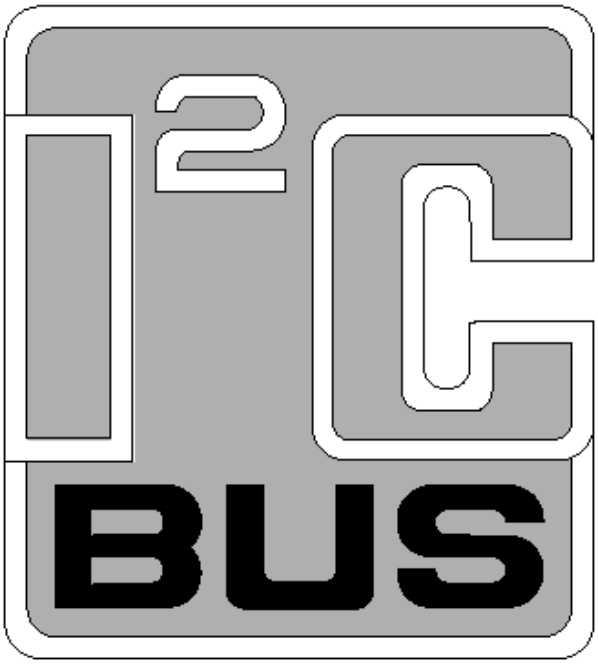
APPLICATION NOTE



AN10216-01
I²C MANUAL

Abstract – **The I²C Manual provides a broad overview of the various serial buses, why the I²C bus should be considered, technical detail of the I²C bus and how it works, previous limitations/solutions, comparison to the SMBus, Intelligent Platform Management Interface implementations, review of the different I²C devices that are available and patent/royalty information. The I²C Manual was presented during the 3 hour TecForum at DesignCon 2003 in San Jose, CA on 27 January 2003.**

Jean-Marc Irazabal – **I²C Technical Marketing Manager**
Steve Blozis – **I²C International Product Manager**

Specialty Logic Product Line
Logic Product Group

Philips Semiconductors **March 24, 2003**

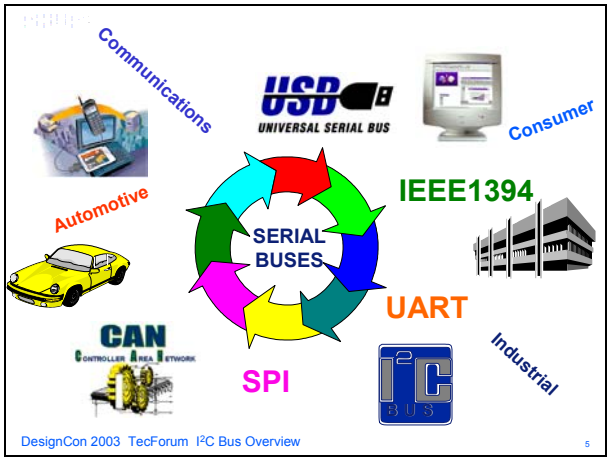
TABLE OF CONTENTS

OVERVIEW

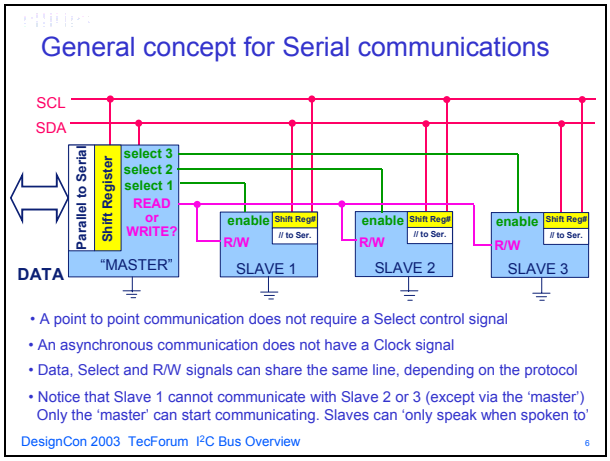
Description

Philips Semiconductors developed the I²C bus over 20 years ago and has an extensive collection of specific use and general purpose devices. This application note was developed from the 3 hour long I²C Overview TecForum presentation at DesignCon 2003 in San Jose, CA on 27 January 2003 and provides a broad overview of how the I²C bus compares to other serial buses, how the I²C bus works, ways to overcome previous limitations, new uses of I²C such as in the Intelligent Platform Management Interface, overview of the various different categories of I²C devices and patent/royalty information. Full size Slides are posted as a PDF file on the Philips Logic I²C collateral web site as **DesignCon 2003 TecForum I²C Bus Overview** PDF file. Place holder and title slides have been removed from this application note and some slides with all text have been incorporated into the application note speaker notes.

Serial Bus Overview



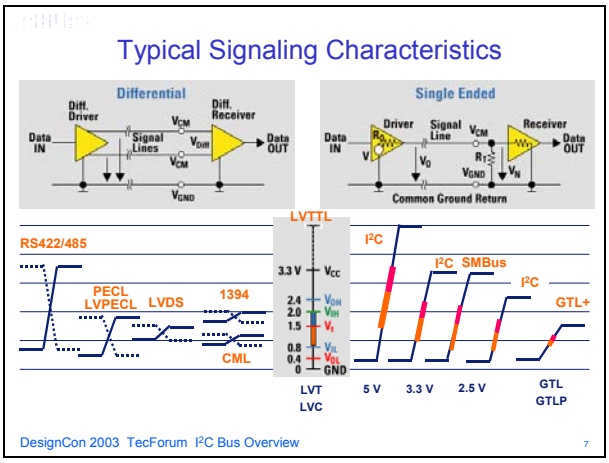
Slide 5



Slide 6

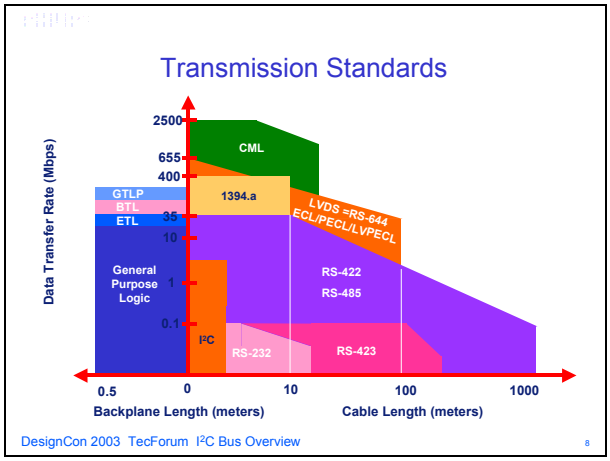
Buses come in two forms, serial and parallel. The data and/or addresses can be sent over 1 wire, bit after bit, or over 8 or 32 wires at once. Always there has to be some way to share the common wiring, some rules, and some synchronization. Slide 6 shows a serial data bus with three shared signal lines, for bit timing, data, and R/W. The selection of communicating partners is made with one separate wire for each chip. As the number of chips grows, so do the selection wires. The next stage is to use multiplexing of the selection wires and call them an address bus.

If there are 8 address wires we can select any one of 256 devices by using a 'one of 256' decoder IC. In a parallel bus system there could be 8 or 16 (or more) data wires. Taken to the next step, we can share the function of the wires between addresses and data but it starts to take quite a bit of hardware and worst is, we still have lots of wires. We can take a different approach and try to eliminate all except the data wiring itself. Then we need to multiplex the data, the selection (address), and the direction info - read/write. We need to develop relatively complex rules for that, but we save on those wires. This presentation covers buses that use only one or two data lines so that they are still attractive for sending data over reasonable distances - at least a few meters, but perhaps even km.



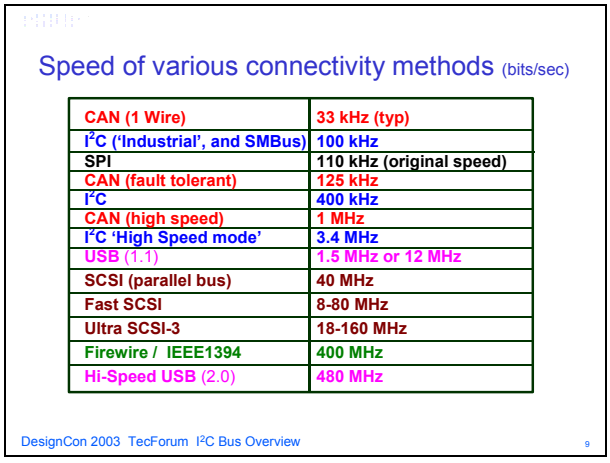
Slide 7

Devices can communicate differentially or single ended with various signal characteristics as shown in Slide 7.



Slide 8

The various data transmission rates vs length or cable or backplane length of the different transmission standards are shown in Slide 8.

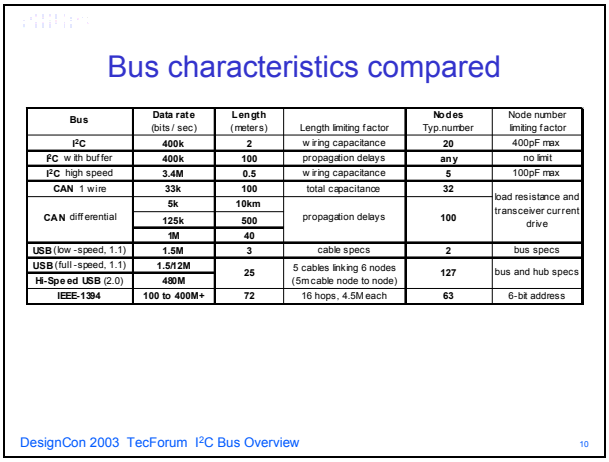


Slide 9

Increasing fast serial transmission specifications are shown in Slide 9. Proper treatment of the 480 MHz version of USB - trying to beat the emerging 400 MHz 1394a spec - that is looking to an improved 'b' spec - - etc is beyond the scope of this presentation. Philips is developing leading-edge components to support both USB and 1394 buses.

Today the path forward in USB is built on "OTG" (On The Go) applications but the costs and complexity of this are probably beyond the limits of many customers. If designers are identified as designing for large international markets then please contact the USB group for additional support, particularly of Host and OTG solutions. Apologies for inclusion of the parallel SCSI bus. It is intended for comparison purposes and also because it may be used within the PC software as a general data path that USB drivers can use.

Terminology for USB: The use of older terms such as the spec version 1.1 and 2.0 is now discouraged. There is just "USB" (meaning the original 12 Mbits/sec and 1.5 Mbits/sec speeds of USB version 1.1) and Hi-Speed USB meaning the faster 480 Mbits/sec option included in spec version 2.0. Parts conforming to or capable of the 480 Mbits/sec are certified as Hi-Speed USB and will then feature the logo with the red stripe "Hi-Speed" fitted above the standard USB logo. The reason to avoid use of the new spec version 2.0 as a generic name is that this version includes all the older versions and speeds as well as the new Hi-Speed specs. So USB 2.0 compliance does NOT imply Hi-Speed (480 Mbits/sec). ICs can be compliant with USB 2.0 specifications yet only be capable of the older 'full speed' or 12 Mbits/sec.



Bus	Data rate (bits / sec)	Length (meters)	Length limiting factor	Nodes Typ.number	Node number limiting factor
I²C	400k	2	w iring capacitance	20	400pF max
I²C w ith buffer	400k	100	propagation delays	any	no limit
I²C high speed	3.4M	0.5	w iring capacitance	5	100pF max
CAN 1 wire	33k	100	total capacitance	32	load resistance and transceiver current drive
CAN differential	5k	10km	propagation delays	100	
	125k	500			
	1M	40			
USB (low -speed, 1.1)	1.5M	3	cable specs	2	bus specs
USB (full -speed, 1.1)	1.5/12M	25	5 cables linking 6 nodes (5m cable node to node)	127	bus and hub specs
Hi-Speed USB (2.0)	480M				
IEEE-1394	100 to 400M+	72	16 hops, 4.5M each	63	6-bit address

Slide 10

In Slide 10 we look at three important characteristics:

- Speed, or data rate
- Number of devices allowed to be connected (to share the bus wires)
- Total length of the wiring

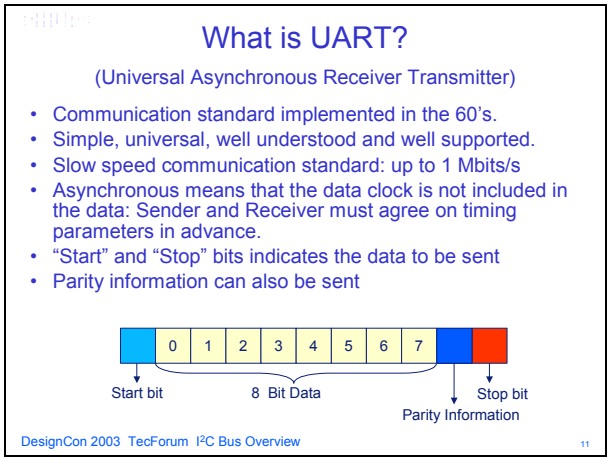
Numbers are supposed to be realistic estimates but are based on meeting bus specifications. But rules are made to be broken! When buffered, I²C can be limited by wiring propagation delays but it is still possible to run much longer distances by using slower clock rates and maybe also compromising the bus rise and fall-time specifications on the buffered bus because it is not bound to conform to I²C specifications.

The figure in Slide 10 limiting I²C range by propagation delays is conservative and allows for published response delays in chips like older E² memories. Measured chip responses are typically < 700 ns and that allows for long cable delays and/or

operation well above 100 kHz with the P82B96. The theoretical round-trip delay on 100 m of cable is only approx 1 µs and the maximum allowed delay, assuming zero delays in ICs, is about 3 µs at 100 kHz. The figures for CAN are not quite as conservative; they are the 'often quoted values'. The round trip delay in 10 km cable is about 0.1 ms while 5 kbps implies 0.2 ms nominal bit time, and a need to sample during the second half of the bit time. That is under the user's control, but needs attention.

USB 2 and IEEE-1394 are still 'emerging standards'. Figures quoted may not be practical; they are just based on the specification restrictions.

