

Isolating buffers using special fixed logic LOW I/O levels

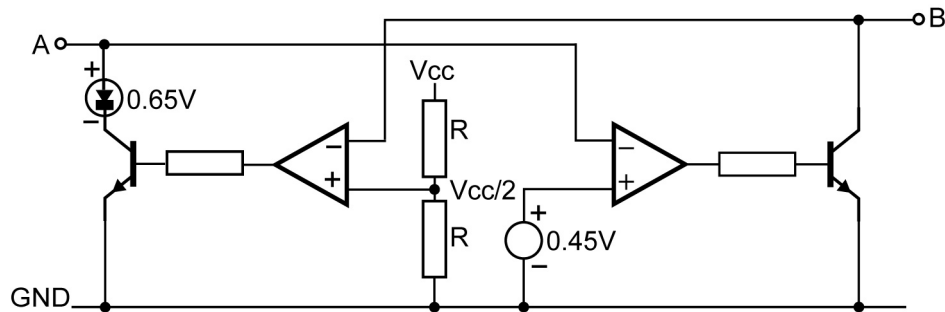


Figure 1: A buffer using special input and output levels on one side (shown here at A)

This class of buffer solves the latching problem explained in [TR002 \(Buffering I²C signals\)](#) by the use of special logic input and output levels on one of its I/Os, side A in Fig. 1. The special levels provide the means of discrimination between these special logic levels, as used by the circuitry inside this buffer at A, and the normal logic levels used by other I²C devices externally connected to the bus at A.

For simplicity, all bus voltage levels quoted in this note are those applying to a Fast-mode or Fm+ system with mandatory hysteresis and with a bus voltage above 2 V. For values applying to other I²C-bus variants refer to [TR004 \(Noise margin in I²C systems\)](#)

In this example side B uses the same I²C switching levels as specified for I²C-bus driving devices. The switching threshold for bus signals at B, below which the side A will be driven LOW, is nominally set at $V_{CC}/2$. The switching threshold for signals applied to B is set by the reference used by the comparator on the left side in Fig. 1. I²C device compliance requires their switching levels to be in the range 35 – 65% V_{CC} , so typically this buffer's comparator level will also be set around 50% V_{CC} as shown. This comparator may also have hysteresis (not shown).

When output B is driven LOW by its internal open collector driver transistor it is driven all the way down to ground. B then even meets the normal I²C logic LOW level required of I²C-bus driver I/O stages (i.e. less than 0.4 V) and can be applied in the same ways as any I²C driver device.

Buffers however are not the originators of the I²C-bus signals, they only reproduce them, so when considered in a system design they are only required to drive the bus lines to the mandatory I²C-bus logic levels. That LOW requirement is 0.25 times the bus supply voltage so even the special output level used by side A can be designed to conform with this I²C-bus requirement. The side A in this example uses the special logic levels.

When B goes LOW (below $V_{CC}/2$) the comparator on the left drives the transistor that will pull side A LOW. The transistor collector pulls all the way down to ground but a series diode has been added and this prevents bus A from being pulled lower than the voltage drop of that diode. In this example, for some specified sink current, the diode drop is set at 0.65 V. When sinking that specified current, the bus at A cannot be pulled below 0.65 V by the action of this buffer.

For bus supplies above 2.2 V this 0.65 V is still a level that must be recognised as a logic LOW level by any I²C compliant devices connected to the bus at A. Their V_{IL} requirement is $0.3 \times 2.2 = 0.66$ V. Further, that is only the 'worst case' requirement. An I²C device with at least 2.2 V supply will typically have its logic switching level (for falling bus edges) above $0.45V_{CC}$, or 0.99 V, so the 0.65 V driven by this buffer still has a typical safety margin of $(0.99 \text{ V} - 0.65 \text{ V}) = 0.34 \text{ V}$ even when driving very low voltage devices. I²C devices with higher supply voltages will have even greater safety (noise) margins when driven LOW by output A.

