



“10c FPGA Charge Pump Enables 5V Interfacing”

Introduction:

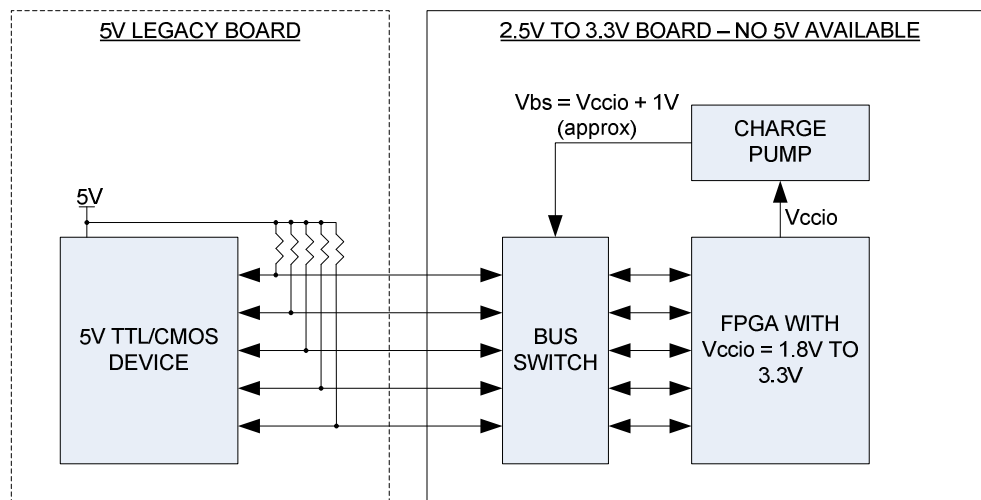
One of the major hassles in electronic design is interfacing modern FPGA's to legacy 5 volt based circuits. This becomes harder when the FPGA to operate at variable bank voltages.

Modern FPGA families like Xilinx™ Spartan™-3 or Virtex™-5 have absolute maximum input signal voltages of about 4V. Furthermore I/O cells usually have a protection diode to the local bank voltage which can be between 1.5V and 3.3V. If a voltage approximately 0.7V, or more, than the bank voltage is applied to the input then a large current can flow. This current can damage the I/O cell and even the main part of the device itself leading to costly repairs and system downtime.

Driving 5V CMOS inputs from the modern FPGA has been an issue for some time. Even when the legacy hardware has pull-ups to 5V on input or bidirectional signals the old technique of applying a simple pull-up to 5V to make CMOS levels no longer works. This is because either the FPGA protection diodes limit the pull-up or because the 3.3V (or less) I/O output drags the signal down to its drive level of 3.3V or less.

One of the best solutions to the problem is to use a bus switch, **without in-built protection diodes to Vcc**, between the FPGA I/Os and the 5V legacy logic. The bus switch passes signals in both directions without control and only acts to limit the drive from one side of the bus switch to the other. This factor allows the 5V pull-ups to be used on CMOS level inputs but also stops that voltage going back to the FPGA and causing damage.

The Interface Solution:



The circuit shown above uses a bus switch such that the FPGA is protected from over-voltages and the 5V CMOS level input level is attained in the 5V legacy system by pull up resistors. Note that if the legacy system uses TTL levels and the FPGA bank voltage is 2.5V or 3.3V then the pull-up resistors are not required. The pull-ups will ensure that 5V CMOS input levels are seen in the legacy system.

The choice of bus switch is determined by the following facts,

- (1) It must not have a protection diode to its Vcc rail.
- (2) It must pass sufficient voltage to show as a high TTL level without pull-ups on the legacy board and must meet the high input level requirement of the FPGA.
- (3) Must operate over a range of supply voltages.

Some bus switches have internal charge pump circuits to allow the passing of up to Vcc rail level voltages (called rail to rail bus switches.) Such bus switches tend to be less easy to source and generally more expensive than bus switches without an internal charge pump power supply. Subsequently here at Enterpoint we use the simpler, without internal charge pump, devices for our boards. This leads to a secondary problem of obtaining a high enough voltage input for the bus switch to function correctly.