UART Overview



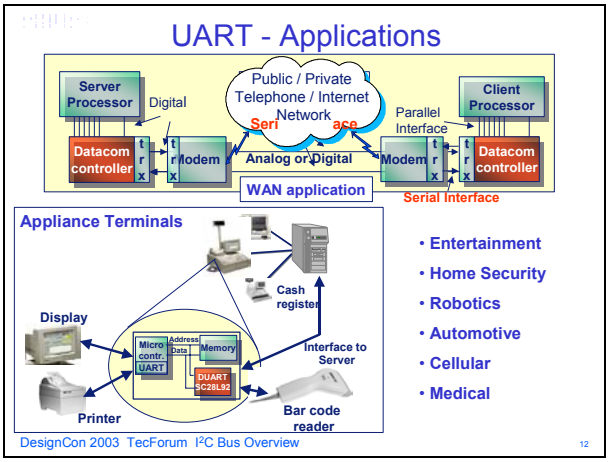
Slide 11

UARTs (Universal Asynchronous Receiver Transmitter) are serial chips on your PC motherboard (or on an internal modem card). The UART function may also be done on a chip that does other things as well. On older computers like many 486's, the chips were on the disk IO controller card. Still older computers have dedicated serial boards.

The UARTs purpose is to convert bytes from the PC's parallel bus to a serial bit-stream. The cable going out of the serial port is serial and has only one wire for each direction of flow. The serial port sends out a stream of bits, one bit at a time. Conversely, the bit stream that enters the serial port via the external cable is converted to parallel bytes that the computer can understand. UARTs deal with data in byte-sized pieces, which is conveniently also the size of ASCII characters.

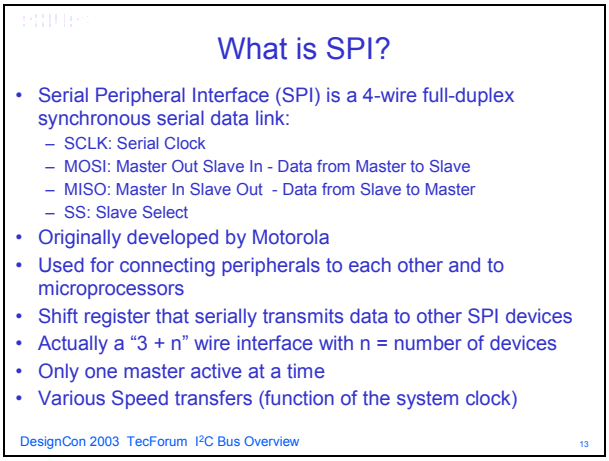
Say you have a terminal hooked up to your PC. When you type a character, the terminal gives that character to its transmitter (also a UART). The transmitter sends that byte out onto the serial line, one bit at a time, at a specific rate. On the PC end, the receiving UART takes all the bits and rebuilds the (parallel) byte and puts it in a buffer.

Along with converting between serial and parallel, the UART does some other things as a byproduct (side effect) of its primary task. The voltage used to represent bits is also converted (changed). Extra bits (called start and stop bits) are added to each byte before it is transmitted. Also, while the flow rate (in bytes/s) on the parallel bus speed inside the computer is very high, the flow rate out the UART on the serial port side of it is much lower. The UART has a fixed set of rates (speeds) that it can use at its serial port interface.



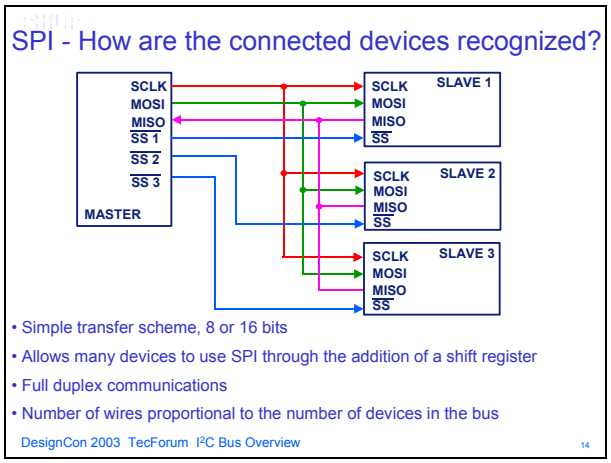
Slide 12

SPI Overview



Slide 13

The Serial Peripheral Interface (SPI) circuit is a synchronous serial data link that is standard across many Motorola microprocessors and other peripheral chips. It provides support for a high bandwidth (1 mega baud) network connection amongst CPUs and other devices supporting the SPI.



Slide 14

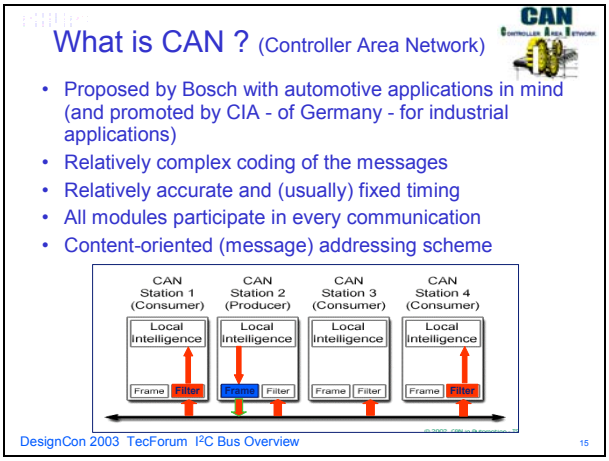
The SPI is essentially a "three-wire plus slave selects" serial bus for eight or sixteen bit data transfer applications. The three wires carry information between devices connected to the bus. Each device on the bus acts simultaneously as a transmitter and receiver. Two of the three lines transfer data (one line for each direction) and the third is a serial clock. Some devices may be only transmitters while others only receivers. Generally, a device that transmits usually possesses the capability to receive data also. An SPI display is an example of a receive-only device while EEPROM is a receiver and transmit device.

The devices connected to the SPI bus may be classified as Master or Slave devices. A master device initiates an information transfer on the bus and generates clock and control signals. A slave device is controlled by the master through a slave select (chip enable) line and is active only when selected. Generally, a dedicated select line is required for each slave device. The same device can possess the functionality of a master and a slave but at any point of time, only one master can control the bus in a multi-master mode configuration. Any slave device that is not selected must release (make it high impedance) the slave output line.

The SPI bus employs a simple shift register data transfer scheme: Data is clocked out of and into the active devices in a first-in, first-out fashion. It is in this manner that SPI devices transmit and receive in full duplex mode.

All lines on the SPI bus are unidirectional: The signal on the clock line (SCLK) is generated by the master and is primarily used to synchronize data transfer. The master-out, slave-in (MOSI) line carries data from the master to the slave and the master-in, slave-out (MISO) line carries data from the slave to the master. Each slave device is selected by the master via individual select lines. Information on the SPI bus can be transferred at a rate of near zero bits per second to 1 Mbits per second. Data transfer is usually performed in eight/sixteen bit blocks. All data transfer is synchronized by the serial clock (SCLK). One bit of data is transferred for each clock cycle. Four clock modes are defined for the SPI bus by the value of the clock polarity and the clock phase bits. The clock polarity determines the level of the clock idle state and the clock phase determines which clock edge places new data on the bus. Any hardware device capable of operation in more than one mode will have some method of selecting the value of these bits.

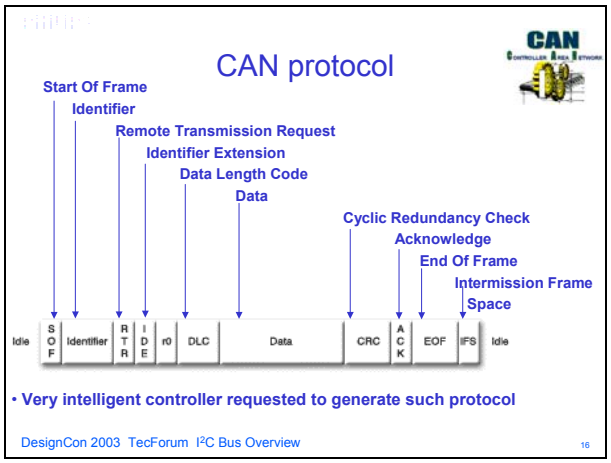
CAN Overview



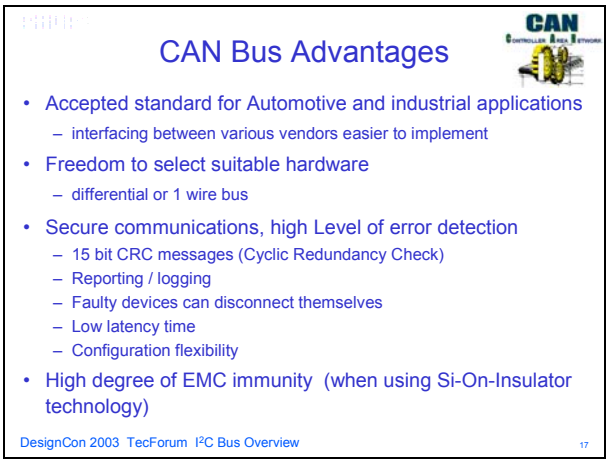
Slide 15

CAN objective is to achieve reliable communications in relatively critical control system applications e.g. engine management or anti-lock brakes. There are several aspects to reliability - availability of the bus when important data needs to be sent, the possibility of bits in a message being corrupted by noise etc., and electrical/mechanical failure modes in the wiring.

At least a ceramic resonator and possibly a quartz crystal are needed to generate the accurate timing needed. The clock and data are combined and 6 'high' bits in succession is interpreted as a bus error. So the clock and bit timings are important. All connected modules must use the same timings. All modules are looking for any error in the data at any point on the wiring and will report that error so the message can be re-sent etc.



Slide 16



Slide 17

Like I²C, the CAN bus wires are pulled by resistors to their resting state called a 'recessive' state. When a transceiver drives the bus it forces a voltage called the 'dominant' state. The identifier indicates the meaning of the data, not the intended recipient. So all nodes receive and 'filter' this identifier and can decide whether to act on the data or not. So the bus is using 'multicast' - many modules can act on the message, and all modules are checking the message for transmission errors. Arbitration is 'bit wise' like I²C - the module forcing a '1' beats a module trying for a '0' and the loser withdraws to try again later.

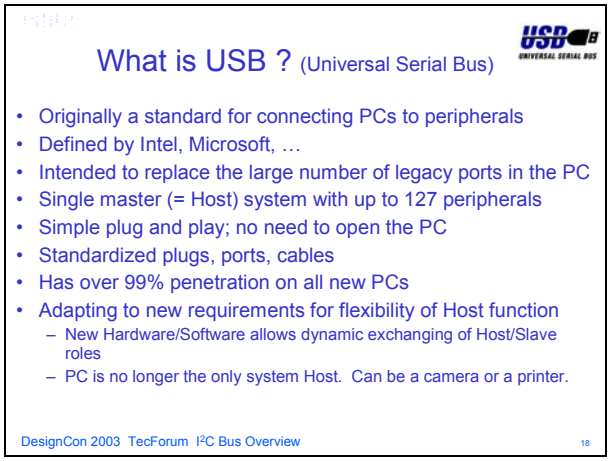
- DLC: data length code
- CRC: cyclic redundancy check (remainder of a division calculation). All devices that pass the CRC will acknowledge or will generate an error flag after the data frame finishes.
- ACK: acknowledge.
- Error frame: (at least) 6 consecutive dominant bits then 7 recessive bits.

A message 'filter' can be programmed to test the 11-bit identifier and one or two bytes of the data (In general up to 32 bits) to decide whether to accept the message and issue an interrupt. It could also look at all of the 29-bit identifier.

I²C products from many manufacturers are all compatible but CAN hardware will be selected and dedicated for each particular system design. Some CAN transceivers will be compatible with others, but that is more likely to be the exception than the rule. CAN designs are usually individual systems that are not intended to be modified. Philips parts greatly enhance the feature of reliability by their ability to use part-broken bus wiring and disconnect themselves if they are recording too many bus errors.

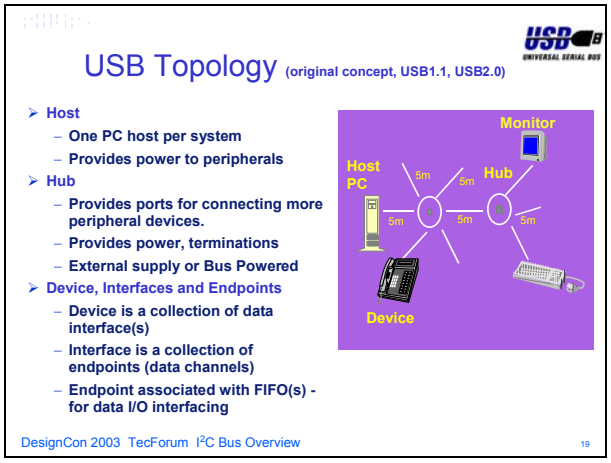
There are several aspects to reliability - availability of the bus when important data needs to be sent, the possibility of bits in a message being corrupted by noise etc., and the consequences of electrical/mechanical failure modes in the wiring. All these aspects are treated seriously by the CAN specifications and the suppliers of the interface ICs - for example Philips believes conventional high voltage IC processes are not good enough and uses Silicon-on-insulator technology to increase ruggedness and avoid the alternative of using common-mode chokes for protection. To give an example of immunity, a transceiver on 5 V must be able to cope with jump-start and load-dump voltages on its supply or bus wires. That is 40 V on the supply and +/- 40 V on the bus lines, plus transients of –150 V/+100 V capacitively coupled from a pulse generator in a test circuit!

