. REVR reads Bh.
02	REVR reads Ch.
03	Correct typo on P21 regarding BAUDMULT pin
04	Add ESD Rating in Electrical specs

2.0 Pinout



"XXX#"=active low PCMCIA signal, "nXXX"=active low VPU signal
Numbers in (brackets) are for PCB tools that cannot handle alpha-numeric pin numbers.

●=PCMCIA/CF I/F	●=MULTI FUNC	●=UART	●=VPUPP	●=OSC

3.0 Signal Types

SIGNAL NAME(S)	BALL NUMBER(S)	TYPE	FUNCTION
D7:0	G7, H7, G6, G5, F4, H4, G4, F3	I/O with integrated 100K pull downs	PCMCIA bi-directional data bus. D0 is LSB.
A10:0	D3, D5, C5, B4, C4, D2, D1, G1, G2, H2, A7	I	PCMCIA address bus.
CE1#	H1	I	PCMCIA chip select.
OE#	B6	I	PCMCIA memory read strobe.
IORD#	B2	I	PCMCIA I/O read strobe.
IOWR#	A2	I	PCMCIA I/O write strobe.
WE#	F2	I	PCMCIA memory write strobe.
IREQ#	E2	O/D with integrated 100K pull up	PCMCIA interrupt request.
RESET	A1	SI with integrated 100K pull up	PCMCIA reset.
INPACK#	G3	O/D with integrated 100K pull up	PCMCIA input port acknowledge.
REG#	B5	I	PCMCIA attribute or I/O select.
PWRDWN	A3	O	Power down control output.
FN0	B3	I with integrated 100K pull down	Function select for multi-VPU.
nINTRACK	C2	O/D, SI with integrated 100K pull up	Interrupt acknowledge for multi-VPU.
VPUPPEN	H3	I with integrated 100K pull down	Enable VPU parallel port or EXTERNAL FIFO mode.
VPUPPD7:0 / EXTFIFOD7:0	D7, E6, E5, F8, E7, F7, F6, G8	I/O	VPU parallel port or EXTERNAL FIFO data. VPUPPD0 is LSB.
VPUPPRnW / nEXTFIFOEN	E3	I with integrated 100K pull up	VPU parallel port direction control or EXTERNAL FIFO mode enable.
nVPUPPINTR / nR	C3	O	VPU parallel port interrupt or EXTERNAL FIFO read strobe.
VPUPPSTAT / nW	D4	O	VPU parallel port TX/RX data status or EXTERNAL FIFO write strobe.
nVPUPPCTRL	E4	I with integrated 100K pull up	VPU parallel port TX/RX data strobe.
TXRDY	B7	O	Flag indicating TX data in being sent.
nDTR	C7	O, O/D or HiZ	UART modem control signal.
SOUT	A8	O, O/D or HiZ	UART serial data out.
nRTS	B8	O, O/D or HiZ	UART modem control signal.
SIN	C8	SI	UART serial data in.
nCTS	C6	SI	UART modem control signal.
nDCD	C1	SI	UART modem control signal.
nDSR	D6	SI	UART modem control signal.
nRI	F1	SI	UART modem control signal.

XIN	H6	I with integrated 100K parallel bias resistor to Xout	Crystal / Oscillator in.
XOUT	F5	O	Crystal / Oscillator out.
BAUDMULT	H8	I	Baud rate multiplier select.
nLED	A5	O/D	UART Activity LED drive.
GND	A6, B1, D8, H5	PWR	0V.
VCC	A4, E1, E8	PWR	3.3V or 5V power.

I=input, SI=schmitt input, O=output, O/D=open drain, PWR=power pin,
"XXX#"=active low PCMCIA signal, "nXXX"=active low VPU signal
Ball number lists for buses (BusN:0) are in order with MSBit first.

4.0 Signal Descriptions

D7:0

The 8-bit bi-directional data path to/from the PCMCIA host. These signals default to inputs at the VPU when no bus transaction is in progress. All communication between the VPU UART configuration and data registers and the host occur on these data lines, and always as 8-bit wide transactions.

The D7:0 lines have built-in 100K pull down resistors.

A10:A0

The address used during transactions with the VPU. A10 is decoded by the VPU to enable the internal CIS E². The CIS is mapped from address 0h to 3FFH in attribute space. PCMCIA Card Configuration Registers are mapped between 400H and 7FFH.

☝ If you need to lower the boundary between the CIS E² and the Card Configuration Registers, the A10 pin can be wired to one of the lower address lines e.g. wire pin A9 to signal A9, pin A10 to signal A9 to relocate the CCRs to address 200H and shorten the CIS E² region to 0h to 1FFH. (Normally this is not required but if for some reason it is important to map the CCRs to an address other than 400H, then this technique can be used).

CE1#

Chip select input to the VPU. This signal qualifies all transactions to/from the VPU from the PCMCIA host.

OE#

Memory Read Strobe input to the VPU. This signal qualifies memory read transactions from the VPU by the PCMCIA host.

WE#

Memory Write Strobe input to the VPU. This signal qualifies memory write transactions to the VPU by the PCMCIA host.

IORD#

I/O Read Strobe input to the VPU. This signal qualifies I/O read transactions from the VPU by the PCMCIA host.

IOWR#

I/O Write Strobe input to the VPU. This signal qualifies I/O write transactions to the VPU by the PCMCIA host.

IREQ#

Interrupt request to the host. This signal flags to the host that the VPU requires service. For multi-VPU mode, connect all IREQ# signals together.

The IREQ# line has a built-in 100K pull-up resistor. Note that for faster IREQ# rise time, you may chose to provision a 10K (or less) pull up resistor on the PCB layout.

nINTRACK

This signal allows a multi-VPU configuration to share the single IREQ# PCMCIA line. The nINTRACK signal will assert low whenever the VPU detects that a 0 has been written to its CSR Bit1 while its CSR Bit0 is a 1. All VPUs will drive/monitor this shared signal and will deassert their IREQ# output when it is low. For multi-VPU mode, wire all nINTRACK signals together. For single VPU mode this signal can be left as no connect. The input stage for the nINTRACK signal is a Schmitt input type to cope with the relatively slow R-C controlled rise time as the signal de-asserts.

The nINTRACK line has a built-in 100K pull-up resistor.

RESET

RESET input to the VPU. This signal clears all internal registers when asserted. The RESET signal is a Schmitt input type (the host tri-states the RESET signal during power-up to allow the pull up resistor (built-in) to affect a RESET assertion. The rise times can therefore be slow. The Schmitt input hysteresis deals with this).

The RESET line has a built-in 100K pull-up resistor.

INPACK#

Input port acknowledge from the VPU. This signal indicates a VPU internal I/O read "hit" and is used in some host systems to enable down-stream data buffers during I/O reads from the VPU. In multi-VPU mode, connect all INPACK# signals together.

The INPACK# line has a built-in 100K pull-up resistor. Note that for faster INPACK# rise time, you may chose to provision a 10K (or less) pull up resistor on the PCB layout.

REG#

Register select input to the VPU. This signal qualifies I/O or attribute memory accesses to/from the VPU by the PCMCIA host.

PWRDWN

Power Down control output. PWRDWN goes high in power down state. This pin is driven from the "PwrDwn" bit of the CSR.

FN0

Function number select input to the VPU. In single VPU mode, wire this pin to GND (or leave as n/c). In multi (dual)-VPU mode, this signal controls which logical function the VPU serves on a multifunction PC-Card (i.e. function 0 or 1). The internal EEPROM is disabled if the FN0 pin is set to '1' (so that the second function's EEPROM does not contend with the first function's EEPROM that holds the card's CIS).

FN0 controls the offset in attribute space, where the function's CCR's are mapped.

The FN0 line has a built-in 100K pull-down resistor.

VPUPPEN / nEXTFIFOEN

This signal is used to select the VPU Parallel Port mode. In this mode, the UART TX and RX data are accessed via an 8-bit data port rather than being serialised. This may be useful for connection to a uP/uC that does not have a built in UART. In this mode, the main baud rate generator is disabled to save power and reduce switching noise (the main oscillator still runs). The UARTs RX FIFO timeouts are also disabled in this mode. VPU Parallel Port mode is not intended to be selected and deselected dynamically.

When this signal is low, the EXTERNAL FIFO mode can be enabled by setting the VPUPPRnW signal low.

The VPUPPEN line has a built-in 100K pull-down resistor.

VPUPPD7:0 / EXTFIFOD7:0

This is the 8-bit VPU Parallel Port data bus or the EXTERNAL FIFO data bus and is bi-directional.

In VPU Parallel Port mode the direction is controlled by VPUPPRnW.
In EXTERNAL FIFO mode the direction is set depending on whether a read or write access is being performed. In the idle state or for a FIFO write cycle, the bus is in output mode. For a FIFO read cycle the bus switches to input mode.

If neither mode is used, leave these signals as no connect and tie VPUPPRnW high. See section 7 for application notes.

VPUPPRnW / nEXTFIFOEN

In VPU Parallel Port mode this signal controls the direction of the VPU Parallel Port. When the signal is low, the VPU drivers are tri-stated to allow an on-card uP to write to the UART RBR (receiver buffer register). When the signal is high, the VPU drives the port to send TX data to the uP from the THR (transmitter holding register).

In non VPU Parallel Port mode (VPUPPEN=low), this signal performs the function of enabling EXTERNAL FIFO mode if set low. Setting it high disables the mode and sets the 8 parallel port bus lines as outputs.

The VPUPPRnW line has a built-in 100K pull-up resistor.

nVPUPPINT / nR

In VPU Parallel Port mode this signal pulses low for 543ns either when a new TX data character is written into the UART THR (i.e. on the falling edge of the THRE signal) or when the host reads an RX character from the UART RBR (i.e. on the falling edge of the RD signal). It can be used to cause an interrupt in an on-card uP when the VPU needs servicing in Parallel Port mode. In 16550 mode (FIFOs enabled) the interrupt only occurs when either the TX FIFO goes full i.e. 16-bytes available or the RX FIFO goes empty (note there is not one interrupt per character).

In EXTERNAL FIFO mode this signal becomes the nR strobe (for the FIFO) that goes active (low) when a host read is performed on the UART's Scratch Register (offset 7). See section 7 for more detail.

If neither mode is required, this signal can be left as a no-connect.

VPUPPSTAT / nW

In VPU Parallel Port mode the function of this VPU Parallel Port status signal depends on VPUPPRnW. When VPUPPRnW is high in 16450 mode, VPUPPSTAT reflects the state of the internal THRE (transmitter holding register empty) signal of the UART: 0->TX data is available in the THR 1->no TX data is available. In 16550 mode the signal goes low when the TX FIFO is full i.e. there are 16-bytes ready to unload. When VPUPPRnW is low, VPUPPSTAT reflects the state of the internal RD (RX data ready) signal:0->All RX data characters have been read by the host, 1->an RX character is waiting to be read by the host.

In EXTERNAL FIFO mode this signal becomes the nW strobe (for the FIFO) that goes active (low) when a host write is performed on the UART's Scratch Register (offset 7). See section 7 for more detail.