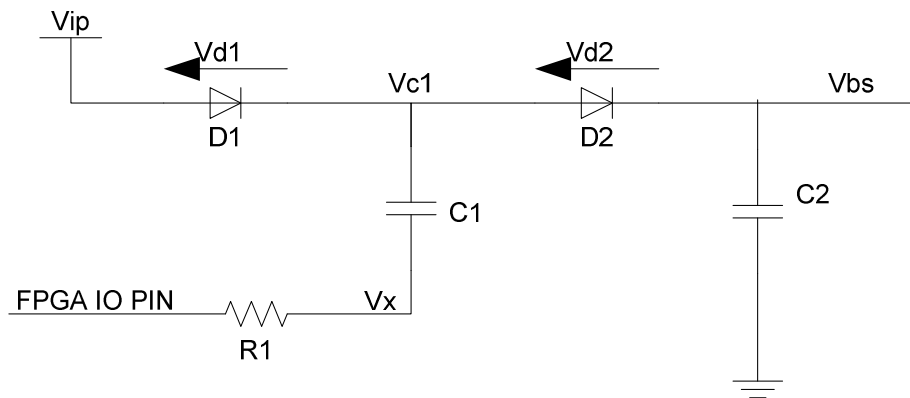
Typically the bus switch voltage input will need to be about 1V above the high level logic voltage to be passed through. Commonly bus switches will pass through 0.8 to 1.2 volts less than the supply rail they operate from. For example to pass through a 3.3V high signal the bus switch will need to operate at a 4.3V supply voltage. In boards with a 5V supply this can be generated with a simple resistor network. However in boards with only 3.3V, or less, supply this approach does not work. The answer is to have a charge pump circuit this time externally to the bus switch. Discrete Charge Pump devices can be bought from 50p (US \$1) or upwards in medium quantities which is a considerable amount for low cost, high volume, designs.

A related and further problem occurs in FPGA's where the bank voltage is varied by jumper or by another selection method to meet different IO standards, as happens in our FPGA development boards. Even if we feed the bus switch with 4.3V for 3.3V pass through we would still be set to high if the bank voltage is then set to 2.5V. To cope with this issue we needed a charge pump that tracks the bank voltage of the FPGA.

Enterpoint's solution is our own low cost Vccio (bank voltage) tracking charge pump circuit which we will now go on to explain.

The Charge Pump Solution:

The bus switch current requirements are very low so our charge pump does not need to supply much current. As previously mentioned we also needed a way of varying the output voltage. Commonly Enterpoint designs are FPGA boards and circuits and it was to FPGA technology we looked for the solution and came up with the following simple circuit.



The basic circuit consists of 2 diodes, 2 capacitors, 1 resistor and a single FPGA pin with supporting programmed logic. In this configuration this charge pump should be able to supply up to about 10mA depending on values and the function programmed into the FPGA. This circuit can be built for less than US 10c. A full cost breakdown is available in Appendix 1.

The circuit consists of a small capacitor C1 which acts as the pushing capacitor, C2 which is acting as the energy store, R1 is used to help protect the circuit from surging currents and two normal silicon diodes that exhibit normal behaviour i.e. a voltage of 0.7 volts is dropped across its self during operation.

The simplest FPGA I/O programming is basically a clock output with a frequency generally chosen to have a clock period of less than $4.4 \times R1 \times C1$ for greatest current output. The value of R1 is chosen to avoid overstressing the FPGA I/O pin. A value of 150R would be a good choice with a voltage rail input of 3.3V or 100R for 2.5V voltage rail input. For the purposes of our test circuit we can use $C1 = 1\mu\text{F}$ and $C2 = 4.7\mu\text{F}$. D1 and D2 can be 1N4148 diodes.

Circuit Operation and setup:

Keeping in mind the main focus of this TechTip which is interfacing to 5 volt systems using a normal bus switch and FPGA which has variable Vccio bank voltage we need to be able provide a voltage rail about 1V greater than the Vccio to power the bus switch such that the signal swing 0V to Vccio is accommodated and passed through.