USB Overview



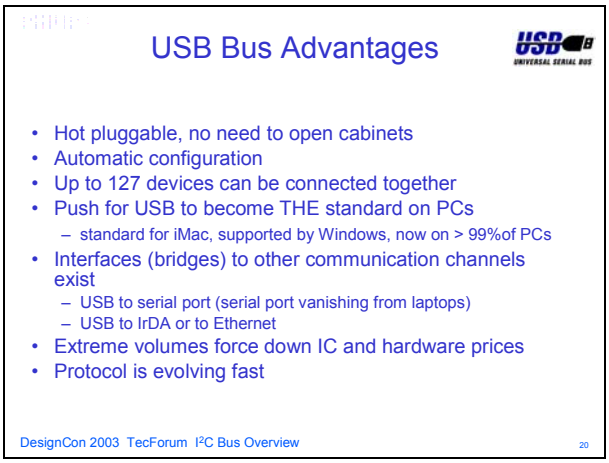
Slide 18

USB is the most complex of the buses presented here. While its hardware and transceivers are relatively simple, its software is complex and is able to efficiently service many different applications with very different data rates and requirements. It has a 12 Mbps rate (with 200 Mbps planned) over a twisted pair with a 4-pin connector (2 wires are power supply). It also is limited to short distances of at most 5 meters (depends on configuration). Linux supports the bus, although not all devices that can plug into the bus are supported. It is synchronous and transmits in special packets like a network. Just like a network, it can have several devices attached to it. Each device on it gets a time-slice of exclusive use for a short time. A device can also be guaranteed the use of the bus at fixed intervals. One device can monopolize it if no other device wants to use it.



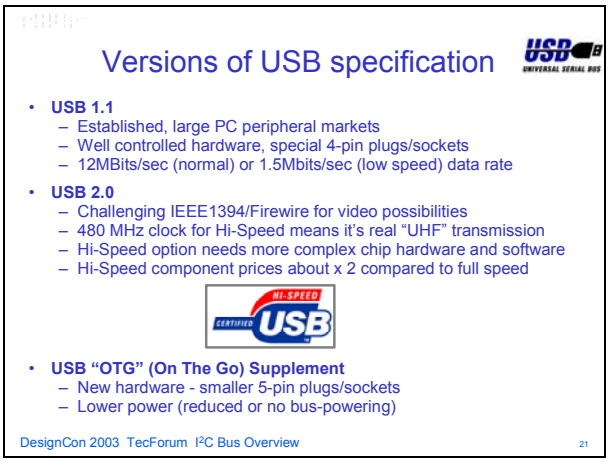
Slide 19

Slide 19 shows a typical USB configuration.



Slide 20

USB aims at mass-market products and design-ins may be less convenient for small users. The serial port is vanishing from the laptop and gone from iMac. There are hardware bridges available from USB to other communication channels but there can be higher power consumption to go this way. Philips is innovating its USB products to minimize power and offer maximum flexibility in system design.



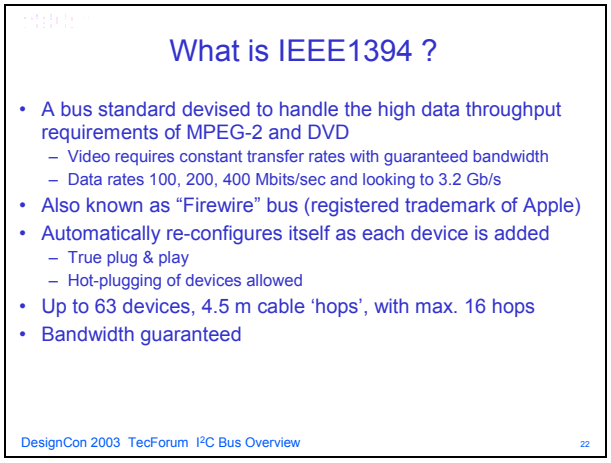
Slide 21

For USB 1.1 and 2.0 the hardware is well established. The shape of the plug/socket at Host end is different from the shape at the peripheral end. USB is always a single point-to-point link over the cable. To allow connection of multiple peripherals a HUB is introduced. The Hub functions to multiplex the data from the 'downstream' peripherals into one 'upstream' data linkage to the Host. In Hi-Speed systems it is necessary for the system to start communicating as a normal USB 1.1 system and then additional hardware (faster transceivers etc) is activated to allow a higher speed. The Hi-Speed system is much more complex (hardware/software) than normal USB (1.1). For USB

and Hi-Speed the development of 'stand-alone' Host ICs such as ISP1161 and ISP1561 allowed the Host function to be embedded in products such as Digital Still Cameras or printers so that more direct transfer of data was possible without using the path Camera → PC → Printer under control of the PC as the host. That two step transfer involves connecting the camera to the PC (one USB cable) and also the PC to the printer (second USB cable). The goal is to do without the PC.

The next step involved the shrinking of the USB connector hardware, to make it more compatible with small products like digital cameras, and making provision (extra pin) for dynamic exchanging of Host and slave device functions without removing the USB cable for reversing the master/slave connectors. The new hardware and USB specification version is called "On The Go" (OTG). The OTG specification no longer requires the Host to provide the 1/2 A power supply to peripherals and indeed allows arbitration to determine whether Host or peripheral (or neither) will provide the system power.

1394 Overview



Slide 22

1394 may claim to be more proven or established than USB but both are 'emerging' specifications that are trying to out-do each other! Philips strongly supports BOTH. 1394 was chosen by Philips as the bus to link set-top boxes, DVD, and digital TVs. 1394 has an 'a' version taking it to 400 Mb/sec and more recently a 'b' version for higher speed and to allow longer cable runs, perhaps 100 meter hops!

1394 sends information over a PAIR of twisted pairs. One for data, the other is the clocking strobe. The clock is simply recovered by an Ex-Or of the data and strobe line signals. No PLL is needed. There is provision for lots of remote device powering via the cable if the 6-pin plug connection version is used. The power wires are specified to well over 1A at 8-30 volts (approx) - leading to some unkind references to a 'fire' wire!

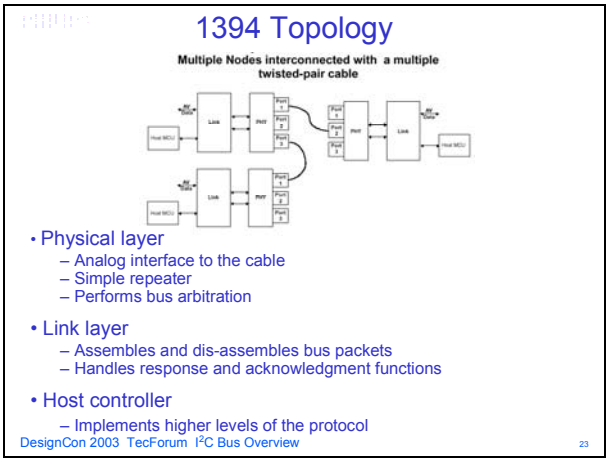
1394 software or message format consists of timeslots within which the data is sent in blocks or 'channels'. For real-time data transfer it is possible to guarantee the availability of one or more channels to guarantee a certain data rate. This is important for video because it's no good sending a packet of corrected data after a blank has appeared on the screen!

Microsoft says, "IEEE 1394 defines a single interconnection bus that serves many purposes and user scenarios. In addition to its adoption by the consumer electronics industry, PC vendors—including Compaq, Dell, IBM, Fujitsu, Toshiba, Sony, NEC, and Gateway—are now shipping Windows-based PCs with 1394 buses.

The IEEE 1394 bus complements the Universal Serial Bus (USB) and is particularly optimized for connecting digital media devices and high-speed storage devices to a PC. It is a peer-to-peer bus. Devices have more built-in intelligence than USB devices, and they run independently of the processor, resulting in better performance.

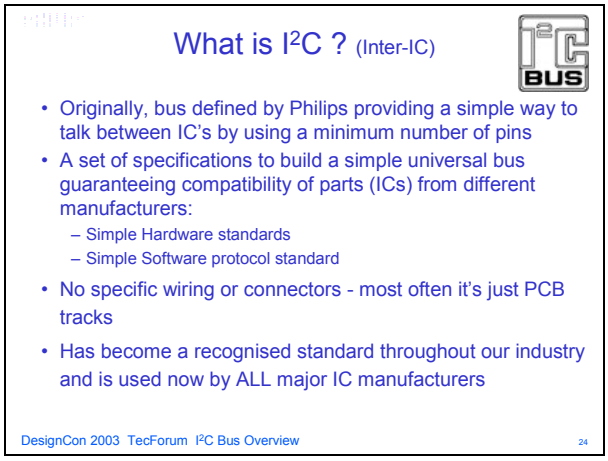
The 100-, 200-, and 400-Mbps transfer rates currently specified in the IEEE 1394a standard and the proposed enhancements in 1394b are well suited to meeting the throughput requirements of multiple streaming input/output devices connected to a single PC. The licensing fee for use of patented IEEE 1394 technology has been established at US $0.25 per system.

With connectivity for storage, scanners, printers, and other types of consumer A/V devices, IEEE 1394 gives users all the benefits of a great legacy-free connector—a true Plug and Play experience and hassle-free PC connectivity."



Slide 23

I²C *Overview*



Slide 24

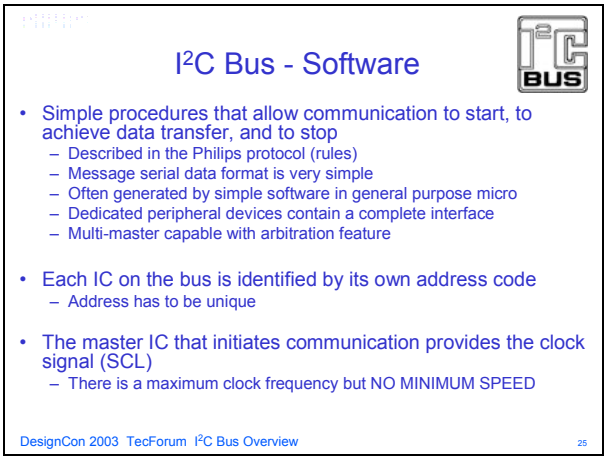
Originally, the I²C bus was designed to link a small number of devices on a single card, such as to manage the tuning of a car radio or TV. The maximum allowable capacitance was set at 400 pF to allow proper rise and fall times for optimum clock and data signal integrity with a top speed of 100 kbps. In 1992 the standard bus speed was increased to 400 kbps, to keep up with the ever-increasing performance requirements of new ICs. The 1998 I²C specification, increased top speed to 3.4 Mbits/sec. All I²C devices are designed to be able to communicate together on the same two-wire bus and system functional architecture is limited only by the imagination of the designer.

But while its application to bus lengths within the confines of consumer products such as PCs, cellular phones, car radios or TV sets grew quickly, only a few system integrators were using it to span a room or a building. The I²C bus is now being increasingly used in multiple card systems, such as a blade servers, where the I²C bus to each card needs to be isolatable to allow for card insertion and removal while the rest of the system is in operation, or in systems where many more devices need to be located onto the same card, where the total device and trace capacitance would have exceeded 400 pF.

New bus extension & control devices help expand the I²C bus beyond the 400 pF limit of about 20 devices and allow control of more devices, even those with the same address. These new devices are popular with designers as they continue to expand and increase the range of use of I²C devices in maintenance and control applications.

I²C Features

- Only two bus lines are required: a serial data line (SDA) and a serial clock line (SCL).
- Each device connected to the bus is software addressable by a unique address and simple master/slave relationships exist at all times; masters can operate as master-transmitters or as master-receivers.
- It's a true multi-master bus including collision detection and arbitration to prevent data corruption if two or more masters simultaneously initiate data transfer.
- Serial, 8-bit oriented, bi-directional data transfers can be made at up to 100 kbit/s in the Standard-mode, up to 400 kbit/s in the Fast-mode, or up to 3.4 Mbit/s in the High-speed mode.
- On-chip filtering (50 ns) rejects spikes on the bus data line to preserve data integrity.
- The number of ICs that can be connected to the same bus segment is limited only by the maximum bus capacitive loading of 400 pF.



Slide 25

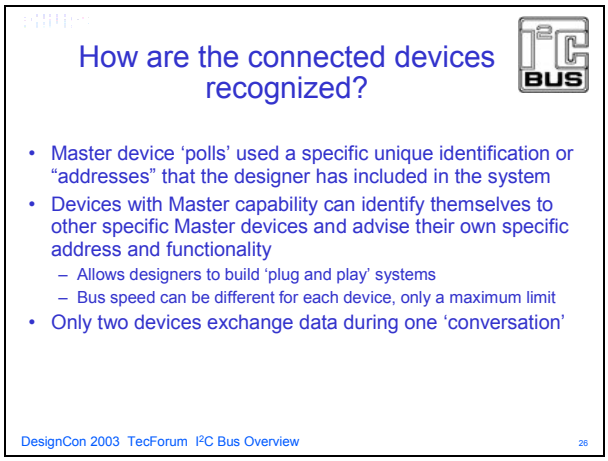
I²C Communication Procedure

One IC that wants to talk to another must:

1) Wait until it sees no activity on the I²C bus. SDA and SCL are both high. The bus is 'free'.
2) Put a message on the bus that says 'its mine' - I have STARTED to use the bus. All other ICs then LISTEN to the bus data to see whether they might be the one who will be called up (addressed).
3) Provide on the CLOCK (SCL) wire a clock signal. It will be used by all the ICs as the reference time at which each bit of DATA on the data (SDA) wire will be correct (valid) and can be used. The data on the data wire (SDA) must be valid at the time the clock wire (SCL) switches from 'low' to 'high' voltage.
4) Put out in serial form the unique binary 'address' (name) of the IC that it wants to communicate with.
5) Put a message (one bit) on the bus telling whether it wants to SEND or RECEIVE data from the other chip. (The read/write wire is gone!)

6) Ask the other IC to ACKNOWLEDGE (using one bit) that it recognized its address and is ready to communicate.
7) After the other IC acknowledges all is OK, data can be transferred.
8) The first IC sends or receives as many 8-bit words of data as it wants. After every 8-bit data word the sending IC expects the receiving IC to acknowledge the transfer is going OK.
9) When all the data is finished the first chip must free up the bus and it does that by a special message called 'STOP'. It is just one bit of information transferred by a special 'wiggling' of the SDA/SCL wires of the bus.

The bus rules say that when data or addresses are being sent, the DATA wire is only allowed to be changed in voltage (so, '1', '0') when the voltage on the clock line is LOW. The 'start' and 'stop' special messages BREAK that rule, and that is how they are recognized as special.