If neither mode is required, this signal can be left as a no-connect.

nVPUPPCTRL

The function of this VPU Parallel Port control signal depends on VPUPPRnW. When VPUPPRnW is high in 16450 mode, nVPUPPCTRL is used as an active low strobe to set the (latched) internal THRE UART signal high (so informing the host that another TX character can be sent. NB: the THRE signal will be low all the time nVPUPPCTRL is low so no further host TX data will be written until nVPUPPCTRL is de-asserted). In this case the pulse should be at least 20ns wide but a maximum of 1us to avoid stalling the host TX data flow. In

16550 mode, the strobe is used to acknowledge each byte of data from the TX FIFO i.e. 16-pulses are required before the host will write more data to the TX FIFO. When VPUPPRnW is low, nVPUPPCTRL is used as an active low strobe (data stable on rising edge) to write data into the RBR (FIFO in 16550 mode) so causing the internal RD signal (latched) to be set high (informing the host that a new RX character is now available). In this case the pulse should be at least 100ns wide. In 16550 mode the host may choose to only read data after it has been triggered by the RX FIFO going past its programmed threshold (it could also look at the RD status bit).

In non VPU Parallel Port or in EXTERNAL FIFO mode, this signal should be tied high or left as n/c.

The nVPUPPCTRL line has a built-in 100K pull-up resistor.

SOUT, SIN

Standard 16550 UART serial data out and in signals.
In VPU Parallel Port mode, SIN should be tied low, SOUT should be no connect.
SIN is a Schmitt input with hysteresis to make interfacing to low slew rate signals easier.

TXRDY

This signal shows that there is data being transmitted or waiting to be transmitted. It stays high while the TX FIFO contains data, and will return low when the last bit of the last byte has been serialised. The signal can be used in RS485 half duplex mode to enable TX drivers dynamically "on the fly". No special software support is needed to use this feature if the TXRDY signal is used as the appropriate tri-state control (TXRDY='1' => enable drivers). The TXRDY signal will also assert when the BC bit in the LCR is set, ensuring that the break condition can be sent correctly if required.
For RS232 type applications this signal can be left as no connect.

nDTR, nRTS

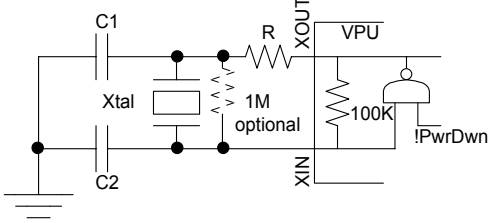
Standard 16550 modem control output signals.

nCTS, nDCD, nDSR, nRI

Standard 16550 modem status input signals.
These are all Schmitt input with hysteresis to make interfacing to low slew rate signals easier.

XIN, XOUT

Phase shift crystal oscillator connections with integrated parallel 100K bias resistor. Use an optional 1MOhm resistor in parallel with a fundamental mode crystal, and load caps appropriate to the crystal used (ensure the caps are large enough to prevent overtone oscillations over the operating temperature range). The oscillator will be disabled when the PwrDwn bit in the CSR is set (XOUT = !(XIN AND !PwrDwn)) . XIN can also be driven from an external oscillator producing a square wave with a 50% duty cycle (+/-10%). R should be selected to produce stable oscillation whilst filtering edge harmonics to an acceptable level.
The VPU16551 must use a 10MHz or greater crystal for correct UART operation. See section 8.0.



C1, C2 match to crystal per manufactures spec,
R helps reduce clock noise typ 100-200ohms

Crystal Connections

The VPU can still partly operate in PwrDwn mode…see individual register descriptions for details.

BAUDMULT

Use BAUDMULT=1 for a baud rate multiplier of 1 i.e. setting a baud rate of say 115.2KBaud will actually yield 115.2KBaud (this mode **does** give "industry standard" baud rates when used with a 14.7456MHz crystal).

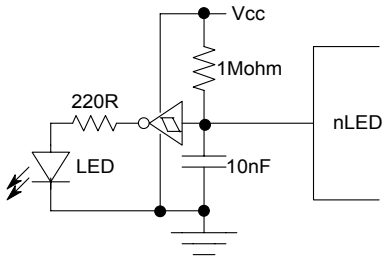
Use BAUDMULT=0 for a baud rate multiplier of 8 i.e. setting a baud rate of say 115.2KBaud will actually yield 921.6KBaud (this mode **does not** give "industry standard" baud rates).

Note that the host is unaware of this setting, it only affects the physical transmission and reception rates that the VPU uses. This also means that the host-side configuration should pick a baud rate setting that is scaled by the appropriate amount (i.e. divided by 1 or 8). If you intend to use a frequency other than 14.7456MHz, allowance must also be made for this e.g. if you require 812.5KBaud using a 13MHz crystal, you would set a multiplier of x8 and pick a baud rate from the host selection dialogue or setup file of 115.2KBaud.

☝ You **must** pull this signal high if you want industry standard baud rates. Do not leave this pin as n/c.

nLED

This signal pulses low whenever a UART register is accessed by the host. The pulse width is the same as the host's bus cycle strobe width i.e. can be as low as 185ns. This pin has an open drain driver to allow simple attachment to an RC pulse width stretch circuit as shown below.



Using a Schmitt inverter ensures good pulse stretch for LED drive

nLED Example Connections

5.0 Attribute Space Memory Map

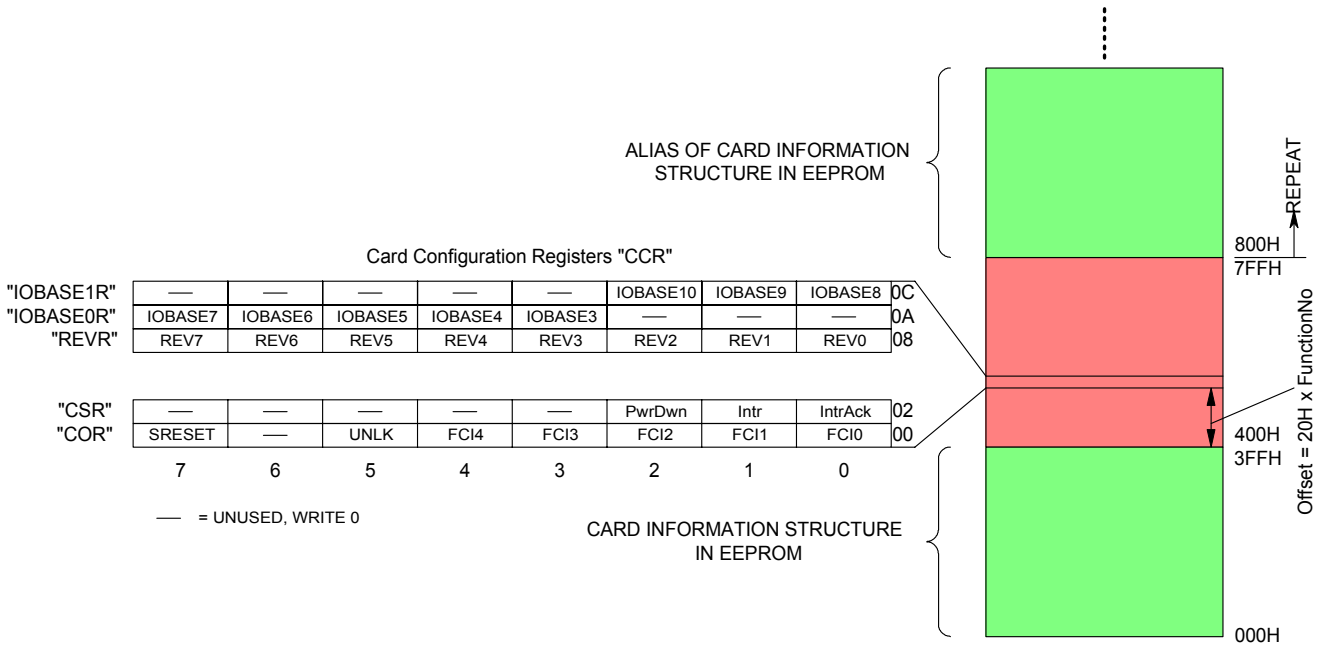


Figure 5-1 VPU16551 Attribute Space Memory Map

The registers mapped into attribute space are normally only accessed by the Operating System to configure and control the card. It is unusual for the application programmer to have to access these registers.

5.1 INTERNAL EEPROM

An internal 512byte EEPROM is mapped by the VPU between 000H and 3FFH in attribute space. This EEPROM holds the Card Information Structure for the card.
Only even bytes are implemented. Odd bytes alias their corresponding even byte.
Therefore, a total of 200h unique bytes are available for CIS tuple data.

The built-in EEPROM is only enabled if the FN0 pin is set to 0. So for multi(dual)-function cards only functions 0's EEPROM is enabled to prevent contention with the EEPROM in other function's VPU.

The EEPROM can only be written to once an "unlock" bit has been set in the COR. Once the EEPROM has been written it should be immediately re-locked (NB: the bit is volatile and returns to "locked" once power is removed or the card is reset). The EEPROM can only be written to when the VPU is powered at 5V +/-10%. The EEPROM can be written (once unlocked) using the "write read-poll" method i.e. write target byte to target address, then read the target address repeatedly until the data matches the target data or timeout if total duration exceeds 20ms. Then move on to next address until all bytes are programmed.

As mentioned in the pinout definitions above, the 400H address of the CCRs could be moved if required by connecting VPU pin A10 to a lower address line. To get 200H, wire VPU pins A10 and A9 both to signal A9 (leaving signal A10 as unused). To get 100H wire VPU pins A10, A9 and A8 all to signal A8 (leaving signals A9 and A10 as unused). Doing this will of course reduce the amount of CIS EEPROM space that is accessible.

Please contact Elan for further application information regarding In-Circuit-Programming the EEPROM before commencing your PCB layout.

5.2 Configuration Option Register "COR"

The COR serves as the main enable register for the VPU. The VPU will not respond to I/O cycles until the COR has been appropriately configured by the host.

The COR resides at offset (400H+20HxFunctionNo) in attribute space where FunctionNo is the VPU function programmed via FN0 and ranges from 0 to 1.

The bits in the COR are defined as follows:

SRESET (COR Bit7, RESET state '0')

Soft Reset. This bit when set performs the same reset action as the RESET PCMCIA signal except that it does not clear itself, and may only clear certain internal registers (see register descriptions for reset conditions).

UNLK (COR Bit5, RESET & SRESET state '0')

EEPROM write unlock bit. Set this bit high and FCI2:0 low to allow writes to the EEPROM from the host.

FCI4:0 (COR Bits4:0, RESET & SRESET state '0')

Function Configuration Index bits. These five bits control the VPU's actions as an I/O target. FCI0 enables the VPU as an IO target when set. FCI1 enables the "IOBASE0R" and "IOBASE1R" registers to control I/O hit decoding (else the VPU simply uses CE1# only and just decodes A2:0 to select one of the UART registers). FCI2 enables the VPU to drive the IREQ# line when it requires service. FCI3 & 4 can be used to define extra logical configurations for the function.

For a single function card with a single VPU, the CIS should describe Configuration Entry Numbers (which will be written to FCI4:0 by the host) with FCI2:0 set to 101b, (enable the function, do not use IOBASE1:0R, enable IREQ# generation). FC4:3 are used to identify each configuration. For example, a card with 4 logical configurations would define four Configuration Table Entry Tuples with indexes of 05H, 0DH, 15H and 1DH. Fifth, sixth etc configurations 25H, 2DH etc would also be valid despite over-lapping the UNLK bit because FCI2:0 are non-zero hence keeping the lock intact.