Commonly we have voltage rails 3.3V, 2.5V and the core voltage of the FPGA available to help us meet our requirements. In SpartanTM-3 and VirtexTM-4 the core voltage is 1.2V whilst the newer VirtexTM-5 has a core voltage of 1V. As it happens 2.5V is a convenient voltage to use as the input rail V_{ip} and hence the bus switch supply voltage is given by the following equation, at light loading i.e. current drawn from C2 is negligible.

$$V_{bs} = V_{ip} + V_{ccio} - (V_{d1} + V_{d2})$$

If we set V_{input} to 2.5V we get the following bus switch voltages.

Vccio	Voltage Output	Nominal Bus switch pass through	Signal minimum pass through	Signal maximum pass through
3.3v	4.4v	3.4v	3.2v	3.6v
2.5v	3.6v	2.6v	2.4v	2.8v
1.8v	2.9v	1.9v	1.7v	2.1v

As can be seen our bus switch voltage tracks V_{ccio} at such a level to pass through V_{ccio} but does not pass a voltage as great as a diode drop above V_{ccio}.

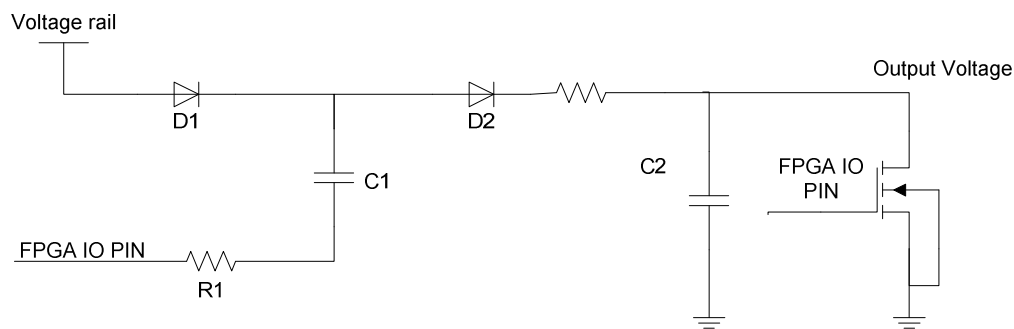
There are three conditions the FPGA charge pump can experience, abnormal conditions are 0 volt and 3.3 volt static conditions and a normal pulsing condition, and these will be explained in Appendix 2 in more detail.

Conclusions:

From the circuit descriptions one can see that for the purpose of bus switch powering the basic circuit works sufficiently to allow correct operation and hence no need for expensive circuitry, to power bus switches, to allow 3.3 volt systems to interface with a 5 volt system.

Additions possible to increase functionality:

The simplest addition possible is a shutdown feature, with the addition of a logic n-channel MOSFET sitting on the output and the FPGA as the control (with pull ups initiated,) the top plate of C2 can quickly be discharge through the MOSFET to ground effectively shutting the charge pump off.



Disclaimer:

All information provided in this TechTip should not be taken as a full treatment of the subject matter and the information provided is only that we have observed in lab testing on a breadboard and theoretical knowledge. If designed into a system please make sure you have understood how the circuit works and the practical limitations of the circuit.

APPENDIX 1:

The circuit consists of 3 0603 devices and 2 surface mount components for surface mount circuitry or 5 legged components for through hole circuitry, and of course a FPGA or CPLD.

The next couple of tables show individual pricing for both surface mount and through-hole components with reel/1000 quantity pricing.

Surface mount circuit BOM:

Component	Designator	Price 1000 off	Running cost
27Ω 0603 Resistor	R1	1.8c	1.8c
1μF 0603 Capacitor	C1	0.6c	2.4c
4.7μF 0603 Capacitor	C2	2.5c	4.9c
DL4448	D1	2.4c	7.3c
DL4448	D2	2.4c	9.7c

From the surface mount BOM we can see that the overall cost for this circuit is under 10c, when compared to a charge pump solution at thousand off pricing coming in at around US\$ 1, the above circuit is clearly 1/10 the price and about 1/5 the physical footprint area.