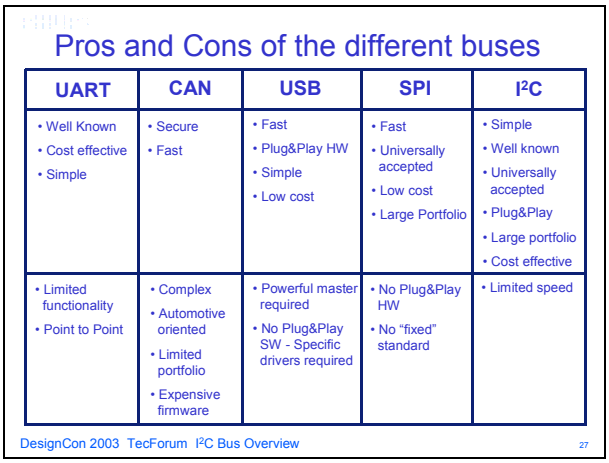
Slide 26

Any device with the ability to initiate messages is called a 'master'. It might know exactly what other chips are connected, in which case it simply addresses the one it wants, or there might be optional chips and it then checks what's there by sending each address and seeing whether it gets any response (acknowledge).

An example might be a telephone with a micro in it. In some models, there could be EEPROM to guarantee memory data, in some models there might be an LCD display using an I²C driver. There can be software written to cover all possibilities. If the micro finds a display then it drives it, otherwise the program is arranged to skip that software code. I²C is the simplest of the buses in this presentation. Only two chips are involved in any one communication - the Master that initiates the signals and the one Slave that responded when addressed.

But several Masters could control one Slave, at different times. Any 'smart' communications must be via the transferred DATA, perhaps used as address info. The I²C bus protocol does not allow for very complex systems. It's a 'keep it simple' bus. But of course system designers are free to innovate to provide the complex systems - based on the simple bus.

Serial Bus Comparison Summary



Slide 27

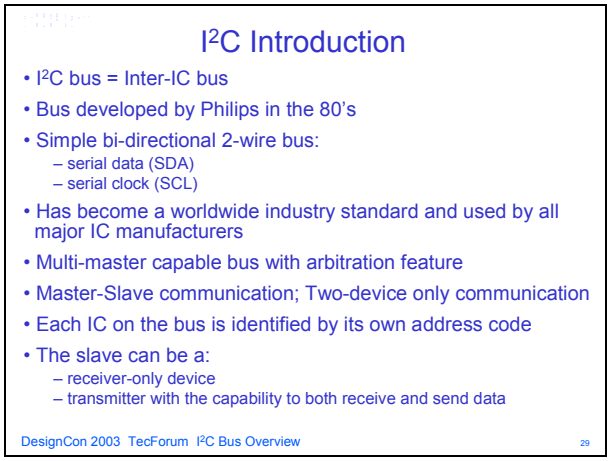
Most Philips CAN devices are not plug & play. That is because for MOST chips the system needs to be fixed and nothing can be added later. That is because an added chip is EXPECTED to take part in EVERY data conversation but will not know the clock speed and cannot synchronize. That means it falsely reports a bus timing error on every message and crashes the system.

Philips has special transceivers that allow them listen to the bus without taking part in the conversations. This special feature allows them to synchronize their clocks and THEN actively join in the conversations. So, from Philips, it becomes POSSIBLE to do some minor plug/play on a CAN system.

USB/SPI/MicroWire and mostly UARTS are all just 'one point to one point' data transfer bus systems. USB then uses multiplexing of the data path and forwarding of messages to service multiple devices.

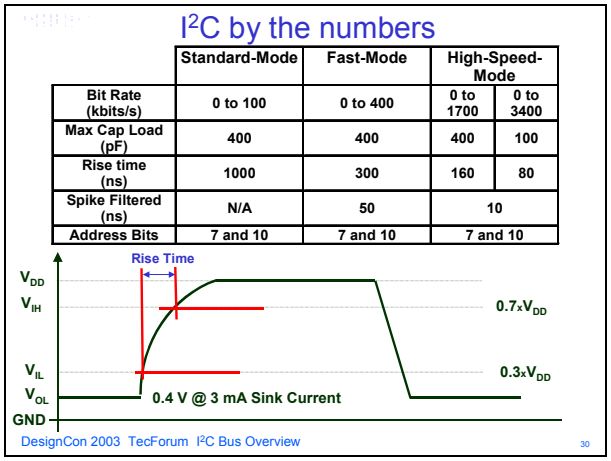
Only CAN and I²C use SOFTWARE addressing to determine the participants in a transfer of data between two (I²C) or more (CAN) chips all connected to the same bus wires. I²C is the best bus for low speed maintenance and control applications where devices may have to be added or removed from the system.

I²C Theory Of Operation



Slide 29

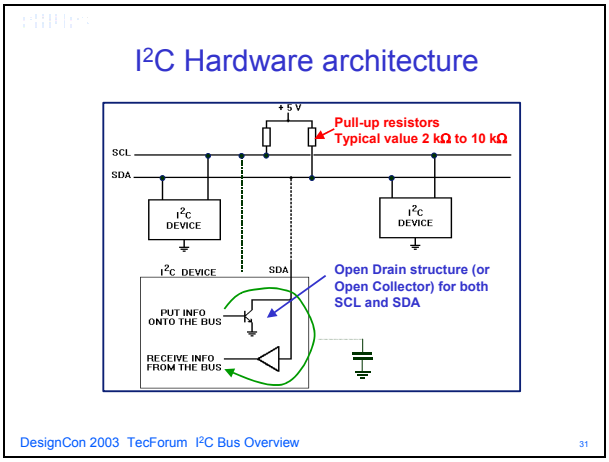
The I²C bus is a very easy bus to understand and use. Slides 29 and 30 give a good explanation of bus specifics and the different speeds. Many people have asked where rise time is measured and the specification stipulates it's between 30% and 70% of V_{DD}. This becomes important when buffers 'distort' the rising edges on the bus. By keeping any waveform distortions below 30% of V_{DD}, that portion of the rising edge will not be counted as part of the formal rise time.



Slide 30

I²C is a low to medium speed serial bus with an impressive list of features:

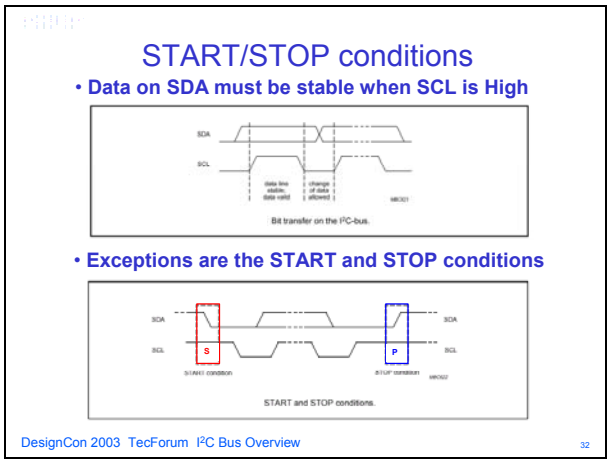
- Resistant to glitches and noise
- Supported by a large and diverse range of peripheral devices
- A well-known robust protocol
- A long track record in the field
- A respectable communication distance which can be extended to longer distances with bus extenders
- Compatible with a number of processors with integrated I²C ports (micro 8,16,32 bits) in 8048, 80C51 or 6800 and 68xxx architectures
- Easily emulated in software by any microcontroller
- Available from an important number of component manufacturers



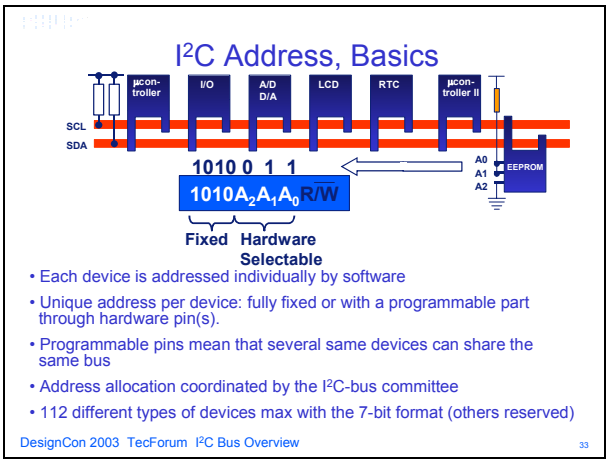
Slide 31

I²C *Bus Terminology*

- **Transmitter** - the device that sends data to the bus. A transmitter can either be a device that puts data on the bus of its own accord (a 'master-transmitter'), or in response to a request from data from another devices (a 'slave-transmitter').
- **Receiver** - the device that receives data from the bus.
- **Master** - the component that initializes a transfer, generates the clock signal, and terminates the transfer. A master can be either a transmitter or a receiver.
- **Slave** - the device addressed by the master. A slave can be either receiver or transmitter.
- **Multi-master** - the ability for more than one master to co-exist on the bus at the same time without collision or data loss.
- **Arbitration** - the prearranged procedure that authorizes only one master at a time to take control of the bus.
- **Synchronization** - the prearranged procedure that synchronizes the clock signals provided by two or more masters.
- **SDA** - data signal line (Serial DAta)
- **SCL** - clock signal line (Serial CLock)



Slide 32



Slide 33

START and STOP Conditions

Within the procedure of the I²C bus, unique situations arise which are defined as START (S) and STOP (P) conditions.

START: A HIGH to LOW transition on the SDA line while SCL is HIGH

STOP: A LOW to HIGH transition on the SDA line while SCL is HIGH

The master always generates START and STOP conditions. The bus is considered to be busy after the START condition. The bus is considered to be free again a certain time after the STOP condition. The bus stays busy if a repeated START (Sr) is generated instead of a STOP condition. In this respect, the START (S) and repeated START (Sr) conditions are functionally identical. The S symbol will be used as a generic term to represent both the START and repeated START conditions, unless Sr is particularly relevant.

Detection of START and STOP conditions by devices connected to the bus is easy if they incorporate the necessary interfacing hardware. However, microcontrollers with no such interface have to sample the SDA line at least twice per clock period to sense the transition.

HARDWARE CONFIGURATION

Slide 33 shows the hardware configuration of the I²C bus. The 'bus' wires are named SDA (serial data) and SCL (serial clock). These two bus wires have the same configuration. They are pulled-up to the logic 'high' level by resistors connected to a single positive supply, usually +3.3 V or +5 V but designers are now moving to +2.5 V and towards 1.8 V in the near future.

All the connected devices have open-collector (open-drain for CMOS - both terms mean only the lower transistor is included) driver stages that can transmit data by pulling the bus low, and high impedance sense amplifiers that monitor the bus voltage to receive data. Unless devices are communicating by turning on the lower transistor to pull the bus low, both bus lines remain 'high'. To initiate communication a chip pulls the SDA line low. It then has the responsibility to drive the SCL line with clock pulses, until it has finished, and is called the bus 'master'.

BUS COMMUNICATION

Communication is established and 8-bit bytes are exchanged, each one being acknowledged using a 9th data bit generated by the receiving party, until the data transfer is complete. The bus is made free for use by other ICs when the 'master' releases the SDA line during a time when SCL is high. Apart from the two special exceptions of start and stop, no device is allowed to change the state of the SDA bus line unless the SCL line is low.

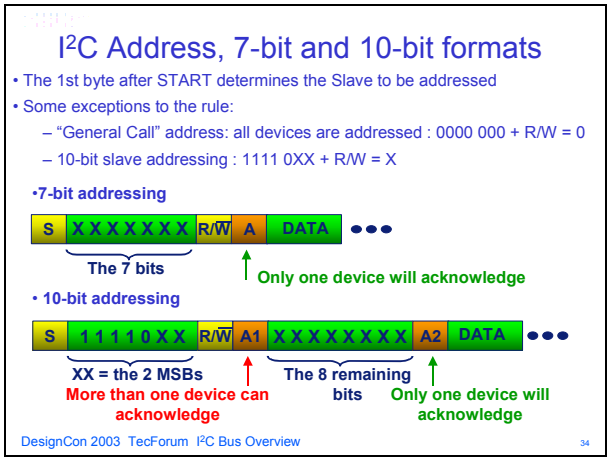
If two masters try to start a communication at the same time, arbitration is performed to determine a "winner" (the master that keeps control of the bus and continue the transmission) and a "loser" (the master that must abort its transmission). The two masters can even generate a few cycles of the clock and data that 'match', but eventually one will output a 'low' when the other tries for a 'high'. The 'low' wins, so the

'loser' device withdraws and waits until the bus is freed again.

There is no minimum clock speed; in fact any device that has problems to 'keep up the pace' is allowed to 'complain' by holding the clock line low. Because the device generating the clock is also monitoring the voltage on the SCL bus, it immediately 'knows' there is a problem and has to wait until the device releases the SCL line.

For full details of the bus capabilities refer to Philips Semiconductors Specification document 'The I²C bus specification' or 'The I²C bus from theory to practice' book by Paret and Fenger published by John Wiley & Sons.

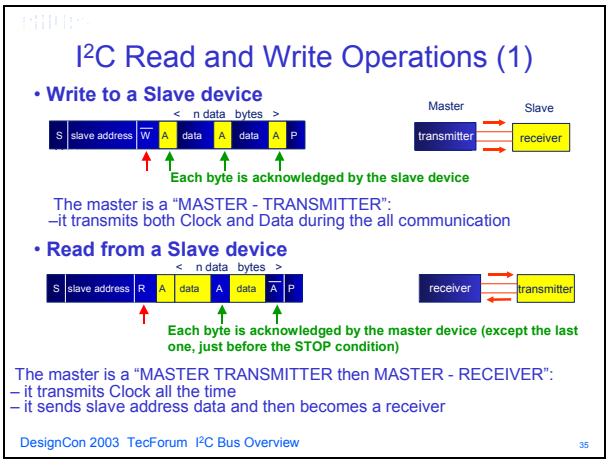
The I²C specification and other useful application information can be found on Philips Semiconductors web site at
http://www.semiconductors.philips.com/i2c/



Slide 34

Slide 34 shows the I²C address scheme. Any I²C device can be attached to the common I²C bus and they talk with each other, passing information back and forth. Each device has a unique 7-bit or 10-bit I²C address. For 7-bit devices, typically the first four bits are fixed, the next three bits are set by hardware address pins (A0, A1, and A2) that allow the user to modify the I²C address allowing up to eight of the same devices to operate on the I²C bus. These pins are held high to V_{CC}, sometimes through a resistor, or held low to GND.