For multi(dual)-function applications with more than one VPU, the CIS' Configuration Entry Numbers should have FCI2:0 set to 111b to additionally enable the IOBASE1:0R hit decoder (so that each function gets its own unique set of 8 I/O addresses). It is the host's responsibility to correctly configure each function's IOBASE1:0R registers with a valid 8-byte aligned I/O address (normally Card Services does this).

For best plug-n-play compatibility, all logical configurations should <u>not</u> request fixed I/O base addresses. Instead at least one configuration should request a block of I/O located on an 8-byte boundary anywhere in I/O space.

<u>Memory Mapped Mode:</u>

For hosts that do not explicitly define an IO space, the VPU16551 can operate in a Memory Mapped Mode (i.e. IORD# and IOWR# can be tied high). The VPU's UART registers that are normally accessible only via IO space can also be accessed from Common Memory space starting at offset 0h and with the same offsets as the IO space organisation. Additionally, the Card Configuration Registers that are normally mapped only to Attribute Memory space can also be accessed from Common Memory space at offset 400h in the same order and spacing as the attribute space organisation. For a normal "COM" port type card these two features will be largely unimportant and indeed, the CIS would be unlikely to include a Configuration Entry that defined a memory mapped configuration because most operating systems cannot deal with a memory mapped UART. However, for platforms such as the Handspring Visor, the VPU could be used in a pure Common Memory configuration so long as the driver software was written to use memory accesses to the UART/CCRs. Note that Common Memory access is to the UART registers is always possible regardless of the bit settings in FCI4:0. In a memory only configuration, connect the IORD# and IOWR# pins to a logic '1'. The CE1# signal can be used with an external decoder to map the registers into the host memory space. The REG# signal must be used to differentiate between common and attribute memory accesses. If no attribute accesses are required, this line

should be wired to a logic '1' allowing only common memory accesses. If attribute accesses are required, this signal could be controlled from a decoder or from a register bit.

5.3 Configuration And Status Register "CSR"

The CSR resides at offset (402H+20HxFunctionNo) in attribute space where FunctionNo is the VPU function programmed via FN0 and ranges from 0 to 1.

PwrDwn (CSR Bit2, RESET & SRESET state '0')

Power down bit. Setting this bit high gates the main oscillator (and jams Xout high) and hence stops all clocking activity in the VPU. This reduces power consumption. The PCMCIA interface is still functional in this state, but all UART serialiser and FIFO functions will be non-operational due to the lack of clock. Modem control signals are still active. The same logic applies if an external clock is fed into the VPU via Xin.

Intr (CSR Bit1, RESET & SRESET state '0')

Interrupt status bit. This bit reflects the VPU's internal interrupt request state (high = request pending). This bit is not influenced by FCI2 (which when low simply stops the VPU's IREQ# signal from being asserted). This bit is read only.

IntrAck (CSR Bit0, RESET & SRESET state '0')

Interrupt acknowledge bit. This bit controls the interrupt acknowledge scheme. If it is 0, writing to the VPU's Intr bit will have no effect on this VPU or any other in a multi-function case (nINTRACK will not assert). If IntrAck is a 1, then writes of 0 to the Intr bit are as described above.

5.4 VPU Revision Register "REVR"

The REVR resides at offset (408H+20HxFunctionNo) in attribute space where FunctionNo is the VPU function programmed via FN0 and ranges from 0 to 1. The register is read only and reports the silicon revision in the bottom 4 bits.

5.3 I/O Base Address Register "IOBASE0R" & "IOBASE1R"

The IOBASE0R resides at offset (40AH+20HxFunctionNo) in attribute space and IOBASE1R resides at offset (40CH+20HxFunctionNo) in attribute space where FunctionNo is the VPU function programmed via FN0 and ranges from 0 to 1.

IOBASE10:4 (IOBASE0R Bits7:3+IOBASE1R Bits2:0, RESET & SRESET state '0')

This 8-bit (combined) register holds the I/O base address at which the VPU's UART registers will be mapped. FCI1 controls whether I/O hit decoding uses this base address, or whether the base address is ignored and I/O hits are decoded purely with CE1# (as with a single function implementation). When the VPU uses this address, the bottom 3 bits are ignored because the UART maps in eight x 8-bit wide I/O registers. The host is responsible for mapping the base address to an 8 byte boundary, using information it reads from the CIS about how many address lines are decoded by the function (always 3).

In a host system that decodes more than 11-bits for I/O space accesses, the VPU's I/O registers will alias every 800H bytes unless the host PCMCIA bridge itself does complete I/O decoding to qualify the addresses passed to the VPU (which is common).

6.0 I/O Map

The system I/O base address where the VPU's UART registers are mapped is defined by host software. Therefore, all discussions will refer to registers at a certain offset relative to this base address.

6.1 UART Block Diagram

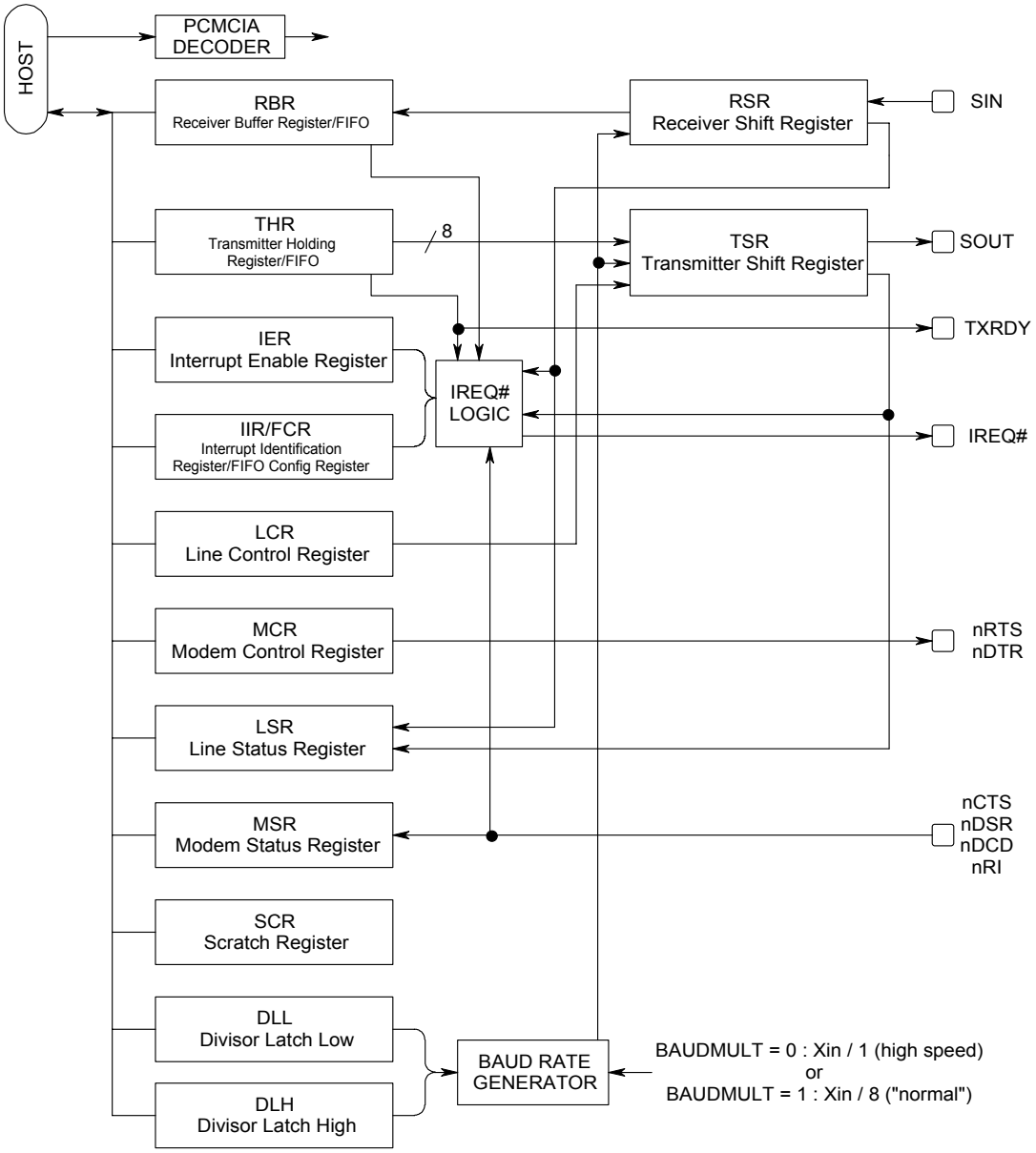


Figure 6.1-1 UART Block Diagram

6.2 Receiver Buffer Register/FIFO "RBR"

Offset:00H when DLAB = 0, Read only, reset state 00H, In PwrDwn state: inactive

7	6	5	4	3	2	1	0
RXD7	RXD6	RXD5	RXD4	RXD3	RXD2	RXD1	RXD0

This is the received data character. RXD0 is the least significant bit (received serially first).
The data arrives here from the Receiver Shift Register.
The RBR is accessible by the host when the DLAB bit is 0.
In 16450 mode, the RBR is a single byte storage for incoming characters. In 16550 mode the RBR is a 16-byte deep RX FIFO, with each read to the RBR accessing progressively more recent data.
When data arrives in the RBR, the RD bit is set in the LSR (may cause an interrupt if enabled via the IIR). In 16550 mode the RD bit remains set until all the FIFO data is read out of the RBR.

6.3 Transmitter Holding Register/FIFO "THR"

Offset:00H when DLAB = 0, Write only, reset state 00H, In PwrDwn state: inactive

7	6	5	4	3	2	1	0
TXD7	TXD6	TXD5	TXD4	TXD3	TXD2	TXD1	TXD0

This is the data character to transmit. TXD0 is the least significant bit (sent serially first).
The data is written here by the host.
In 16450 mode the data is copied from here directly into the Transmitter Shift Register, or can be accessed from the VPU Side-Band Parallel Port. In 16550 mode, the data written to the THR enters the TX FIFO and only goes into the TSR when it is the next byte to be transmitted i.e. it is the oldest byte.
The THR is accessible by the host when the DLAB bit is 0.
In 16450 mode, when the data has been sent from the THR to the TSR, the THRE flag is set in the LSR (may cause an interrupt if enabled via the IIR). In 16550 mode, the entire FIFO must be empty before THRE goes active.

6.4 Interrupt Enable Register "IER"

Offset:01H when DLAB = 0, Read/write, reset state 00H, In PwrDwn state: active

7	6	5	4	3	2	1	0
0	0	0	0	MSIen	RLSIen	THREIen	RDIen

This register enables various interrupt sources that the UART may generate.
The IER is accessible when the DLAB bit is 0.
RDIen: When set, enables interrupt when the RD flag in the LSR is set (receive data available).
THREIen: When set, enables interrupt when the THRE flag in the LSR is set (transmitter holding register empty).
RLSIen: When set, enables interrupt when any of the OE, PE, FE or BI flags in the LSR is set (receiver line status).
MSIen: When set, enables interrupt when any of the DCTS, DDSR, TERI or DDCD flags in the MSR is set (modem status).