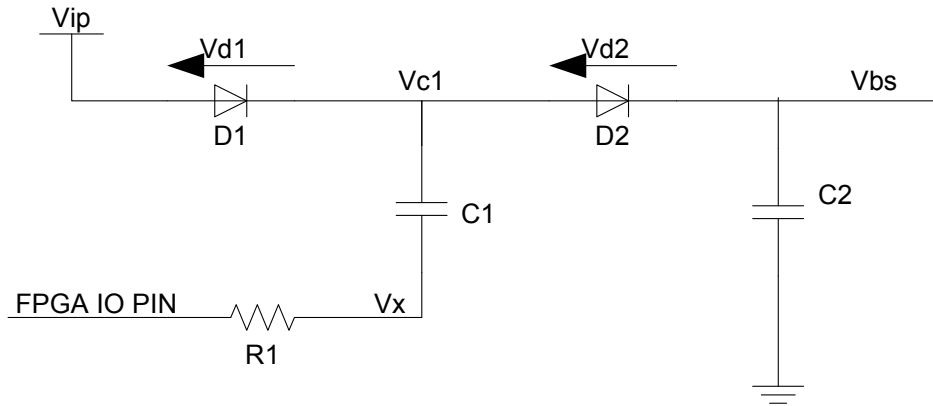
Through hole circuit BOM:

Component	Designator	Price 1000 off	Running cost
27Ω Resistor	R1	0.5c	0.5c
1μF Capacitor	C1	3c	3.5c
4.7μF Capacitor	C2	3c	6.5c
1N4148 diode	D1	0.8c	7.3c
1N4148 diode	D2	0.8c	8.1c

As with the surface mount circuit one can see that this circuit also costs fewer than 10c and would still take up less space due to the lack of the charge pump circuitry.

APPENDIX 2:

Conditions: $V_{ip} = 2.5v$, $V_{ccio} = 3.3v$ and $V_{d1} = V_{d2} = 0.7v$, low load effects discharge of C2 at a negligibly rate.



Abnormal operation: FPGA pin outputs a static 0 volts:

After power up V_x is fixed at a static 0 volts, V_{c1} would sit at a potential less than V_{ip} so the capacitor C1 will charge up through the diode D1 until $V_{c1} = V_{ip} - V_{d1}$ this would mean $V_{c1} = 2.5 - 0.7 = 1.8$ volts. V_{bs} is sitting at a lower potential than V_{c1} so the capacitor C2 will charge through D2 until $V_{bs} = V_{c1} - V_{d2}$ this would mean $V_{bs} = 1.8 - 0.7 = 1.1$ volts, this would mean the bus switch would pass at a maximum 0.1 volts through to the FPGA.

Abnormal operation: FPGA pin outputs a static 3.3 volts:

After power up V_x is fixed at a static 3.3 volts, some time thereafter the same condition as the 0 volt static condition would be seen but a negative voltage drop is observed across C1.

FPGA pin applies a pulse chain at power up:

On power up either situation above could arise weather the pulse chain starts with logic 0 or 1, the charge pumping starts when the FPGA outputs logic 0. When at 0 volts the first condition is observed and V_{c1} sits at 1.8 volts, as the FPGA pulses a logic 1 the lower plate of C1 charges up to a potential of 3.3 volts and the top plate follows this potential

rise as it has a positive voltage over itself, this means that $V_{c1} = V_{ip} - V_{d1} + V_{ccio} = 2.5 - 0.7 + 3.3 = 5.1$ volts.

V_{c1} is greater than V_{bs} so charge flows onto $C2$ and V_{bs} is now fully charged, $V_{bs} = V_{ip} - V_{d1} - V_{d2} + V_{ccio} = 2.5 - 0.7 - 0.7 + 3.3 = 4.4$ Volts. As the FPGA pin returns back to 0 volts the charge on the output capacitor can not flow back onto $C1$ due to the diode so it sits on the output capacitor $C2$ until it is discharged through an effective resistance of a circuit sitting on the new rail, this 4.4 volts is sufficient to operate a bus switch to interface at the V_{ccio} voltage.

The circuitry attached to the output of the charge pump will have an effective resistance and the charge on top of $C2$ will discharge through the effective resistance in the normal RC behaviour until the static condition in the first condition is reached with 1.1 volts sitting on the output capacitors top plate. On the next pulse more charge is pushed onto the output capacitor to effectively top up the reserve, if the pulsing to top up the output charge is greater than the rate at which it's discharged a constant 4.4 volts with ripple is seen on the output.