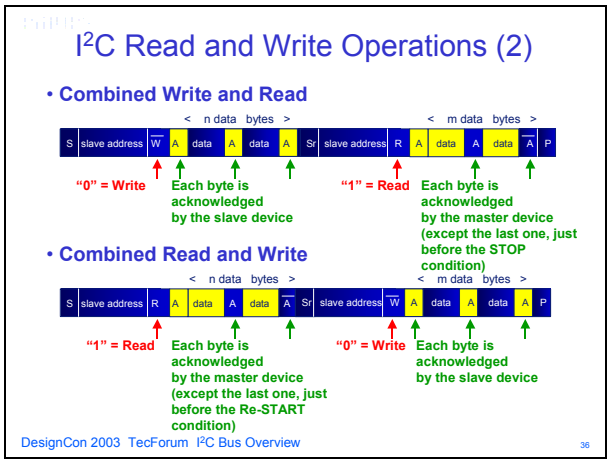
The last bit of the initial byte indicates if the master is going to send (write) or receive (read) data from the slave. Each transmission sequence must begin with the start condition and end with the stop condition.
On the 8th clock pulse, SDA is set 'high' if data is going to be read from the other device, or 'low' if data is going to be sent (write). During its 9th clock, the master releases SDA line to accomplish the Acknowledge phase. If the other device is connected to the bus, and has decoded and recognized its 'address', it will acknowledge by pulling the SDA line low. The responding chip is called the bus 'slave'.



Slide 35

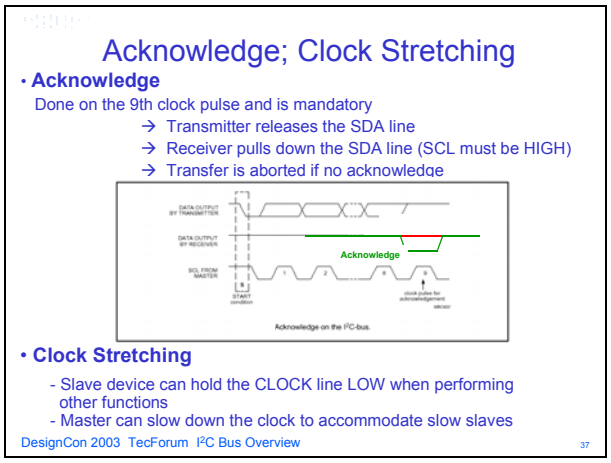
Terminology for Bus Transfer

- **F (FREE)** - the bus is free; the data line SDA and the SCL clock are both in the high state.
- **S (START) or S_R (Repeated START)** - data transfer begins with a start condition (not a start bit). The level of the SDA data line changes from high to low, while the SCL clock line remains high. When this occurs, the bus is 'busy'.
- **C (CHANGE)** - while the SCL clock line is low, the data bit to be transferred can be applied to the SDA data line by a transmitter. During this time, SDA may change its state, as along as the SCL line remains low.
- **D (DATA)** - a high or low bit of information on the SDA data line is valid during the high level of the SCL clock line. This level must be maintained stable during the entire time that the clock remains high to avoid misinterpretation as a Start or Stop condition.
- **P (STOP)** - data transfer is terminated by a stop condition, (not a stop bit). This occurs when the level on the SDA data line passes from the low state to the high state, while the SCL clock line remains high. When the data transfer has been terminated, the bus is free once again.



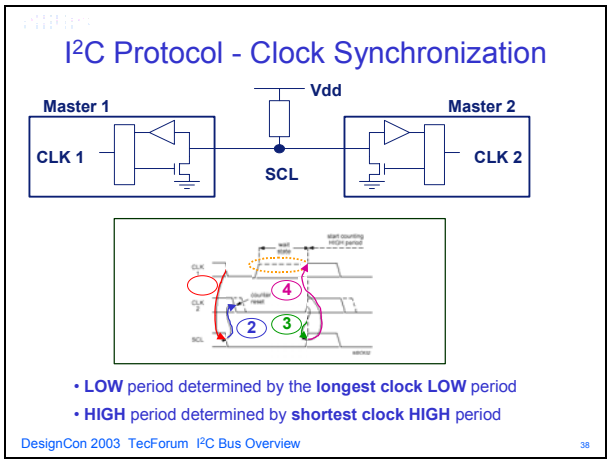
Slide 36

Slide 36 shows a combined read and write operation.



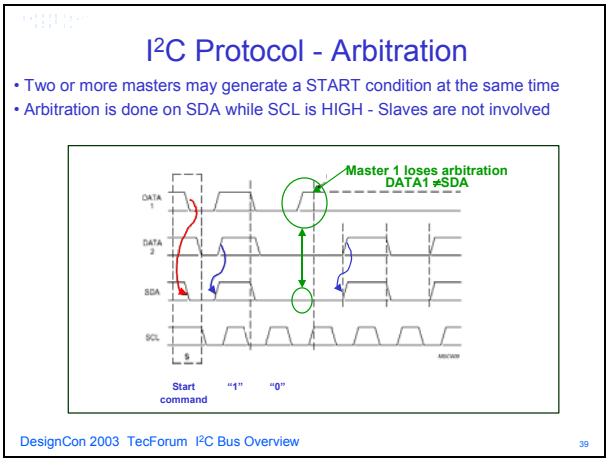
Slide 37

Slide 37 shows how the Acknowledge phase is done and how slave devices can stretch the clock signal. Most Philips slave devices do not control the clock line.



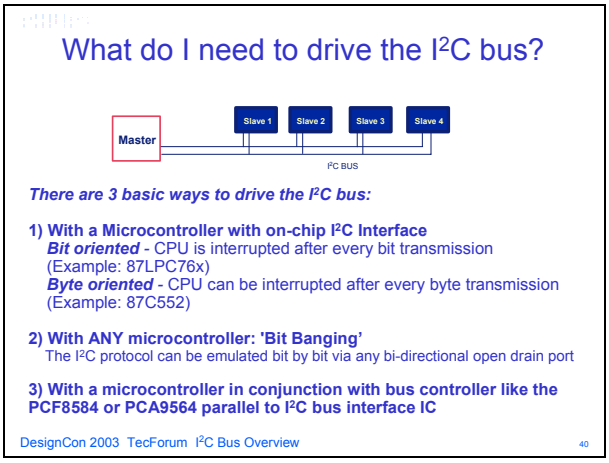
Slide 38

Slide 38 shows how multiple masters can synchronize their clocks, for example during arbitration. When bus capacitance affects the bus rise or fall times the master will also adjust its timing in a similar way.



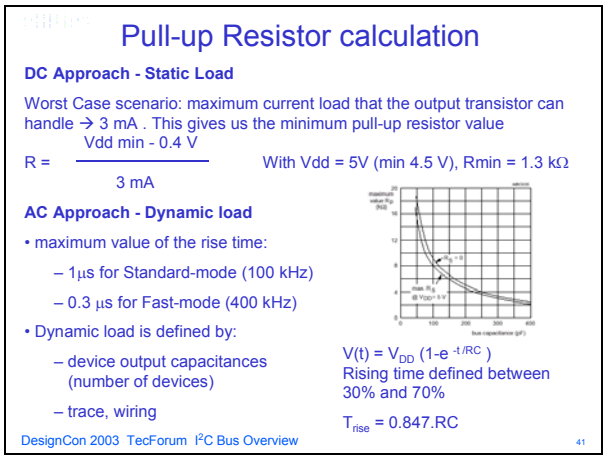
Slide 39

If there are two masters on the same bus, there are arbitration procedures applied if both try to take control of the bus at the same time. When two chips try to start communication at the same time they may even generate a few cycles of the clock and data that 'match', but eventually one will output a 'low' when the other tries for a 'high'. The 'low' wins, so the 'loser' device withdraws and waits until the bus is freed again. Once a master (e.g., microcontroller) has control, no other master can take control until the first master sends a stop condition and places the bus in an idle state.



Slide 40

Slide 40 shows there are multiple ways to control I²C slaves.



Slide 41

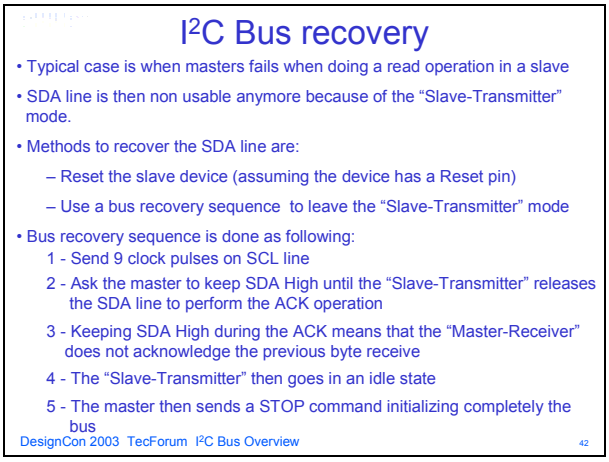
Slide 41 shows the typical resistor values needed for proper operation. C is the total capacitance on either SDA or SCL bus wire, with R as its pull-up resistor.

I²C *Designer Benefits*

- Functional blocks on the block diagram correspond with the actual ICs; designs proceed rapidly from block diagram to final schematic.
- No need to design bus interfaces because the I²C bus interface is already integrated on-chip.
- Integrated addressing and data-transfer protocol allow systems to be completely software-defined.
- The same IC types can often be used in many different applications.
- Design-time reduces as designers quickly become familiar with the frequently used functional blocks represented by I²C bus compatible ICs.
- ICs can be added to or removed from a system without affecting any other circuits on the bus.
- Fault diagnosis and debugging are simple; malfunctions can be immediately traced.
- Assembling a library of reusable software modules can reduce software development time.

I²C *Manufacturers Benefits*

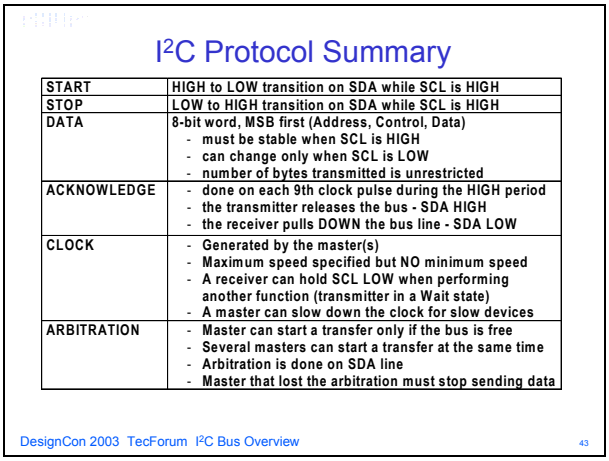
- The simple 2-wire serial I²C bus minimizes interconnections so ICs have fewer pins and there are not so many PCB tracks; result - smaller and less expensive PCBs
- The completely integrated I²C bus protocol eliminates the need for address decoders and other 'glue logic'
- The multi-master capability of the I²C bus allows rapid testing/alignment of end-user equipment via external connections to an assembly-line
- Increases system design flexibility by allowing simple construction of equipment variants and easy upgrading to keep design up-to-date
- The I²C bus is a de facto world standard that is implemented in over 1000 different ICs (Philips has > 400) and licensed to more than 70 companies



Slide 42

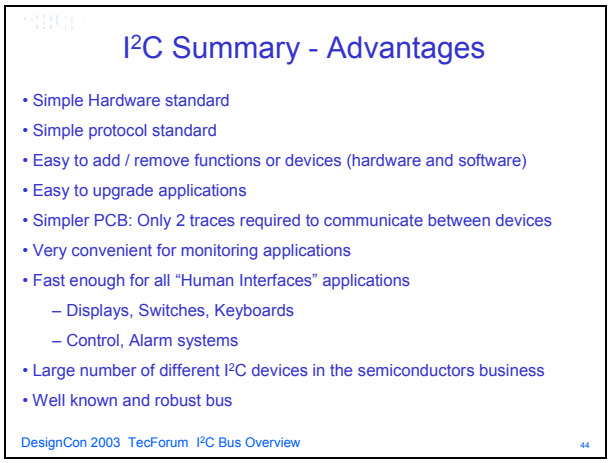
Slide 42 shows how a hung bus could be recovered. The bus can become hung for several reasons, e.g.….

1. Incorrect power-up and/or reset procedure for ICs
2. Power to a chip is interrupted – brown-outs etc
3. Noise on the wiring causes false clock or data signals



Slide 43

Slide 43 provides a summary of the I²C protocol.

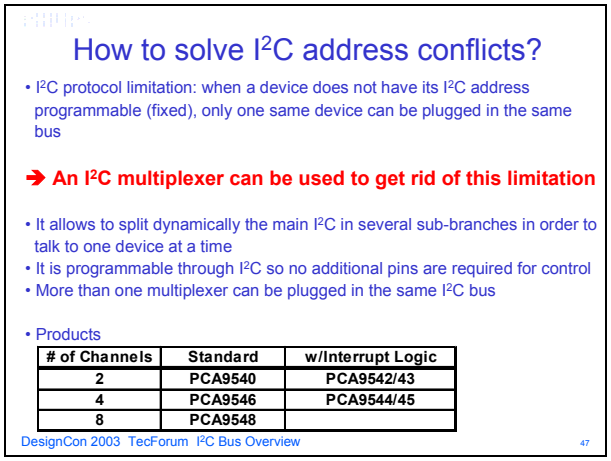


Slide 44

Slide 44 summarizes the advantages of the I²C bus.