6.5 Interrupt Identification Register "IIR"

Offset:02H, Read only, reset state 01H, In PwrDwn state: partly active (no data or timeouts)

7	6	5	4	3	2	1	0
16550Md	16550Md	0	0	IIC2	IIC1	IIC0	nINT

This register is used to identify with a fixed priority, which interrupts are pending.
Only interrupts that are enabled via the IER will be reported in the IIR.
nINT: is low whenever an interrupt is pending (the "Int" bit in the CSR is the complement of this bit).
IIC2:0: form an interrupt ID code which is prioritised in the following way:

IIC2	IIC1	IIC0	PRIORITY (1=highest)	Interrupt Type	Reset Method
0	1	1	1	RLSI	Read LSR
0	1	0	2	RDI	Read RBR
1	1	0	2	CTI	Read RBR
0	0	1	3	THREI	Write THR OR read IIR*
0	0	0	4	MSI	Read MSR

*The action of having read the IIR and got an interrupt ID code of 1 (THREI) will have also cleared the THREI interrupt (this has no effect on the THRE bit in the LSR).

RLSI: Receiver Line Status Interrupt i.e. a receiver error condition

RDI: Received Data Interrupt i.e. an RX char is available in 16450 mode, or the RX FIFO threshold has been reached/passed in 16550 mode.

CTI: Character Timeout Interrupt (16550 mode only) : one or more unread RX chars has been present in the RX FIFO for four or more complete character periods without there being any new RX characters put into the FIFO or RX characters taken out of the FIFO by the host reading the RBR. CTI is disabled in VPU Parallel Port mode.

THREI: Transmitter Holding Register Empty Interrupt i.e. the THR has gone empty in 16450 mode or the TX FIFO has been emptied in 16550 mode.

MSI: Modem Status Interrupt i.e. a change of state on the Modem Status Lines has been detected.

16550Md: these two identical bits show, when set, that the VPU's TX and RX FIFOs are enabled via the FCR

6.6 FIFO Configuration Register "FCR"

Offset:02H, Write only, reset state 00H, In PwrDwn state: inactive

7	6	5	4	3	2	1	0
RXLVL1	RXLVL0	0	0	0	TXReset	RXReset	16550Md

This register is used to configure and enable the 16550 FIFOs.
16550Md: setting this bit enables the TX and RX FIFOs for 16550 mode operation, clearing it sets 16450 mode.
RXReset: writing a '1' to this bit clears the RX FIFO logic and the FIFO itself. This bit is self clearing.
TXReset: writing a '1' to this bit clears the TX FIFO logic and the FIFO itself. This bit is self clearing.
RXLVL1:0:forms a 2 bit code defining the minimum number of RX FIFO bytes that will trigger an RDI. 00:= 1 byte, 01=4 bytes, 10=8 bytes, 11=14 bytes. When the RX FIFO fills to, or passes, this threshold an RDI will be generated if enabled via the IER. Note that the RD bit in the LSR goes active when **any** data is in the RX FIFO and is not conditional on this threshold setting, allowing polled mode operation without interrupts.

6.7 Line Control Register "LCR"

Offset:03H, Read/write (no effect in VPU Parallel Port mode) , reset state 00H, In PwrDwn state: don't care

7	6	5	4	3	2	1	0
DLAB	BC	SP	EPS	PEn	NSB	WLS1	WLS0

This register is used to control the transmission / reception format for serial communications.
DLAB: When set allows read / write access to the two baud rate divisor latches (divisor latch access bit).
BC: When set, asynchronously forces the SOUT pin to 0 (break control) and causes the TXRDY pin to assert.
SP: When set, and PEn is set, the parity bit is forced to the complement of the EPS bit (stick parity).
EPS: If PEn is set, parity is even if EPS is set, and odd if EPS is cleared (even parity select).
PEn: When set, a parity bit is generated / checked in TX & RX serial communications (parity enable).
NSB: When set, the number of stop bits generated in TX serial communications will be 1.5 for 5-bit characters and 2 for 6,7 and 8-bit characters. If clear, one stop is transmitted under all conditions (number of stop bits). The receiver only checks for the presence of 1 stop bit under any condition.
WLS1:0: Forms a 2-bit code to select the character length generated / checked in TX & RX serial communications; 00b:5-bit, 01b:6-bit, 10b:7-bit, 11b:8-bit (word length select)

6.8 Modem Control Register "MCR"

Offset:04H, Read/write, reset state 00H, In PwrDwn state: active

7	6	5	4	3	2	1	0
0	0	0	LOOP	OUT2	OUT1	RTS	DTR

This register is used to control the modem control lines.
LOOP: When set puts the UART into a test mode: SOUT is set to 1, SIN is disconnected, TSR output is internally looped back to RSR input, nCTS, nDSR, nDCD, nRI inputs are disconnected and internally looped back to nDTR, nRTS, nOUT1, nOUT2 and these four output pins are forced high.
OUT2: Bit is implemented but not used.
OUT1: Bit is implemented but not used.
RTS: The complement of this bit appears on the nRTS pin.
DTR: The complement of this bit appears on the nDTR pin.

6.9 Line Status Register "LSR"

Offset:05H, Read only, reset state 60H, In PwrDwn state: don't care

7	6	5	4	3	2	1	0
ALOE	TEMT	THRE	BI	FE	PE	OE	RD

This register is used to report the status of the most recent serial operation, or in 16550 mode, the status relating to the data to be read next from the FIFO (the RX FIFO is 11-bits wide and records the data byte plus PE/FE/BI bits).
ALOE: Set in 16550 mode when there is at least one error bit set in the FIFO (PE/FE/BI). Always 0 in 16450 mode.
TEMT: Set when both the transmitter holding resister and shift register are empty (transmitter empty). Same as THRE in VPU Parallel Port mode.
THRE: Set in 16450 mode when the transmitter holding register has been moved to the TSR or read via the VPU Parallel Port. In 16550 mode this bit shows that there is no data in the TX FIFO.
BI: Set when SIN was zero for a whole character receive period (i.e. start+length+parity+stop). Cleared by reading the LSR (break indicator). Forced to 0 in VPU Parallel Port mode.
FE: Set when the bit cell in the receive data where the stop bit was sampled contained a 0 rather than a 1. Cleared by reading the LSR (framing error). Forced to 0 in VPU Parallel Port mode.
PE: Set when the received parity bit does not match the parity bit calculated by the receiver. Cleared by reading the LSR (parity error). Forced to 0 in VPU Parallel Port mode.
OE: Set when a new RX character is moved into the RBR while the RD flag is already set. Cleared by reading the LSR (overrun error). Forced to 0 in VPU Parallel Port mode.
RD: Set when an RX character is moved into the RBR/RX FIFO from the RSR or from the VPU Parallel Port. Cleared by reading the RBR in 16450 mode, or by reading all the RX bytes in the RX FIFO in 16550 mode (receive data (available)).

6.10 Modem Status Register "MSR"

Offset:06H, Read only, reset state ?0H, In PwrDwn state: active

7	6	5	4	3	2	1	0
DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS

This register is used to report the levels on the modem status signals.
DCD: Reads as the complement of nDCD signal, or the OUT2 bit of the MCR in LOOP mode.
RI: Reads as the complement of nRI signal, or the OUT1 bit of the MCR in LOOP mode.
DSR: Reads as the complement of nDSR signal, or the RTS bit of the MCR in LOOP mode.
CTS: Reads as the complement of nCTS signal, or the DTR bit of the MCR in LOOP mode.
DDCD: Reads as 1 when a change on nDCD has occurred since last read of MSR. Cleared by reading MSR.
TERI: Reads as 1 when a rising edge on nRI has occurred since last read of MSR. Cleared by reading MSR.
DDSR: Reads as 1 when a change on nDSR has occurred since last read of MSR. Cleared by reading MSR.
DCTS: Reads as 1 when a change on nCTS has occurred since last read of MSR. Cleared by reading MSR.

6.11 Scratch Register "SCR" / EXTERNAL FIFO Port

Offset:07H, Read/write, reset state 00H, In PwrDwn state: as SCR: active else: inactive

7	6	5	4	3	2	1	0
SCR7	SCR6	SCR5	SCR4	SCR3	SCR2	SCR1	SCR0

Normally, this register is a general purpose scratch register that has no effect on the VPU's operation.
In EXTERNAL FIFO mode, the SCR is disabled and instead access to offset 07h results in a data transfer to or from the EXTFIFOD7:0 pins. A read access sets the EXTFIFOD7:0 as inputs and passes the data on these signals back to the host, with the read strobe appearing at the nR pin (nVPUPPINTR pin). For write cycles, the data from the host appears on EXTFIFOD7:0 with the write strobe appearing on the nW pin (nVPUPPSTAT

pin). Note that the data from the write cycle is latched inside the VPU and so will persist on the data lines after the write cycle has completed.


See section 7 for advice on how to use the EXTERNAL FIFO mode.

6.12 Divisor Latch Low / High "DLL" / "DLH"

DLL Offset:00H when DLAB = 1, Read/write, reset state 00H, In PwrDwn state: don't care

7	6	5	4	3	2	1	0
DIV7	DIV6	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0

DLH Offset:01H when DLAB = 1, Read/write, reset state 00H, In PwrDwn state: don't care

7	6	5	4	3	2	1	0
DIV15	DIV14	DIV13	DIV12	DIV11	DIV10	DIV9	DIV8

This combined register controls the output frequency of the VPU's baud rate generator.

DIV15:0: The baud rate generator will output (Fxin÷BAUDMULTfactor)÷(16x**(DIV15:0)**) Hz. A divisor of 0 is invalid. BAUDMULT factor is 1 if the BAUDMULT pin = 1, or 8 if the BAUDMULT pin = 0. Some example divisors are given below for a 14.7456MHz crystal.

BAUD RATE		**DIV15:0 (decimal)**
BAUDMULT pin		
'1'	'0'	
300	2400	384
2400	19200	48
4800	38400	24
9600	76800	12
19200	153600	6
38400	307200	3
57600	460800	2
115200	921600	1

Other crystal frequencies would yield different baud rates and would not match the industry standard.

7.0 VPU Parallel Port / External FIFO Mode

7.1 VPU Parallel Port Mode

When VPU Parallel Port mode is used, the serial TX and RX mechanisms are disabled. Instead, TX and RX data bytes are exchanged under handshake control directly with the RBR and THR registers (or the FIFOs in 16550 mode).
Two methods can be used to initiate transfers from an on-card micro controller: polled or interrupt driven.

The connection of VPUPPD7:0 to a processors' parallel port should be done through 10K resistors. This way, each time the processor switches the VPU from read to write, no period of contention or float will be incurred that could lead to extra power consumption.