The comparator on the right hand side, that pulls bus B LOW, has a 0.45 V reference. That means that bus B will only be pulled LOW when the voltage on bus A is lower than 0.45 V. Because A cannot be driven below 0.65 V by this buffer's internal circuit action, an external logic LOW applied at B will not activate the internal open collector driver of B – so that LOW signal applied at B will not be internally fed back to B – therefore this buffer does not latch LOW.

I²C- bus driver devices are required to pull their output below 0.4 V, so the system designer has only to arrange that any driving devices connect directly to bus A and not via any series resistors or buffers that would increase that bus LOW level. Then their LOW will be correctly copied to bus B.

This buffer can therefore correctly pass external I²C- bus LOW signals applied at either side while not allowing a signal applied at B to be internally copied back to B. Side A is not fully compliant with the I²C- bus specifications because it will not pass worst-case (0.25V_{CC}) bus LOW signals and also will not recognise a LOW signal coming from the A side of another buffer using similar special levels. It's special logic side remains I²C- bus 'compatible' but, because it fails on this one point (one input switching threshold has to be set too low), it cannot be fully I²C- bus 'compliant'.

A consequence of its special A levels is that it is not allowed to make systems that would have more than one of these special A inputs connected to any one section of an I²C- bus. It is allowed to connect any number of B sides to any one bus section, and it is allowed to connect these buffers in 'series' provided each B side has only one A side connected to it.

If two LOW signals are applied at both A and B, pulling them both below 0.4 V, then the internal drive at B does activate the internal pull-down drive at A (to 0.65 V) but this drive will not be apparent while A is below 0.4 V. If the drive at A is then released this A side will only rise to 0.65 V and be held at that level until input B is removed.

It is important to note that there is no transition to a HIGH level at A when the external LOW at A is removed while B is LOW. This type of buffer therefore

meets the I²C requirements for correct handling of bus contention and clock stretching. For more details refer to [TR003 \(False logic glitches\)](#).

The biggest compromise in a buffer of this design is in selecting the level that is recognised by the special input as a LOW. In this example we use 0.45 V as the selected level. That guarantees that all normal I²C devices, that must pull the bus below 0.4 V, can correctly drive this buffer. The compromise is that the guaranteed difference between the 0.4 V bus level and the 0.45 V requirement of the buffer is just 0.05 V.

Typically I²C devices will drive the I²C-bus to below 0.2 V and then the typical safety margin increases to (0.45 V – 0.2 V) = 0.25 V. That is a workable safety margin provided the wiring of the bus at A is kept short and the noise levels of its environment are not extreme. It could be argued that the input threshold should be set somewhat higher, because the output bus drive is only required to be below 0.25V_{CC}. For a nominal supply of 3.3 V, say 3.0 V min, even an output of 0.75 V meets that I²C-bus LOW requirement and could allow an input threshold set at, say, 0.65 V to provide an extra 0.2 V margin.

Although TTL logic levels are not I²C- bus compliant levels they are the levels specified for derivative buses such as SMBus and are the levels commonly available on microcontroller ports that have not been specifically designed for I²C- bus applications. TTL specifies a fixed maximum bus LOW of 0.8 V, very similar to a 3.3V I²C- bus LOW requirement at 0.25x 3.3 = 0.83 V. TTL driver devices mostly have the same V_{OL} requirement as I²C devices, 0.4 V.

The safety margin between the LOW level on the I²C- bus and a higher level that will still be recognised by the buffer as a logic LOW is called the noise margin. In this case the noise margin being compromised is noise margin LOW - the safety margin during a bus LOW at A. When the bus at A is HIGH, i.e. at V_{CC}, the noise margin HIGH for this buffer is actually improved. A normal I²C part may recognise any level on a falling edge below its switching threshold of 0.65V_{CC} as being LOW, but the bus at A must be driven (for example by noise) to below 0.45 V before any false logic signal would appear on bus B. The typical noise margin HIGH for input A is (V_{CC} – 0.45 V) while for normal I²C

parts it is $(V_{CC} - 0.45V_{CC}) = 0.55V_{CC}$. For any bus voltage above 0.82 V the buffer input A has a higher typical noise margin HIGH. The worst case noise margin HIGH for an I²C system, as defined in the specifications, is $0.2V_{CC}$. The bus "HIGH" is required to be always above $0.85V_{CC}$ and the I²C device's

switching threshold, for falling bus edges, must be below $0.65V_{CC}$. Using that worst case bus condition, the noise margin "HIGH" for input A would be $(0.85V_{CC} - 0.45 V)$.