Overcoming Previous Limitations

Address Conflicts



Slide 47

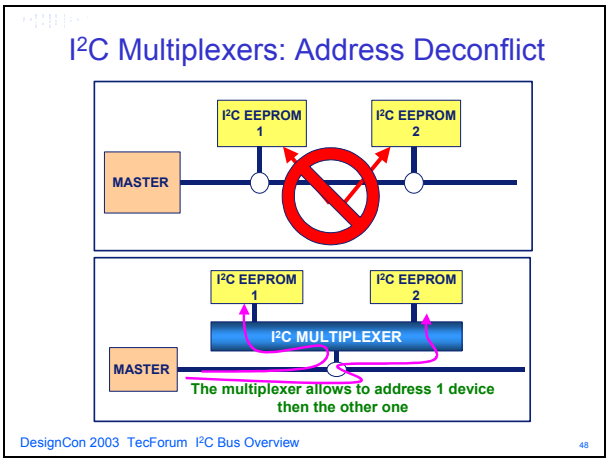
A 7 or 10-bit address that is unique to each device identifies an I²C device.
This address can be:

- Partly fixed, part programmable (allowing to have more than one of the same device on the same bus)
- Fully fixed allowing to have only one single same device on the device.

If more than one same "non programmable" device (fully fixed address) is required in a specific application, it is then necessary to temporarily remove the non-addressed device(s) from the bus when talking with the targeted device. I²C multiplexers allow to dynamically split the main I²C bus into 2, 4 or 8 sub-I²C buses. Each sub-bus (downstream channel) can be connected to the main bus (upstream channel) by a simple 2-byte I²C command.

For example, in an application where 4 identical I²C EEPROMs are used (EE1, EE2, EE3 and EE4), a four channel PCA9546 can be used. The master is plugged to the main upstream bus while the 4 EEPROMs are plugged to the 4 downstream channels (CH1, CH2, CH3 and CH4). If the master needs to perform an operation on EE3, it will have to:
- Connect the upstream channel to CH3
- Simply communicate with EE3.

EE1, EE2 and EE4 are electrically removed from the main I²C bus as long as CH3 is selected. Some of the I²C multiplexers offer an Interrupt feature, allowing collection of the different downstream Interrupts (generated by the downstream devices). An Interrupt output provides the information (transition from High to Low) to the master every time one or more Interrupt is generated (transition from High to Low) by any of the downstream devices.



Slide 48

The SCL/SDA upstream channel fans out to multiple SCx/SDx channels that are selected by the programmable control register. The I²C command is sent via the main I²C bus and is used to select or deselect the downstream channels.

The Multiplexers can select none or only one SCx/SDx channels at a time since they were designed primarily for address conflict resolution such as when multiple devices with the same I²C address need to be attached to the same I²C bus and you can only talk to one of the devices at a time.

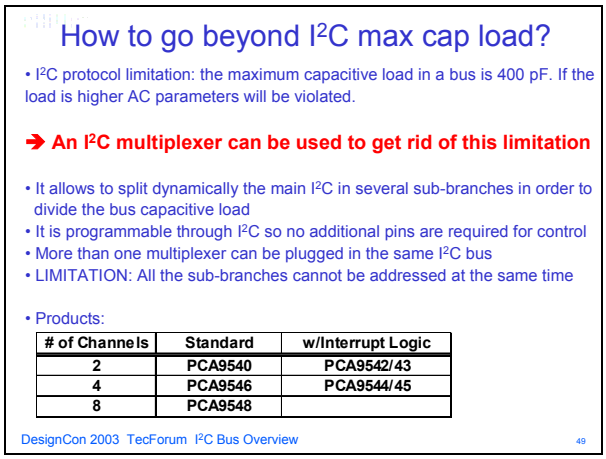
These devices are used in video projectors and server applications. Other applications include:

- Address conflict resolution (e.g., SPD EEPROMs on DIMMs).
- I²C sub-branch isolation

- I²C bus level shifting (e.g., each individual SCx/SDx channel can be operated at 1.8 V, 2.5 V, 3.3 V or 5.0 V if the device is powered at 2.5 V).

Interrupt logic inputs for each channel and a combined output are included on every multiplexer and provide a flag to the master for system monitoring. These devices do not isolate the capacitive loading on either side of the device so the designer must take into account all trace and device capacitance on both sides of the device and on any active channels. Pull up resistors must be used on all channels

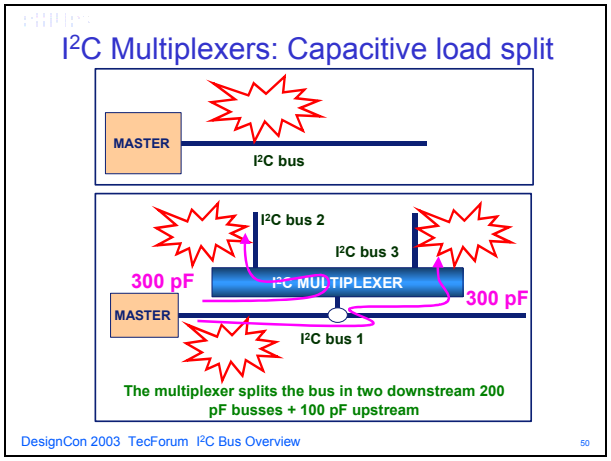
Capacitive Loading > 400 pF (isolation)



Slide 49

The I²C specification limits the maximum capacitive load in the bus to 400 pF. In applications where a higher capacitive load is required, 2 types of devices can be used:

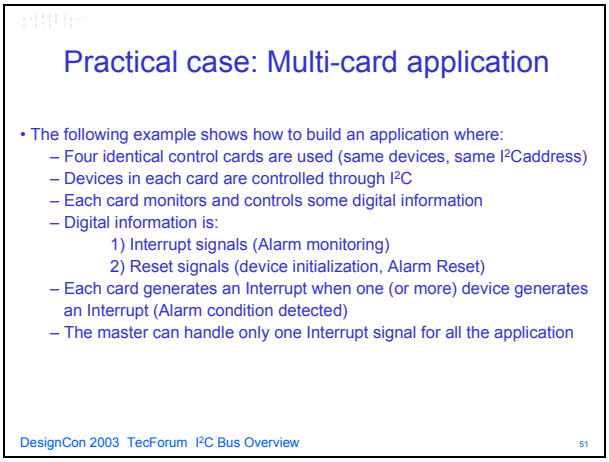
- I²C multiplexers and switches
- I²C buffers and repeaters



Slide 50

Multiplexers allow dynamic splitting of the overloaded I²C bus into several sub-branches with a total capacitive load smaller than the specified 400 pF. Note that this method does not allow the master to access all the buses at the same time. Only part of the bus will be accessible at a time.

Multiplexers allow bus splitting but do not have a buffering capability. Buffers and repeaters allow increasing the total capacitive load beyond the 400 pF without splitting the bus in several branches. If a PCA9515 is used, the bus can be loaded up to 800 pF with 400 pF on each side of the device.



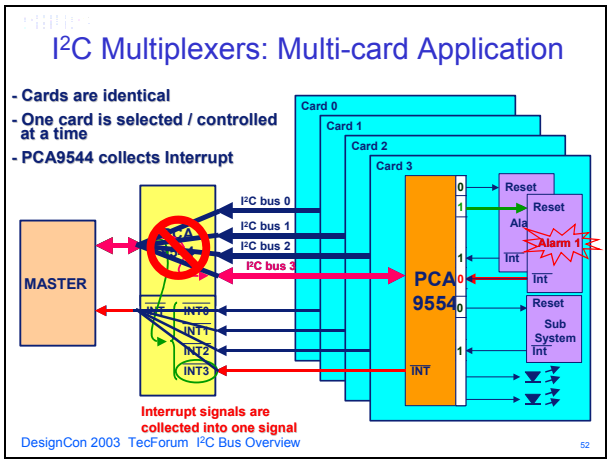
Slide 51

In this application, 4 identical cards are used. Identical means that the same devices are used, and that the I²C devices on each card have the same address. Each card monitors and controls some specific signal and those signals are controlled/monitored through the I²C bus by using a PCA9554 type device.

In this application, each card monitors some alarm system's sub system and controls some LEDs for visual status. Each alarm, when triggered, generates an Interrupt that is sent to the master for processing. PCA9554 collects the Interrupt signals and sends a "Card General Interrupt" to the master. When the master processes the alarm, it sends a Reset signal to the corresponding alarm to clear it. Master receives only an Interrupt signal, which is a combination of all the Interrupt signals in the cards. Since the cards are identical, it is then necessary to deconflict the different addresses and isolate the cards that are not accessed.

PCA9544 in this application has 2 functions:

- Deconflict the I²C addresses by creating 4 sub I²C busses that can be isolated
- Collect the Interrupt from each card and propagate a "General Interrupt" to the master



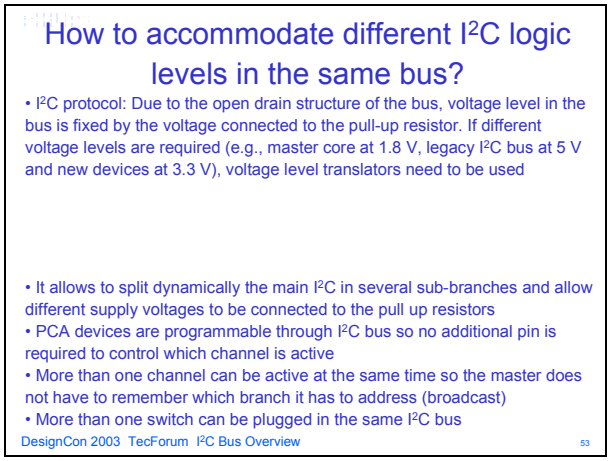
Slide 52

When one card in the application triggers an alarm condition, the PCA9554 collects it through one of its inputs and generates an Interrupt (at the card level). PCA9544 collects the Interrupts (from each card) and sends a "General Interrupt" to the master.

1. Master then interrogates the PCA9544 Interrupt status register in order to determine which card is in cause
2. Master then connects the corresponding sub I²C channel in order to interrogate the PCA9554 by reading its Input register.
3. Once 1) and 2) are done, Master knows which alarm has been triggered and can process it

When this is done, Master can then clear the corresponding alarm by accessing the corresponding card and programming the PCA9554 (write in the output register)

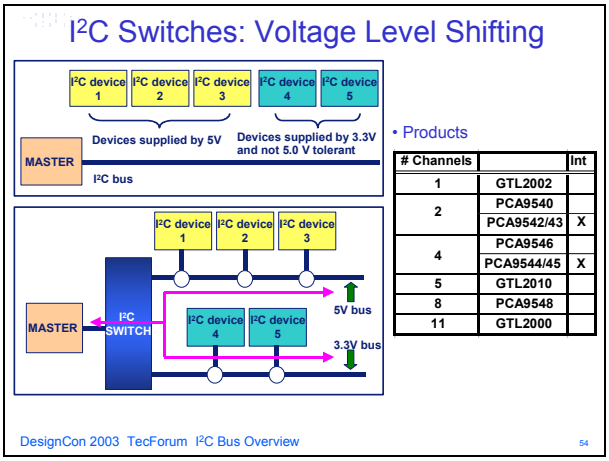
Voltage Level Translation



Slide 53

Due to the open drain architecture of the I²C bus, pull up resistors to a specific voltage is required. Once this is done, all the devices in the bus will have the same high level voltage value, determined by the voltage applied to the pull up resistors. In applications where several voltage levels are required (e.g. accommodate legacy architecture at 5.0 V with newer devices working at 3.3 V only), I²C switches allow creating a bus with different high level voltage values at a minimum cost.

In this example, we have an existing 5.0 V I²C bus and we want to add some new features with devices "non 5.0 V tolerant". An I²C bus can be used. The master controlling the existing and new devices will be located in the upstream channel and the 2 downstream channels will be used with pull up resistors at 5.0 V in one and to 3.3 V in the other one. Software changes will include the drivers for the new 3.3 V devices and a simple 2-byte command allows to program the I²C switch with the 2 downstream channels active all the time. The master then sees an I²C bus with new devices and does not have to take care of the high level voltage required to make them work correctly. It does not have to care either about the location of the device it needs to talk to (downstream channel 0 or channel 1) since both are active at the same time.



Slide 54

The SCL/SDA upstream channel fans out to multiple SCx/SDx channels that are selected by the programmable control register. The Switches can select individual SCx/SDx channels one at a time, all at once or in any combination through I²C commands and very primary designed for sub-branch isolation and level shifting but also work fine for address conflict resolution. Just make sure you do not select two channels at the same time.

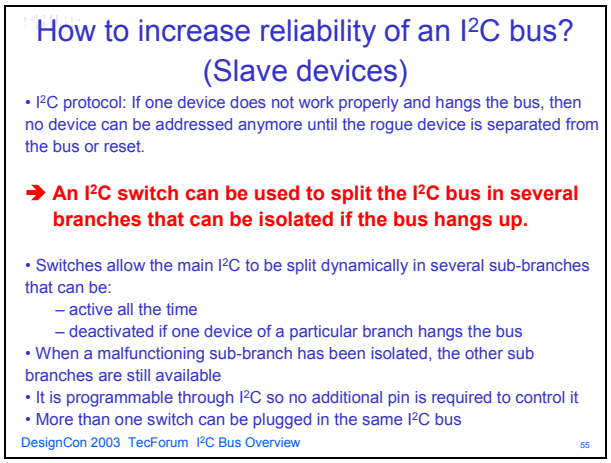
Applications are the same as for the multiplexers but since multiple channels can be selected at the same time the switches are really great for I²C bus level shifting (e.g., individual SCx/SDx channels at 1.8 V, 2.5 V, 3.3 V or 5.0 V if the device is powered at 2.5 V). A

hardware reset pin has been added to all the switches. It provides a means of resetting the bus should it hang up, without rebooting the entire system and is very useful in server applications where it is impractical to reset the entire system when the I²C bus hangs up. The switches reset to no channels selected.

Interrupt logic inputs and output are available on the PCA9543 and PCA9545 and provide a flag to the master for system monitoring. The PCA9546 is a lower cost version of the PCA9545 without Interrupt Logic. The PCA9548 provides eight channels and are more convenient to use then dual 4 channel devices since the device address does not have to shift.

These devices do not isolate the capacitive loading on either side of the device so the designer must take into account all trace and device capacitance on both sides of the device (active channels only). Pull up resistors must be used on all channels.