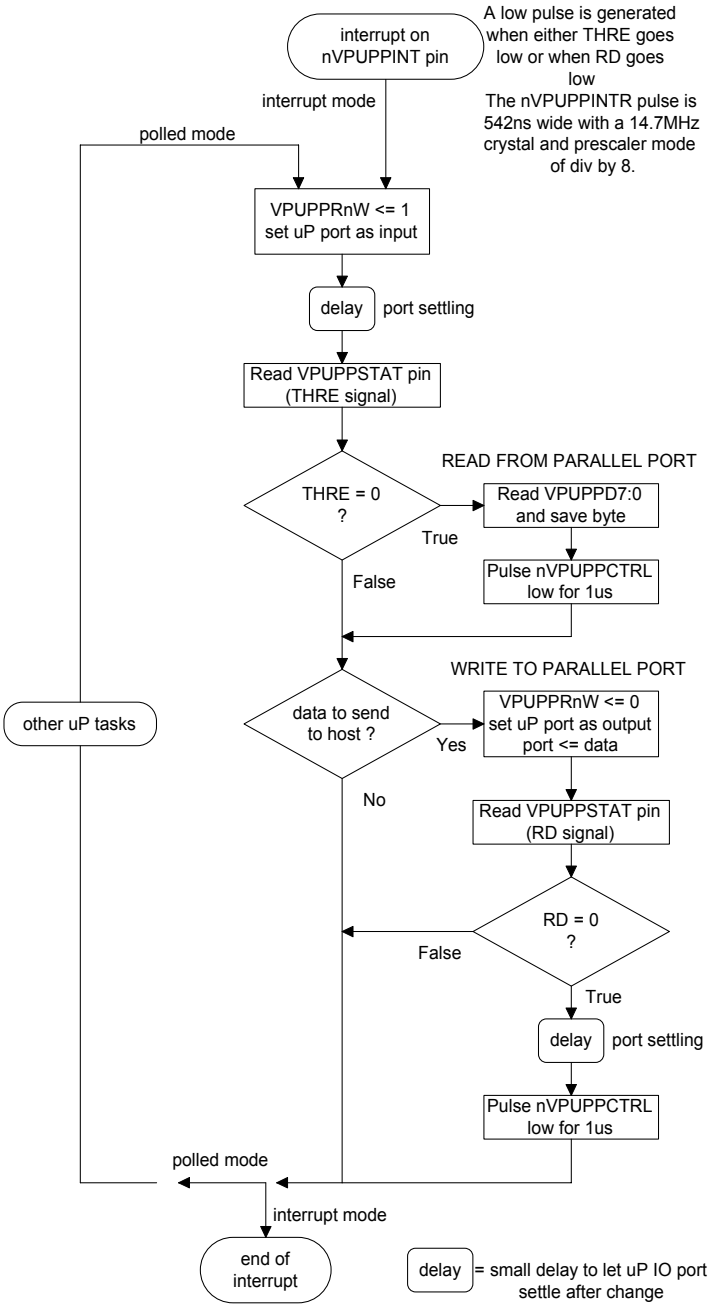


Figure 7.0-1 Parallel Port Operational Flow for 16450 Mode

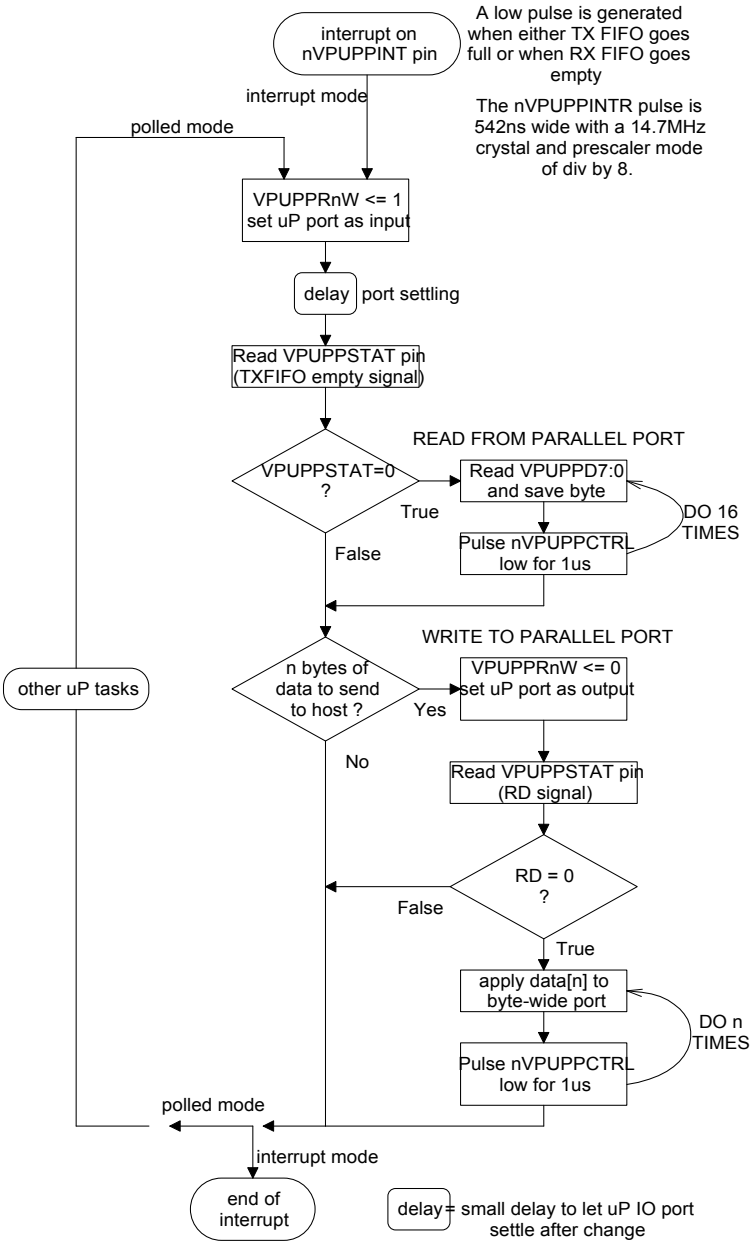
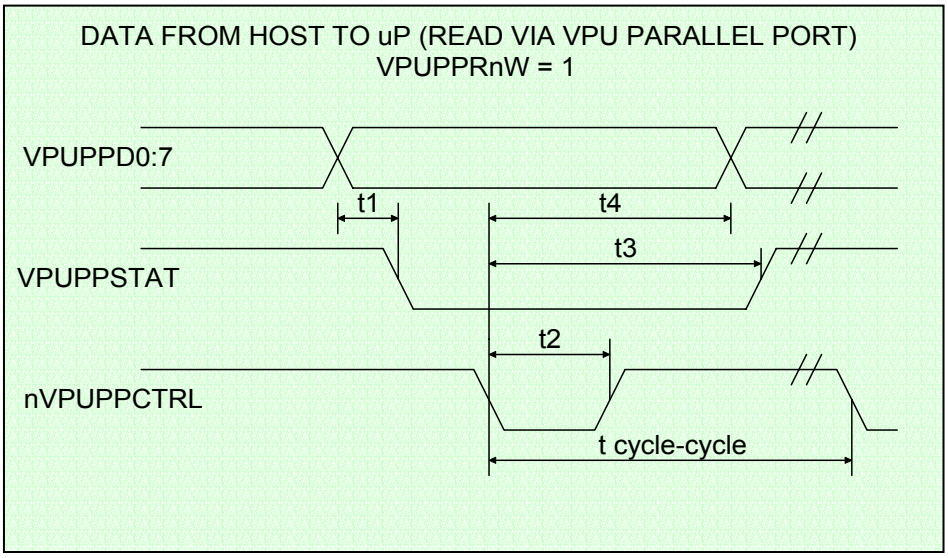


Figure 7.0-2 Parallel Port Operational Flow for 16550 Mode

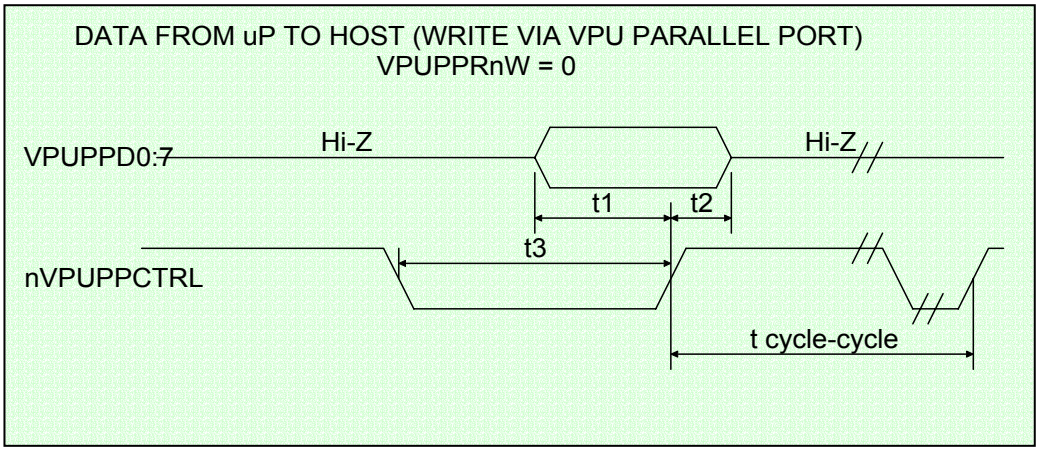
Notes on VPUPP operation in 16550 Mode:

1. The host software and the card-side uP **must** know how many bytes to transfer. For TX (i.e. data from host to uP), the uP will only see the VPUPPSTAT goes low when 16-bytes are ready. This way, the host and the uP don't compete for the FIFO; the host writes 16-bytes and waits for the next FIFO empty indication, whereas the uP waits for the next full indication and then unloads 16-bytes. For RX (i.e. data from uP to host) the host will have configured the FIFO trigger point to one of the values shown in the FCR section. The uP must know this setting (hard-code this or transmit it to the uP using your own proprietary scheme) so that it only loads just enough bytes to trigger the host to fetch them. Again, this stops the uP and the host from competing for the FIFO.
2. Because the TX path must work in 16-byte chunks, the application on the host must be structured for this. Trying to send less than 16-bytes will stall the transfer process because the TX FIFO will never go full.
3. Expect no more than around 10 to 20 kilobytes per second throughput (depends on host and software)
4. If the above operation is too complex, use the port in 16450, where single bytes are transferred.
5. Different O.S.s may display slightly different behaviour when gathering data. You are advised to test across multiple platforms under multiple conditions.



Parameter	min (ns)	max (ns)	Description
t1	16	80[3]	Data stable before VPUPPSTAT low
t2	20	1000[4]	Strobe width to acknowledge TX data
t3	150	150+ t_{Xout}[5]	Delay from strobe assert to VPUPPSTAT de-assert (THRE)
t4	45	45+t_{Xout}[6]	Data hold time from strobe assert
$t_{cycle-cycle}$	3 x t_{Xout}[6]	-	Minimum read-to-read period to allow correct internal FIFO operation

[3] VPUPPSTAT will only transition low in 16550 mode when the TX FIFO transitions to full (this transition will also cause a pulse on the nVPUPPINT pin in 16550 mode).
[4] 1us is advisory. Keeping the strobe this wide will ensure all other "delays" (t3/t4) are transparent to the uP. Note that this TX data acknowledge pulse is edge sensitive (i.e. it cannot be tied low to acknowledge all TX bytes)
[5] VPUPPSTAT only signals that the TX FIFO has gone full with 16-bytes of data, ready to be unloaded by the uP. It will de-assert after the first byte is read.
[6] t_{Xout} is the period of the main crystal oscillator



Parameter	min (ns)	max (ns)	Description
t1	20	-	Data setup before strobe rising edge
t2	0	-	Data hold after strobe
t3	20	1000[7]	Strobe width to load RX data byte to RBR
$t_{cycle\text{-}cycle}$	3 x t_{Xout}[6]	-	Minimum write-to-write period to allow correct internal FIFO operation

[7] 1us is advisory, but pulse width may be longer depending on data rate required. Host will not see that new data is available in the RBR until strobe is de-asserted causing RD bit to go active in LSR. In 16550 mode the uP can write up to 16-bytes into the RX FIFO and then poll the VPUPPSTAT signal to see when the host has read all the data (a pulse will be generated on the nVPUPPINT pin as the RX FIFO goes empty in 16550 mode). On WindowsCE / PocketPC and other O.S.s, writing more data immediately that the FIFO shows as empty can cause "synchronisation" problems. This is because the interrupt service routine "re-checks" for RX data sometime after emptying the FIFO and will proceed to unload this data too if it has the chance (meaning that the uP now does not know how many bytes are in the RX FIFO and so does not know how many bytes to write to make the next RX FIFO data available interrupt to the host. This will cause the transfer process to stall (both side are waiting !)) . Therefore, a hold-off period may be required to allow the host's ISR to completely finish its processing, before the uP writes more data to the RX FIFO. Alternatively, write a custom device driver that behaves exactly the way you need to maximise throughput.