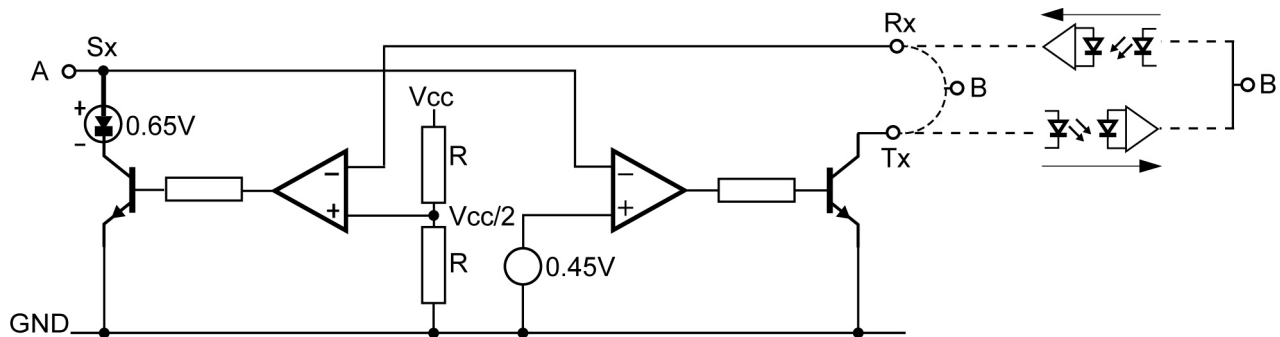


Figure 2: Alternative configuration at B splits bi-directional I²C-bus A into two uni-directional signals

Fig.2 shows that once the special logic levels on one Special I/O (marked Sx) have solved the latching problem it is possible to extend this principle and to separate the two conventional logic signals that we previously showed as simply linked at B in Fig. 1.

Removing the link between the open collector drive transistor output (marked Tx for transmit) and the input to the comparator (marked Rx for receive) allows these two uni-directional logic signals to be made available on two separate pins of the buffer.

It is, of course, allowed to simply join those two pins by a link to form a perfectly standard I²C again as shown by the alternative link between the input and output signals at B. But it is also now possible to transmit those two separate uni-directional component signals over any conventional uni-directional logic signal paths before making that link to re-form a conventional bi-directional I²C- bus B.

Fig. 2 also shows that it is possible to pass those uni-directional signals through opto-couplers to achieve a safety isolation barrier between the two I²C- buses A and B. The only requirement at the point where the two uni-directional signals are joined to form an I²C is that its interface meets the I²C requirements. The driver of the I²C- bus at B (supplying the output signal coming from the buffer) must have an open-collector or open-drain configuration and the receiver from the

I²C- bus at B should use a high input impedance buffer. Both should use logic levels that meet the I²C- bus requirements.

There is one very important difference in operation between this configuration, with special levels used to solve the latching problem, and using time-based methods to block one of the two uni-directional signals. The time-based method always results in the generation of a spurious logic 'HIGH' level pulse on the bus line, as explained in [TR003 \(False logic glitches\)](#), buffers using special logic levels do not.

Important characteristics of this type of buffer:

It provides isolation between the loadings connected at each I/O. That is, the loading of the bus on its Tx/Rx side has no influence on the input impedance of the Sx side, and the I²C- bus loading connected at Sx does not influence the input impedance at Rx. The input impedance at the I/Os, for all voltages from 0 V to the allowed limit (e.g. abs max 18 V for P82B96) remains essentially infinite for all the buffers in this category. (Note: The Sx side of PCA9600 is one exception. The Sx input of PCA9600 sources the current used to temperature stabilise the logic voltages at its Sx. This current is typically <300 μA, and is independent of anything connected to Tx/Rx).