Increase I²C Bus Reliability (Slave Devices)

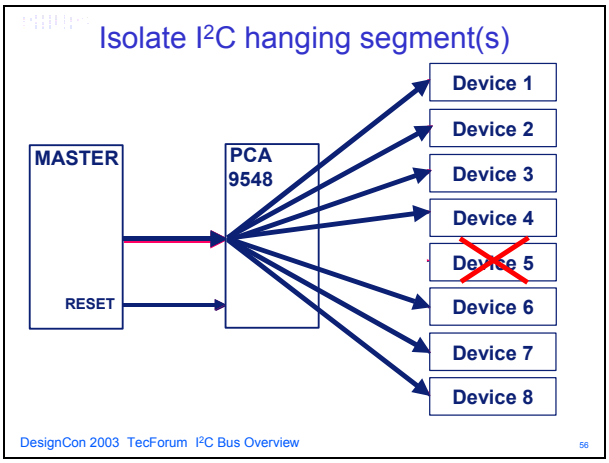


Slide 55

Due to the open drain architecture of the I²C bus, if a device fails in the bus and keeps the clock or data line at a high or low level, the bus is stuck in this configuration and no device can be controlled until the failed device is isolated from the I²C bus. Some architectures require a bus to still be operational even though one or more devices failed and can no longer operate normally.

An I²C switch with a Reset capability allows to:

- Split dynamically the I²C bus in several sub-branches (with one or several devices on each)
- Disconnect all the devices in case the bus hangs
- Reprogram the bus and isolate one or more branch that is not working properly.



Slide 56

Let's take an example where 8 devices (DEV1 to DEV8) are used and where the functional devices need to be controlled even though one or more devices are failing.

Slave devices will be located on each downstream channel of the PCA9548 (8-channel switch with Reset) (CH1 to CH8). At power up, all the downstream channels are disabled. The master (located in the upstream channel) sends a 2 byte command enabling all the downstream channels. The I²C bus is then a normal bus with a master and 8 slave devices. Let's assume that DEV4 (in CH4) fails. The bus then hangs and cannot be normally controlled by the master anymore.

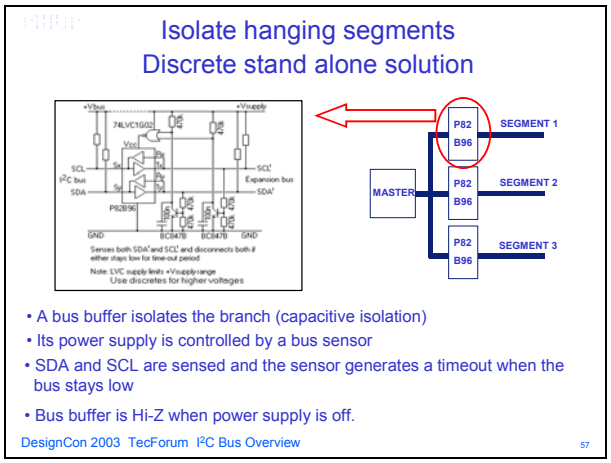
After detection of this condition, the master must go to a maintenance routine where:

- It resets the PCA9548, thus disabling all the downstream channels.
- It enables one by one all the downstream channels (CH1 to CH8) until the bus hangs again (CH4 active).

The master then knows that the device connected to CH4 is responsible of the failure

- It resets again the PCA9548 to take control of the I²C bus
- It programs all the functional channels active (CH1 to 3, CH5 to 8) and disables CH4

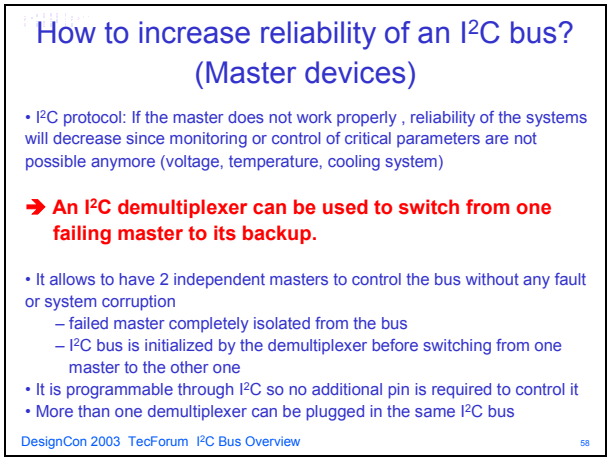
Note that this algorithm can also be applied if more than 1 channel hang the bus at the same time.



Slide 57

Slide 57 shows one discrete solution with option to set timing, by discrete capacitors, to isolate a bus segment.

Increasing I²C Bus Reliability (Master Devices)

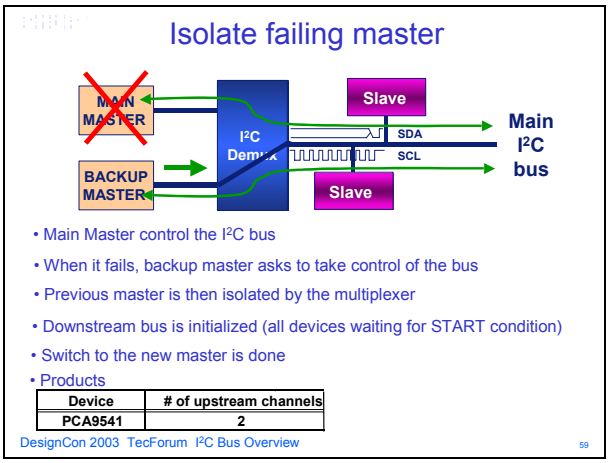


Slide 58

If the I²C master fails or does not work properly, reliability of applications will decrease since monitoring and control of essential parameters cannot be controlled anymore (e.g. temperature monitoring, voltage monitoring, cooling control). It is then often essential to have a backup I²C master to replace a mal functioning main I²C master. The I²C 2:1 master selector is then an essential device allowing switching between 2 masters.

It can be used in:

- A point to point application - master and backup master control one card
- A multi point application - master and backup master control several cards.

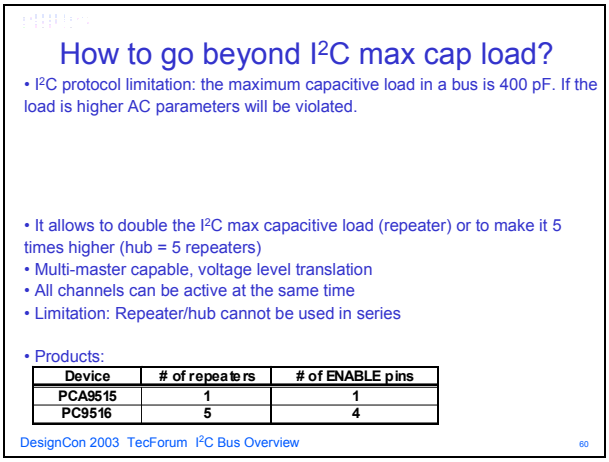


Slide 59

The 2:1 master selector allows switching between one master and its backup (and vice versa if the main master comes back on line). Before switching from one upstream channel to the other one, the device makes sure that the previous device is not on the bus anymore (fully isolated)

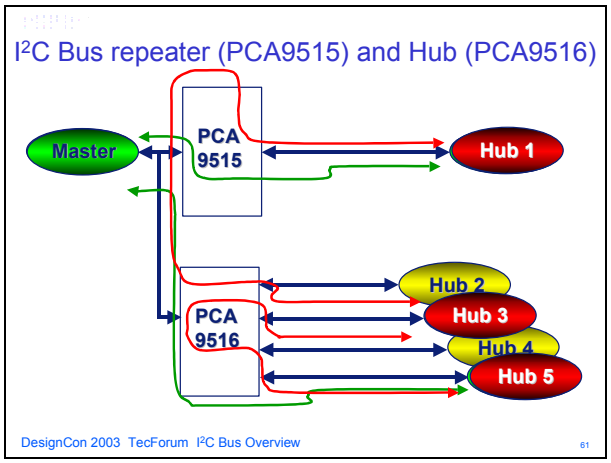
The switching is done after making sure that the downstream bus is in a "clean" configuration. All the downstream devices have been initialized again (essential when the previous master failed in the middle of a transaction and thus the bus is not well initialized) and the bus is in an idle configuration. This is done by converting the 2:1 master selector into a temporary master (just after isolating the failing master) allowing it to send the necessary I²C sequence (9 clock pulses on SCL while SDA is maintained high then a STOP command). While the sequence is done, the downstream I²C bus is well initialized and the switch to the new master can be performed automatically by the PCA9541.

Capacitive Loading > 400 pF (Buffer)



Slide 60

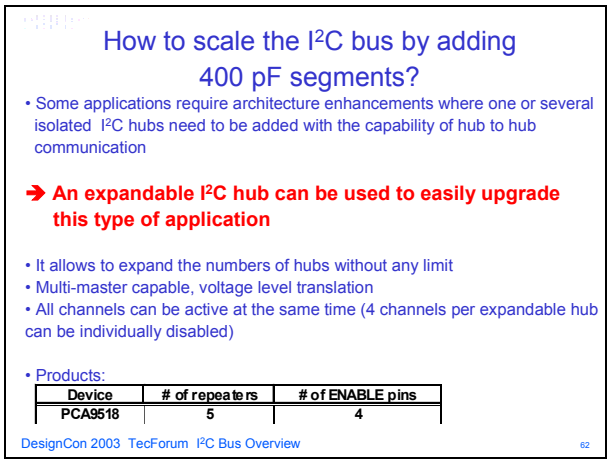
I²C bus repeaters and hubs allow increasing the maximum capacitive load on the bus without degrading the AC performances (rising and falling times) of the data and clock signals. They are multi-master capable.



Slide 61

- Repeaters allow doubling the capacitive load, 400 pF on each side of the device
- Hubs allow multiplying the load by 5 with 400 pF on each hub channel

In Slide 61, the possible communication paths are shown in green. No communication is possible over the red paths, no hub can communicate with any other hub. When communication between all hubs and the master is required then a multi-drop bus approach with P82B96 should be used.

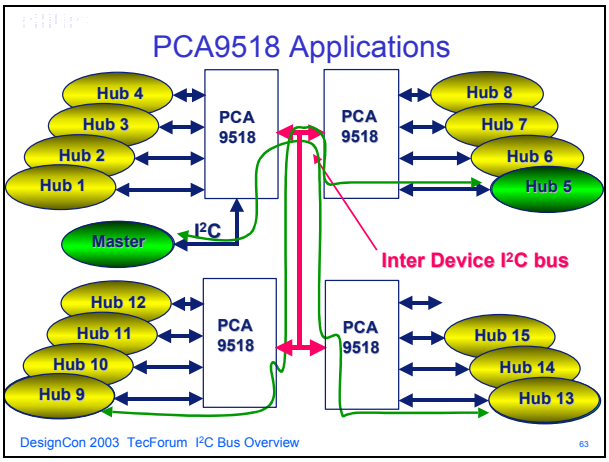


Slide 62

There are some applications where more than 5 channels are required. Sub Masters on Server Blades Application - Main Master is able to isolate any blade with the hardware enable pin via I²C & GPIO

Using the PCA9516 in this application, the sub masters can only talk with sub masters on the same hub or the main master since a low signal can not be sent through two hubs. Sub masters will not be able to arbitrate for bus control if located on different hubs. That is not ideal and limits the designers' ability to expand their I²C bus. The PCA9515 and the PCA9516 can only be used one device (either the PCA9515 or PCA9516) per system since low levels will not be transmitted through the second device.

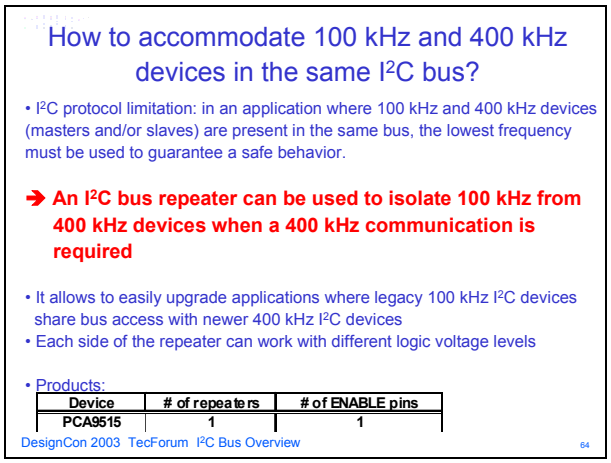
To overcome this limitation, the PCA9518 was released. Similar to the PCA9516 but with four extra open drain signal pins that allow the internal device logic to be interconnected into an unlimited number of segments with only one repeater delay between any two segments.



Slide 63

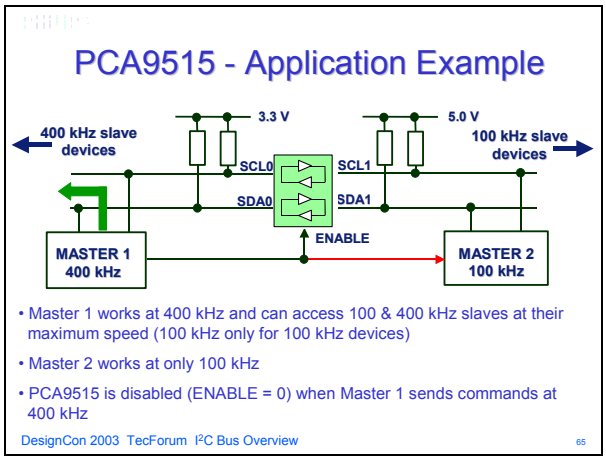
The PCA9518, like the PCA9515/16, is transparent to bus arbitration and contention protocols in a multi-master environment and any master can talk to any other master on any segment. The enable pins can be used to isolate four of the five segments per device. Place a pull up resistor on the un-isolatable segment and leave it unused if there is a requirement to enable or disable the segment.

Using the PCA9518 in this 15 hub application, any sub master can talk to any other sub master on any of the cards and the main master can talk with any sub master with only one repeater delay.