7.2 VPU Parallel Port "Side-band" Mode

Under the following conditions, the VPUPPD7:0 parallel port lines will output the 8 bits of the last TX character:

1. LCR set for 5-bit characters and 1.5 stop bits
2. VPU Parallel Port mode **not** enabled
3. External FIFO mode **not** enabled

The 8 port lines default to 00h at power up. They are driven from transparent latches that are only transparent in the above condition. In all other conditions they hold either 00h or the last transmitted byte.

This means that the 8 lines can be used as a byte-wide "bit twiddled" output port for general purpose use. The following sequence demonstrates how:

1. Call O.S. serial API to set UART to 5-bit characters and 1.5 stop bits
2. Send a sequence of TX characters to affect your desired bit pattern sequence on VPUPPD7:0
3. Call O.S. serial API to set UART back to "normal" transmission characteristics perhaps 8-bit, 1 stop

The last character from step 2 will persist on the 8 lines. All characters sent in this way will **also** be transmitted serially.

The time interval between the changes of state of the lines is governed by the serial character transmission rate in the normal way.

The advantage of this port mode is that it requires only regular O.S. serial port API calls to affect.

7.3 External FIFO Mode

For applications that require higher data throughput than can provided by the standard 16550 architecture, the VPU16551 offers an external FIFO mode. This mode co-exists with the normal 16550 UART and does not affect the normal 16-byte FIFOs used for serial transmission and reception.

7.3.1 Throughput Rates and Large FIFOs

The following approximate table and graph show how FIFO size affects throughput in an interrupt driven configuration (the most system efficient setup for a multi-tasking operating system).

The calculations assume that a bus cycle to access a FIFO byte takes approx. 1.0us, and that after half of the FIFO depth has been emptied in the interrupt service routine, the routine then polls the FIFO's empty flag to fetch any further data that has arrived, until the FIFO is empty. As long as the bus cycle time is fast in comparison with the data input rate, then this allows the processor to spend part of its time doing "other" tasks like painting the screen, or processing the streamed data. Consuming too much effort in an interrupt service routine is a bad idea as it effectively freezes the host from doing other tasks.

For a typical PC based system, interrupt latencies are typically of the order of low 10's of milliseconds.

What can be seen from the graph below is that increasing from 16 to 64 to 128 bytes of FIFO depth pays off for systems with very low interrupt latencies. But, as the latency increases the 3 curves all tend to throughputs that are very modest indeed.

What can also be seen from the graphs is that the only way to protect against such latency is to add **much** larger FIFOs than a normal UART provides. For improved throughput, a dual FIFO could be considered although the returns diminish for such an addition.

Worst interrupt latency (ms)	FIFO depth	Throughput rate Kbytes/sec
0.1	16	64.52
1	16	7.81
10	16	0.80
20	16	0.40
0.1	64	163.27
1	64	29.20
10	64	3.17
20	64	1.59
0.1	128	219.18
1	128	53.69
10	128	6.28
20	128	3.17
0.1	2048	322.82
1	2048	251.47
10	2048	78.34
20	2048	44.38
0.1	4096	327.99
1	4096	286.67
10	4096	126.86
20	4096	78.34
0.1	16384	331.98
1	16384	320.30
10	16384	236.93
20	16384	183.78

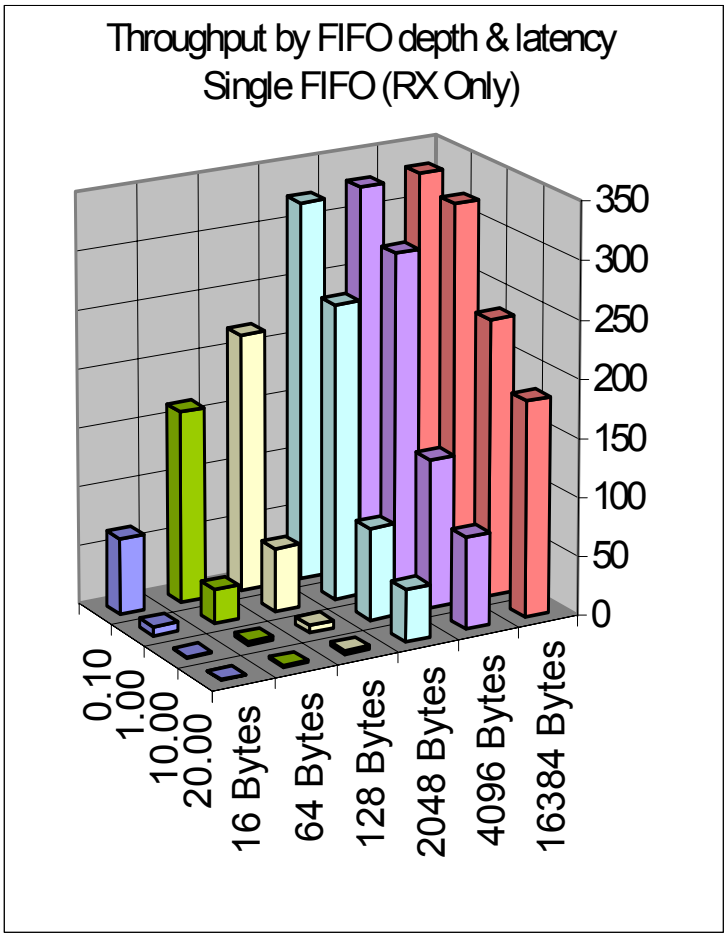


Figure 7.3.1-1 Single FIFO throughput rates (RX only)

The above figures should be considered as illustrative only due to the assumptions made to simplify their calculation. The rates you get may vary considerably from those shown above depending on the exact application.

Worst interrupt latency (ms)	FIFO depth	Throughput rate Kbytes/sec
0.1	16	162.16
1	16	22.90
10	16	2.39
20	16	1.20
0.1	64	328.77
1	64	80.54
10	64	9.42
20	64	4.75
0.1	128	396.69
1	128	138.73
10	128	18.49
20	128	9.42
0.1	2048	491.99
1	2048	430.01
10	2048	190.29
20	2048	117.50
0.1	4096	495.96
1	4096	462.37
10	4096	275.66
20	4096	190.29
0.1	16384	498.98
1	16384	490.03
10	16384	415.47
20	16384	355.39

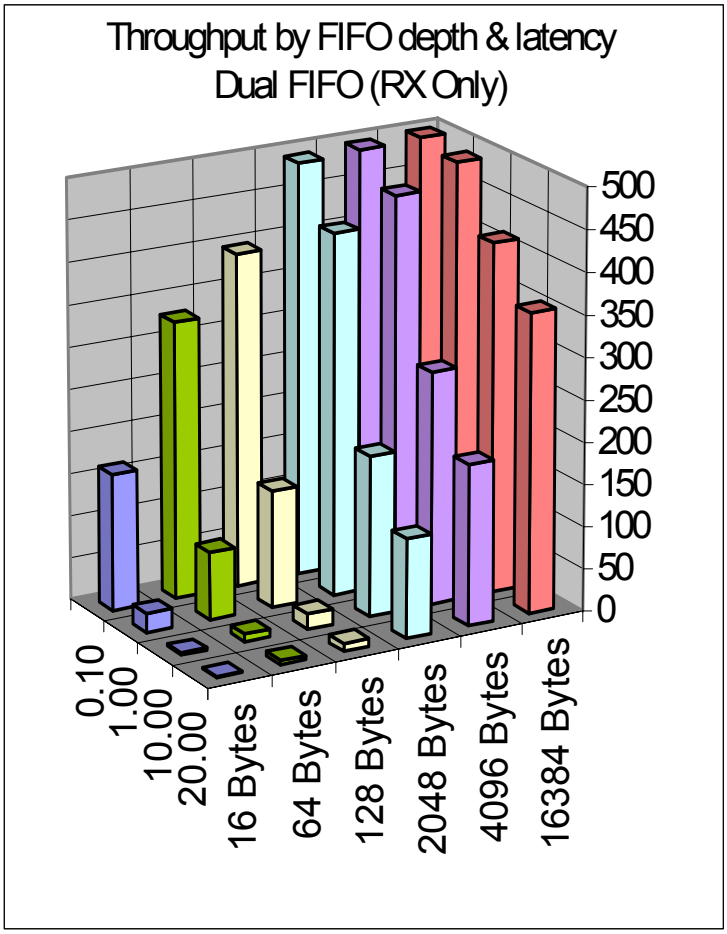


Figure 7.3.1-2 Dual FIFO throughput rates (RX only)

The above figures should be considered as illustrative only due to the assumptions made to simplify their calculation.
The rates you get may vary considerably from those shown above depending on the exact application.

7.3.2 Practical Applications

7.3.2.1 Software Drivers

EXTERNAL FIFO mode can only be used with special device drivers. Whilst the UART still works in the regular way, using the high speed FIFO data path requires special handling. For this purpose Elan can supply a development kit that contains drivers for Windows PocketPC and a WDM for Win9x / 2K / XP. The drivers make certain assumptions about the way the FIFOs are wired up to the VPU. See the Hardware section for more detail.

7.3.2.2 Hardware Configuration

The EXTERNAL FIFO mode works by shadowing the UARTs Scratch Register. When the mode is enabled, reading and writing the Scratch Register at offset 07H, actually accesses one or mores FIFOs connected to EXTFIFOD7:0 pins. There are various ways to configure EXTERNAL FIFO mode depending on your system requirements.

The figures below show recommended configurations for a bi-directional scheme and for a scheme with two forward FIFOs (the return channel being only the normal serial UART channel).