This is the only buffer class that has the capability to split a bi-directional I²C- bus into two uni-directional signals or to re-combine such components to form an I²C-bus again without generating spurious glitches.

Having logic switching levels on the Rx input side related to the V_{CC}, and with parts rated to 18 V (abs. max.) on their V_{CC}, means this category can provide the largest available I²C- bus noise margins on this interface. It provides the highest immunity to induced noise for the bus connected to this Tx/ Rx side. If required, it is even possible to expand the logic voltages of a bus driven by Tx/ Rx to any practical value, or to use separate uni-directional buses operating at any practical logic voltage level, say 100 V, to achieve extreme noise margins. Further, any noise on such high level buses is completely isolated from the bus connected at the Sx side.

It is possible to add an 'Enable' input to control when the buffer will be active. That allows it to be applied in hot insertion applications requiring the interface to be

disabled during the initial connection and until conditions are suitable for allowing bus signals to pass. It can allow these buffers to perform some bus switch/ multiplexer functions.

It's easy to recognise this type of buffer by the characteristic 'step' it produces in the rising edge of the waveform on the side that uses special levels. If this pin has been LOW then its corresponding output has been LOW. When this pin is released it releases the output but the input will be clamped at the 'special' LOW level until the output actually reaches its HIGH level and releases this input. During the delay times involved the input will have a characteristic 'step' shape as shown in Fig. 3.

During the short step time this input remains at a logic LOW level so this 'step' does not contribute to the subsequent rise time of the bus because that is defined between 0.3 and 0.7V_{bus}. Instead this step causes a signal propagation delay.

[In unusual applications, where Tx and Rx are not linked, this characteristic step may not appear]