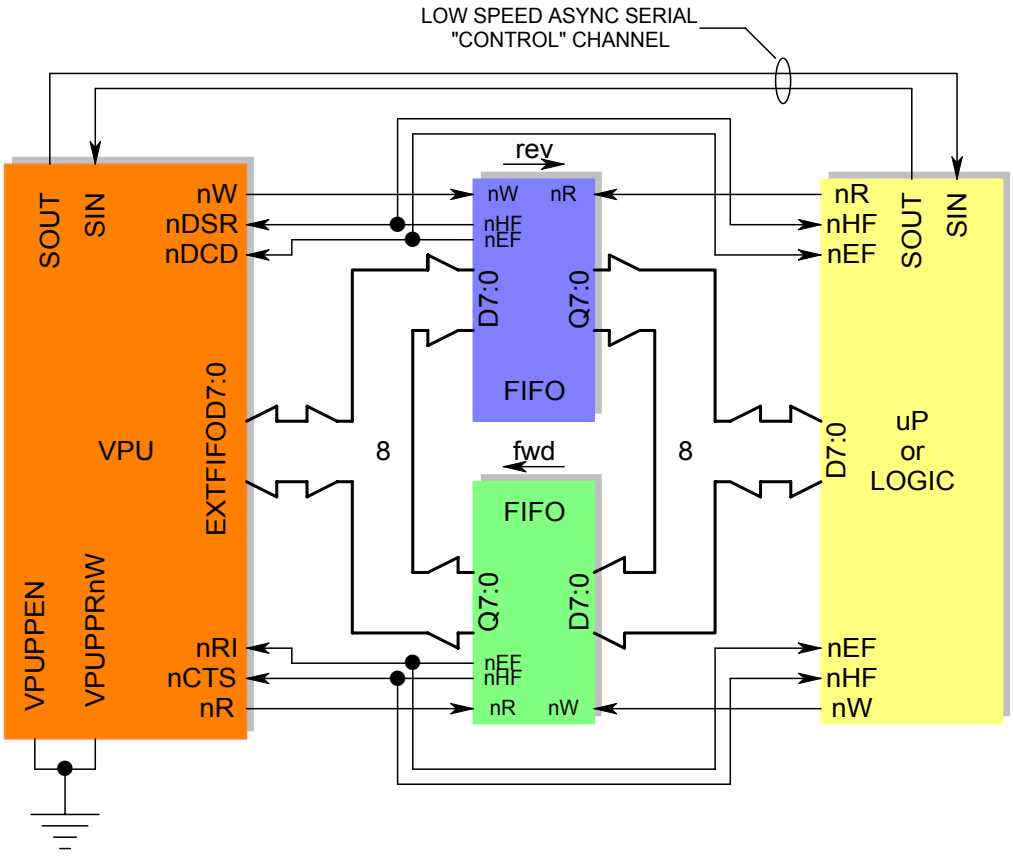


Figure 7.3.2.2-1 Recommended Bi-Directional FIFO Connections

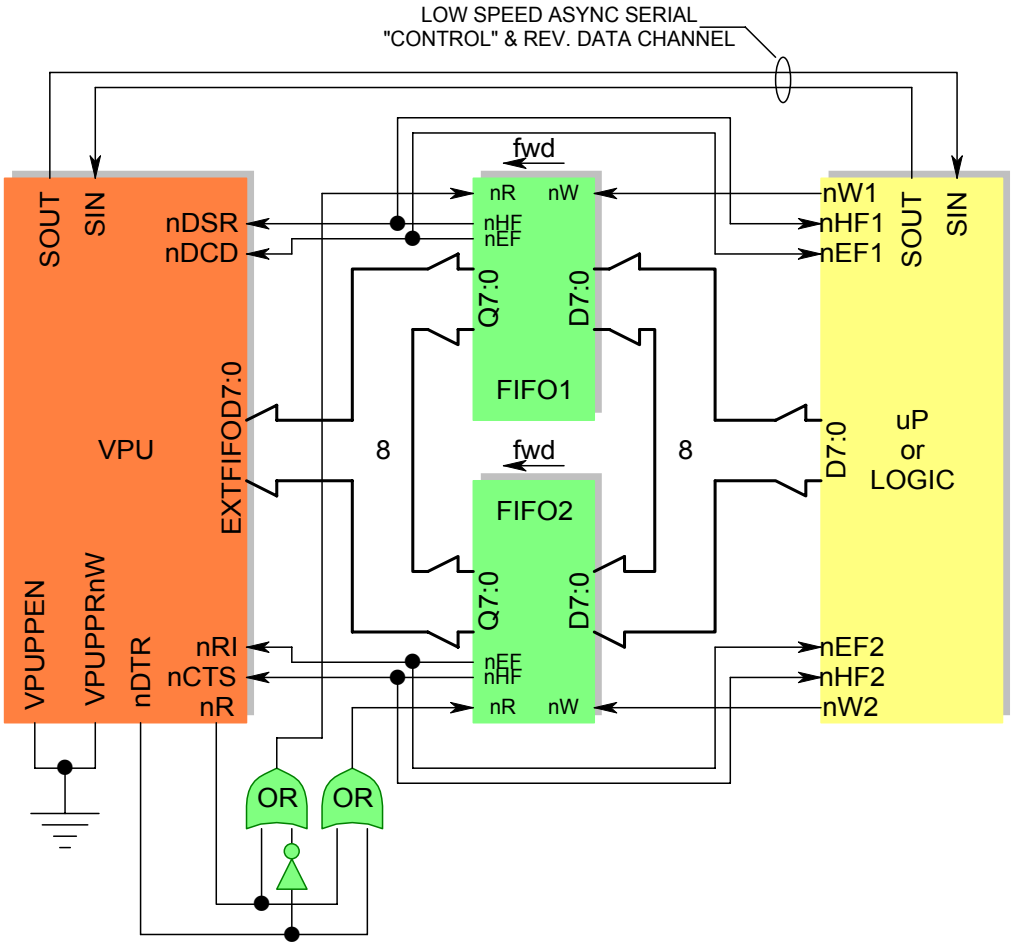
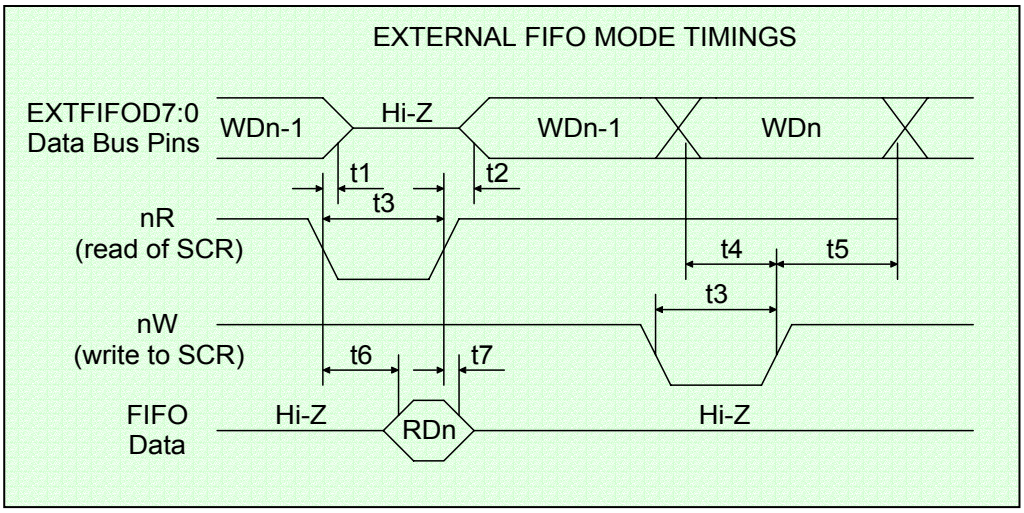


Figure 7.3.2.2-2 Recommended Uni-Directional Dual FIFO Connections

7.3.3 EXTERNAL FIFO Mode Timings



Parameter	min (ns)	max (ns)	Description
t1	6	12	Data bus tri-state after start of Read cycle
t2	3	6	Data bus re-assert after Read cycle
t3	165	600[8]	Read cycle strobe width
t4	40	-	Write cycle data bus setup time
t5	-[9]	-[9]	Data bus hold time after Write cycle
t6	12[10]	50	FIFO data assert after start of Write cycle
t7	0	6[10]	FIFO data de-assert time after Write cycle

[8] The cycle width is controlled by the host, not the VPU. See the PCMCIA timing specification for IO Read Cycles.
[9] The data will persist after a Write cycle until the next Write cycle, or will de-assert during the next Read cycle.
[10] This figure aims to avoid any bus contention as the VPU de-asserts / asserts its data bus pins while the external FIFO is starting to assert / de-assert its data onto the bus. It may be worth connecting the fwd. FIFO(s) to the VPU via 1K series resistors if there is the chance of the bus drivers overlapping for some period (i.e. if t6 < t1 or t7 > t2)

8.0 Electrical Specifications

8.1 ABSOLUTE MAXIMUM RATINGS

Parameter	Limit(s)
Supply voltage at Vcc pin relative to GND	-0.5v to +7.0v
Input voltage, any input relative to GND	-0.5v to (Vcc+0.5v)
Output voltage, any output relative to GND	-0.5v to (Vcc+0.5v)
Output source / sink current	±10mA
Storage temperature	-65°C to +150°C
I/O Pin ESD Rating (Human Body Model)	±2Kv

Stresses at or beyond these limits may cause permanent damage to the device.

8.2 RECOMMENDED OPERATING CONDITIONS

Parameter	Recommended Operating Condition
Supply voltage at Vcc pin relative to GND	2.97v to 5.50v
Operating temperature	0°C to +70°C

8.3 ELECTRICAL CHARACTERISTICS, Vcc = 3.3v

Parameter	Characteristic	
	Min	Max
V_{OH} (I_{OH} = -6mA)	2.4v	
V_{OL} (I_{OL} = 6mA)		0.4v
V_{IL}	-0.3v	0.8v
V_{IH}	2.0v	Vcc + 0.3v
Input hysteresis	0.20v typical	
Input transition time, t_R, t_F[11]	-	500ns
C_{IO}		10pF
I_{CC} standby current, no output load, all inputs at Vcc or GND (direction matching internal pulls, PWRDWN set)[12]	**50uA**	**150uA**
I_{CC} standby current, no output load, all inputs at Vcc or GND (direction matching internal pulls, PWRDWN cleared, Xin=14.7456MHz crystal)[12]	1.40mA	1.95mA
Xin frequency / crystal frequency[13]	10.0MHz	40.0MHz
Internal pull up / down current for signals with nominal 100K integrated resistors	pull up -20uA pull down 32uA	pull up -60uA pull down 150uA