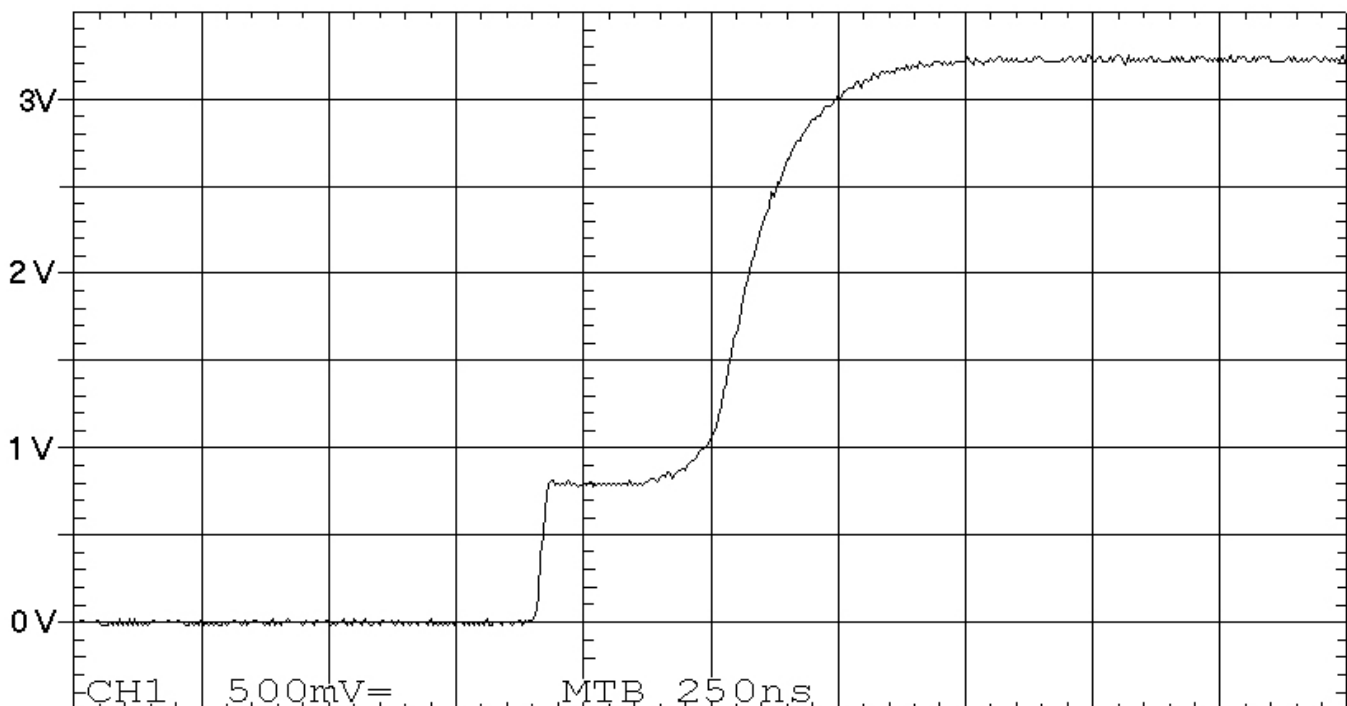


Figure 3:Characteristic waveform at an I/O pin using special levels that prevent latching.

Where this part is most useful:

Whenever it is required to split the I²C- bus signals into its two uni-directional components so that those signals can be carried by other conventional logic signalling systems, across opto (or other) isolating devices, or even over radio links.

Where the highest logic switching levels are required to achieve highest noise margins.

For driving I²C signals on long cables (e.g. Cat5 twisted pair cables as used for Ethernet wiring).

Performing bus logic level shifting where it is important to retain the noise margins of the higher voltage bus.

Designing an I²C system? Visit www.bus-buffer.com for more information or to contact us.

Short summary of features	Unit	Sx side of P82B96	Tx/Rx side of P82B96	Sx side of PCA9600	Tx/Rx side of PCA9600
Input switching level characteristics					
V _{IL} Highest level that guarantees O/P LOW (V)	V	0.6 ¹	0.42V _{cc}	0.425 ²	0.4V _{cc}
V _{IH} Lowest level that guarantees O/P HIGH (V)	V	0.7 ¹	0.58V _{cc}	0.58 ²	0.55V _{cc}
Compatible with 0.4 V I ² C/TTL driver logic levels	-	yes	yes	yes	yes
Meets Fast-mode LOW threshold > 0.35V _{cc}	-	no	yes	no	yes
Compatible Fast-mode 0.25V _{cc} max bus LOW	-	V _{bus} <2.4V	yes	no	yes
HIGH level meets I ² C-bus noise margin specs	-	yes	yes	yes	yes
Output LOW drive level characteristics					
V _{OL} when sinking 3 mA (max)	V	0.88 ¹	0.4	0.74 ²	0.4
Max < Fast-mode noise margin 0.25V _{cc} spec	-	V _{bus} >3.5V	yes	V _{bus} >3V	yes
Compatible with I ² C-bus V _{IL} max level (0.3V _{cc})	-	V _{bus} >2.9V	yes	yes	yes
Compatible with TTL bus max LOW level (0.8 V)	-	I _{SINK} < 0.2mA ¹	yes	yes	yes
Output driver sink capability					
Minimum Standard 3 mA static sink capability	-	yes	yes	yes	yes
Additional, enhanced, static sink capability	-				
Enhanced 30 mA (Fm+) static sink capability	-	no	yes	no	yes
This output suitable for driving long cables	-	no	yes	no	yes
Other I/O characteristics					
Allowed to parallel several of these I/Os	-	no	yes	no	yes
I/O sources current (>10 uA allowed by I ² C spec)	-	no	no	yes	no
Drive has load component due other I/O's load	-	no	no	no	no
Logic level shifting capability on this I/O	-	yes	yes	yes	yes
Connected bus voltage may exceed V _{cc}	-	yes	yes	yes	yes
Noise during LOW on I/P is transferred to O/P	-	no	no	no	no
I2C to/from unidirectional components					
Split transmit and receive	-	no	yes	no	yes
Input (Rx) logic levels are I ² C compliant	-		yes		yes
Output (Tx) drive is I ² C compliant	-		yes		yes
Bus speeds					
Suggested maximum bus speed	Hz	400 k	400 k	1 M	1 M
Approx. propagation delay, this I/P to O/P	ns	70	200	50	70
Can be useful to at least this bus speed	Hz	700 k	700 k	1 M	1 M

Notes:

1. At +25 °C
2. Over rated temperature range