8.4 ELECTRICAL CHARACTERISTICS, Vcc = 5.0v

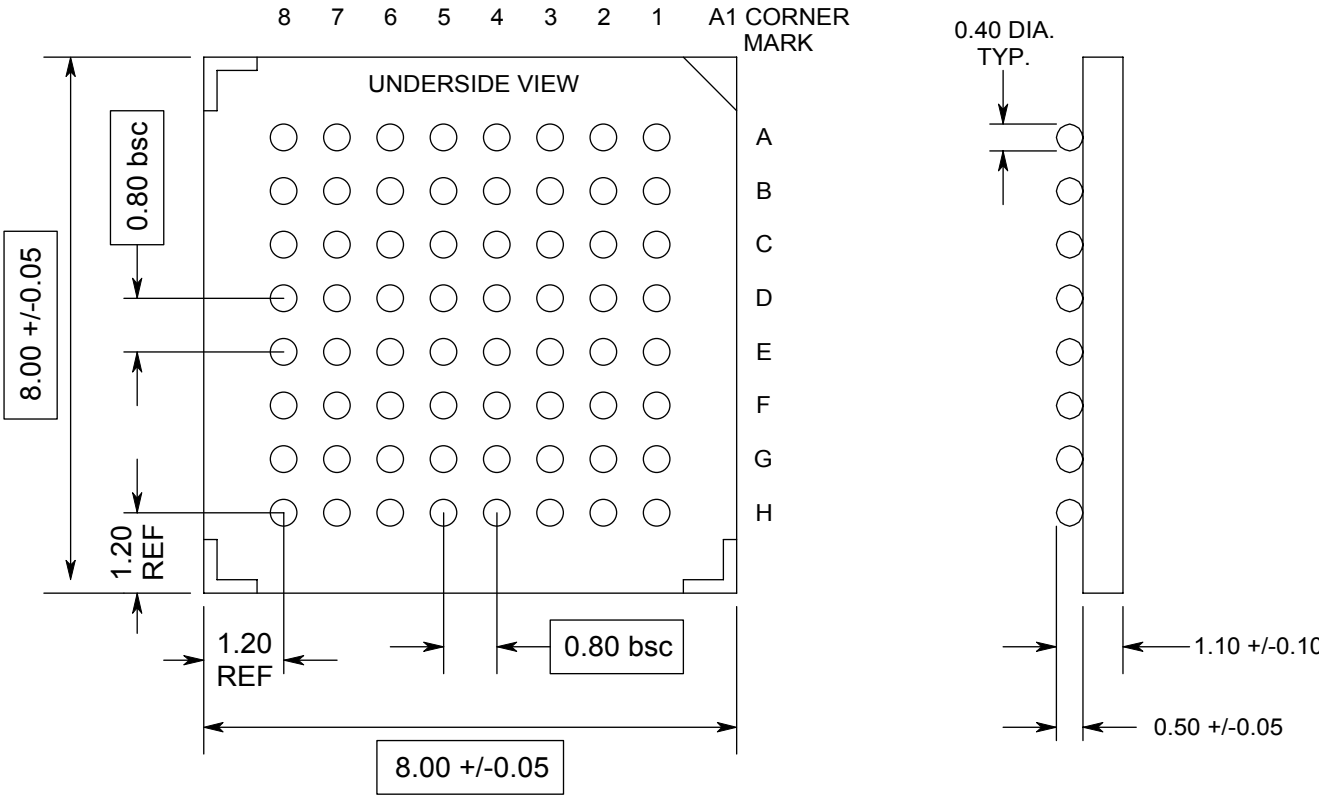
Parameter	Characteristic	
	Min	Max
V_{OH} (I_{OH} = -6mA)	3.8v	
V_{OL} (I_{OL} = 6mA)		0.33v
V_{IL}	-0.3v	0.8v
V_{IH}	2.0v	Vcc + 0.3v
Input hysteresis	0.60v typical	
Input transition time, t_R, t_F[11]	-	500ns
C_{IO}		10pF
I_{CC} standby current, no output load, all inputs at Vcc or GND (direction matching internal pulls, PWRDWN set)[12]	**140uA**	**300uA**
I_{CC} standby current, no output load, all inputs at Vcc or GND (direction matching internal pulls, PWRDWN cleared, Xin=14.7456MHz crystal)[12]	4.50mA	6.85mA
Xin frequency / crystal frequency[13]	10.0MHz	40.0MHz
Internal pull up / down current for signals with nominal 100K integrated resistors	pull up -55uA pull down 79uA	pull up -120uA pull down 330uA

[11] tR and tF are the slowest allowed input transition times to the VPU that will not cause the input buffer to oscillate during the transition region from 10-90% of Vcc (applies to non Schmitt trigger inputs only).
[12] All logic high inputs at greater than (Vcc x 0.9), all logic low inputs at less than 0.2v.
[13] The VPU is a fully static design. The minimum clock frequency reflects the time required by the internal synchronous FIFOs to commit data before the next host bus cycle can occur. On some applications, lower clock frequencies may work acceptably but you are strongly advised to conduct extensive cross-platform compatibility checks as Elan will not guarantee performance in such situations.

9.0 Package Mechanical Specifications

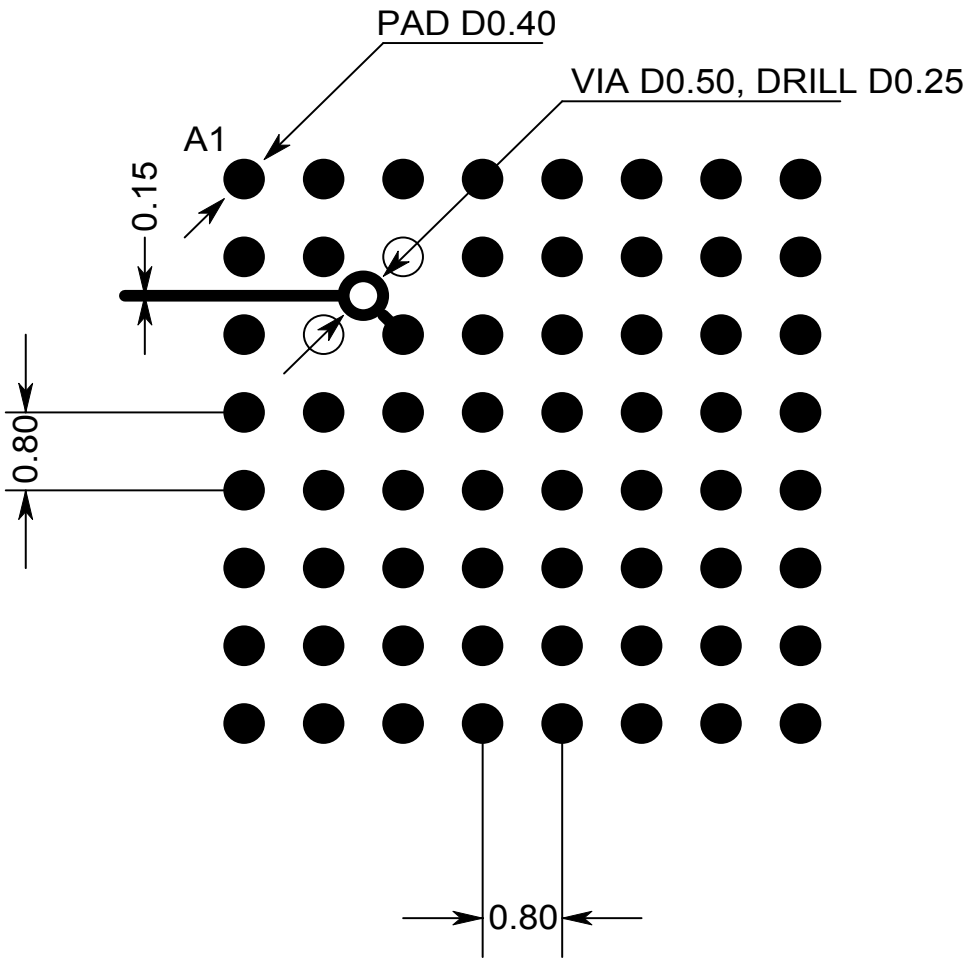
9.1 64-CBGA Outline Drawing

(all dimensions in mm)



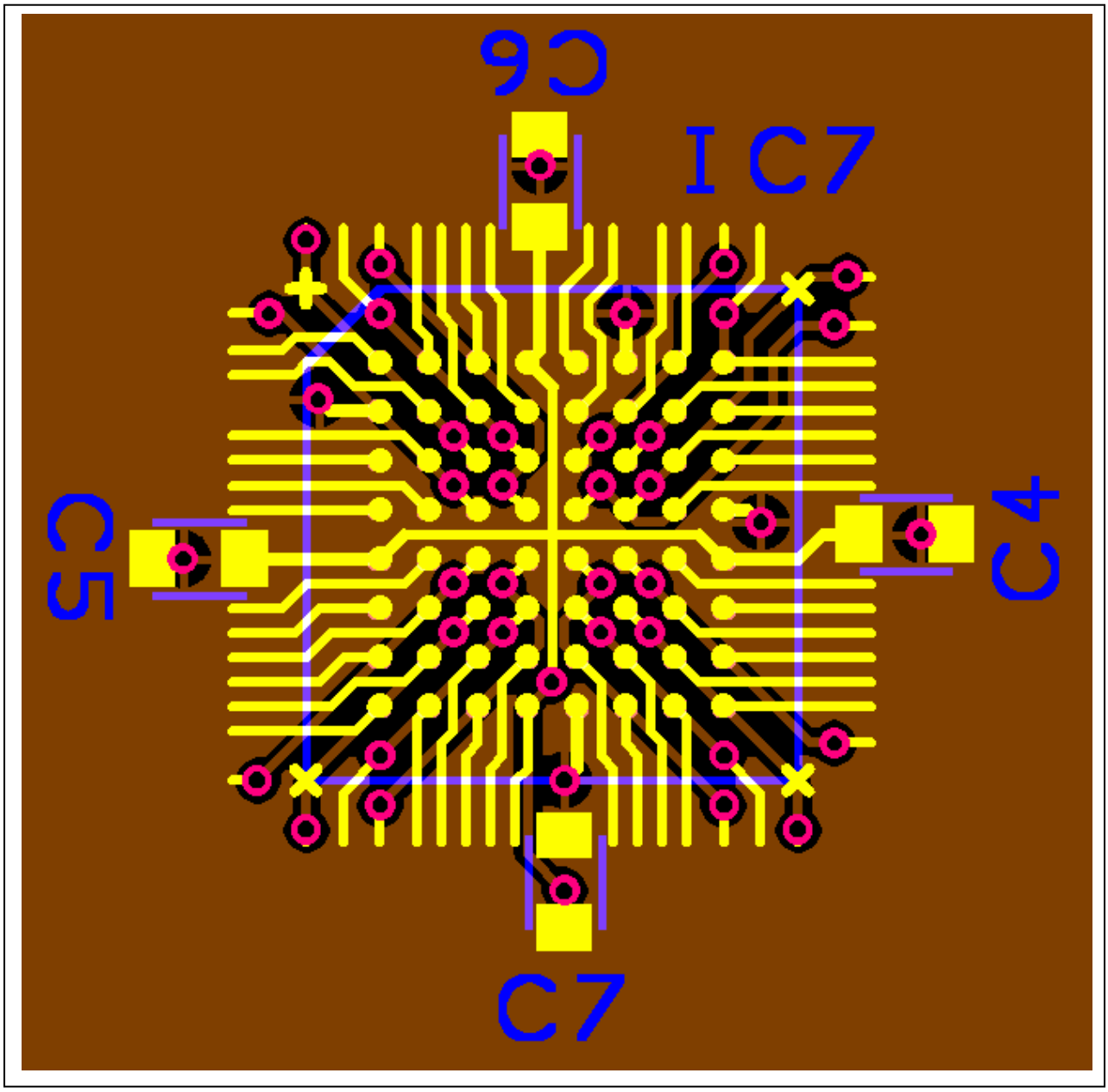
9.2 PCB Footprint Guidelines

(all dimensions in mm)



The above PCB footprint is for guidance only. Consult your PCB vendor for their geometry limitations and multi-layer capabilities. Drilled vias in pads are not recommended, but micro-vias in pads are acceptable.

The solder resist layer **must** fully cover any tracks that run between balls on upper layer especially when taking into account possible printing registration error relative to copper pattern. Areas between balls with no tracks **should** have solder resist to reduce chances of ball-ball bridging during placement / reflow.



2 Layer "Escape" routing suggestion (all balls routed to upper layer (yellow))

10.0 Reference Design Data

A reference design kit is available for the VPU, which includes a set of reference schematics, a working Serial Port card based on the VPU chip, Card Information Structure resource tools, a PCI to PCMCIA adapter with extender card to allow development work on a desktop, and source code and runtime for an EEPROM burning utility to in-circuit program the CIS data.

Please contact your Elan sales rep for further information