Slide 64

Due to the different I²C specification available (100 kHz, 400 kHz and now 3.4 MHz), devices designed for the 100 kHz specification are not suitable to work properly at 400 kHz, while the opposite is true. In applications where upgrades have been performed by using newer 400 kHz devices while keeping the 100 kHz legacy devices, it may become necessary to separate the 400 kHz devices from the 100 kHz devices when a 400 kHz I²C transfer is performed.



Slide 65

The PCA9515 can be used for this purpose. One side of the device will have all the devices running at 400 kHz while the other side will have all the devices running to 100 kHz.

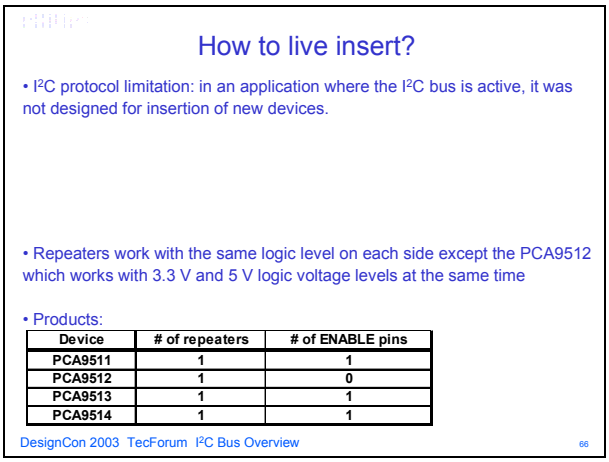
Note that each side of the PCA9515 can work at different logic voltage levels. For example, the "older" 100 kHz devices can run at 5.0 V while the "newer" 400 kHz devices can work at 3.3 V.

There could also be more than one master in the bus:

- One main master with the ability of choosing between 100 kHz and 400 kHz depending on the devices it needs to talk to.
- Two masters, one working at 100 kHz only (can be part of the system legacy) and another one working at 400 kHz.

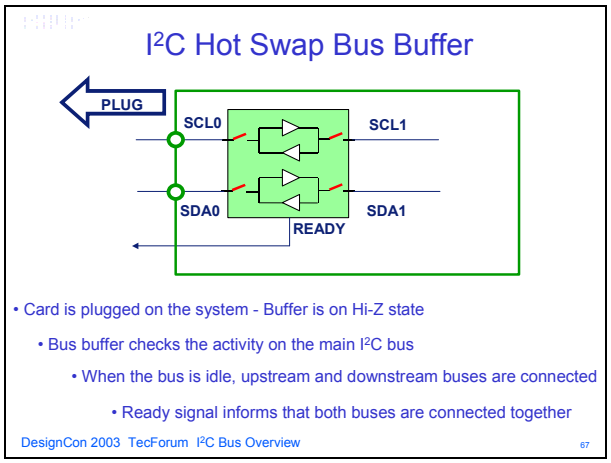
In the 1st case, the master located in the "400 kHz only" side has the capability to control the PCA9515's ENABLE pin in order to disable the device when a 400 kHz communication is initiated (the "100 kHz only" side will then not see the communication). During a 100 kHz communication, the PCA9515 is enabled to allow communication with the other side. In the 2nd case, both masters are located in each side of the PCA9515 and the control is basically the same as above for the 400 kHz devices.

Live Insertion into the I²C Bus



Slide 66

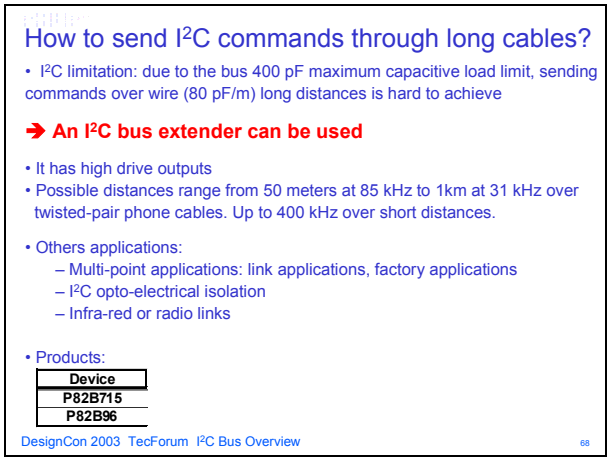
The I²C bus was never designed to be used in live insertion applications, but newer applications in for telecom cards that require 24/7 operation require the ability to be removed and inserted into an active system for maintenance and control applications.



Slide 67

The PCA9511/12/13/14 are designed for these types of live insertion applications.

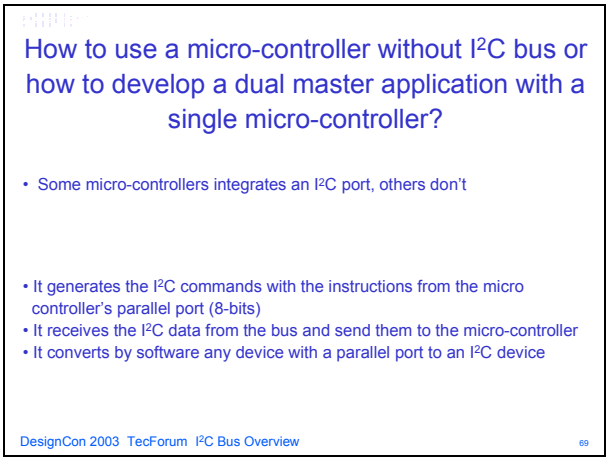
Long I²C *Bus Lengths*



Slide 68

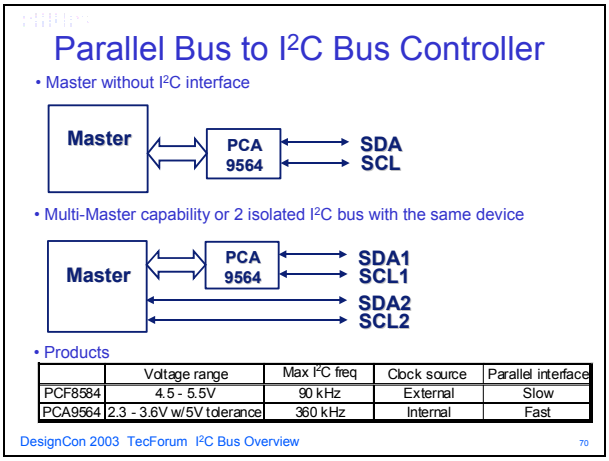
The P82B715 and P82B96 are designed for long distance transmission of the I²C bus.

Parallel to I²C *Bus Controller*



Slide 69

There are many applications where there is a need to convert 8 bits of parallel data into an I²C bus port. The PCF8584 and PCA9564 allow building a single I²C master system using the parallel port of a 8051 type microcontroller that does not have an I²C interface. It also allows building a double master system with using the built-in I²C interface and the parallel port of the same micro-controller.



Slide 70

Philips offers two devices, the PCF8584 and PCA9564. The PCA9564 is similar to the PCF8584 but operates at 2.3 to 3.6 V V_{CC} and up to 360 kHz with various enhancements added that were requested by engineers. The PCA9564 serves as an interface between most standard parallel-bus microcontrollers/ microprocessors and the serial I²C bus and allows the parallel bus system to communicate bi-directionally with the I²C bus. This commonly is referred as the bus master.

Communication with the I²C bus is carried out on a byte-wise basis using interrupt or polled handshake. It

controls all the I²C bus specific sequences, protocol, arbitration and timing. The internal oscillator in the PCA9564 is regulated to within +/- 10%.

	PCA9564	PCF8584	Comments
1. Voltage range	2.3-3.6V	4.5-5.5V	PCA9564 is 5V tolerant
2. Max I²C freq.	360 kHz	90 kHz	Faster I²C
3. Clock source	Internal	External	Less expensive and more flexible
4. Parallel interface processors	Fast	Slow	Compatible with faster

In addition, the PCA9564 has been made very similar to the Philips standard 80C51 microcontroller I²C hardware so existing code can be utilized with a few modifications.

Development Tools and Evaluation Board Overview

Purpose of the Development Tool and I²C Evaluation Board

To provide a low cost platform that allows Field Application Engineers, designers and educators to easily test and demonstrate I²C devices in a platform that allows multiple operations to be performed in a setting similar to a real system environment.

Slide 73

The I2C 2002-1A I2C evaluation board can be purchased from http://www.demoboard.com for $199.

Demo boards include at demoboard.com include:

I2C-Trace: I²C Bus Tracer Kit - I²C Monitor captures and displays I²C bus messages on any PC

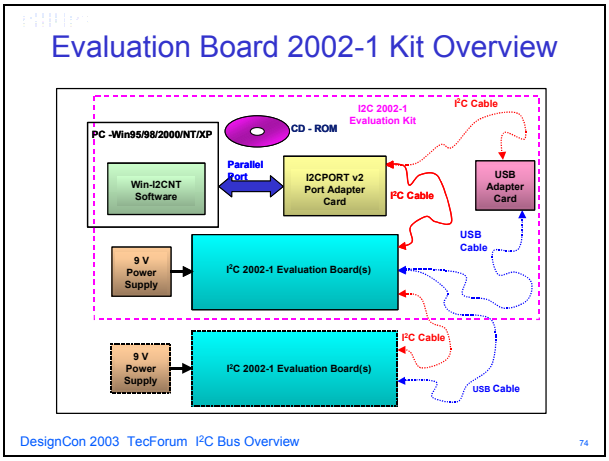
WIN-SMBUS: SMBUS Protocol S/W-H/W Kit - Supports SMBus ICs and the SMBus v1.0 protocol

WIN-I2CNTDLL: 32-bit Win-I2CNT kit including DLL driver and docs - Developer Kit for 32-bit embedded I²C applications

WIN-I2CNT: 32-bit I²C Software/Adapter kit for Win 95/98/ME/2000, NT 4.x - Enhanced kit for I2C control. Free updates from the Website

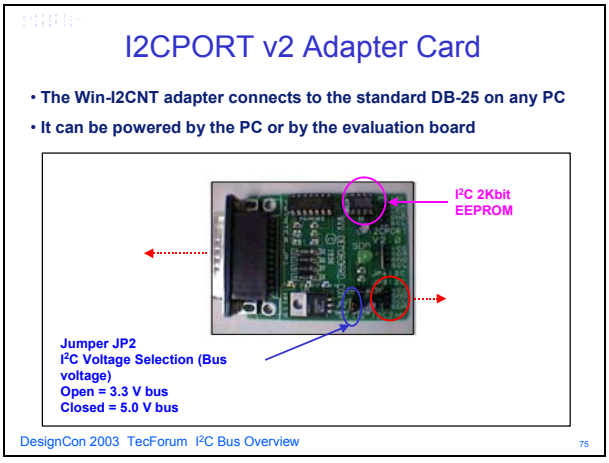
WIN-I2C: General Purpose legacy 16-bit I²C Software/Adapter kit - Basic Legacy Kit for I²C control with PCs running Windows 3.1x

I2CPORT: General Purpose I²C LPT Printer Port Adapter v1.0 - Generic I2C adapter (Not compatible with Win-I2C/Win-I2CNT Software)



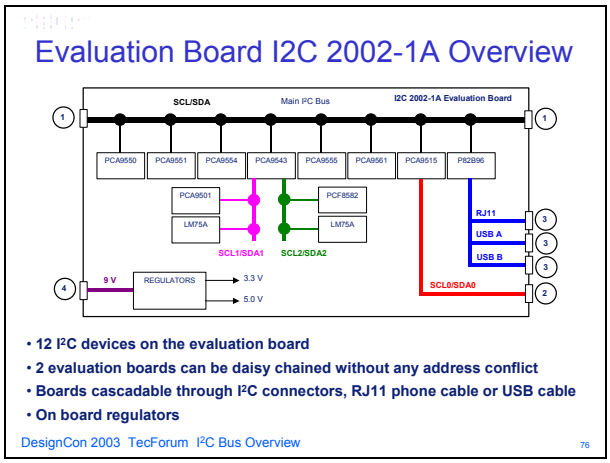
Slide 74

Slide 74 shows how the I2C 2002-1A kit is connected and shows how two evaluation boards can be used at the same time.



Slide 75

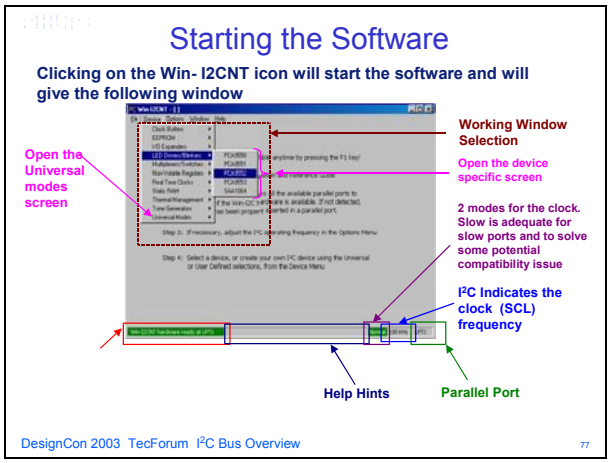
The I2CPORT v2 adapter card plugs into the parallel port and provides the interface between the Personal Computer and the I2C bus operating up to 150 kHz.



Slide 76

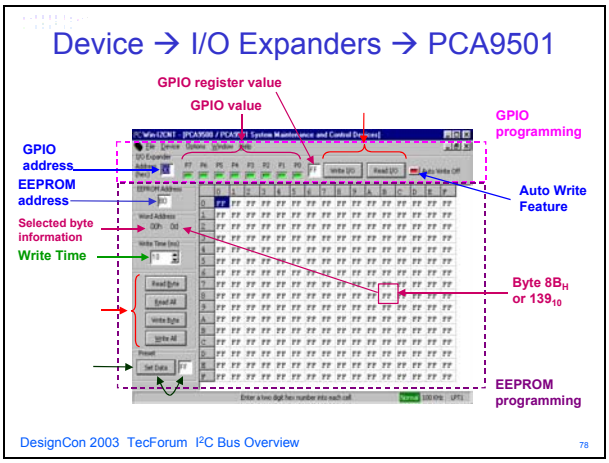
There are many new I²C devices on the evaluation board including GPIO, LED Blinkers, Switches, DIP Switches and Bus Buffers.

Win-I2CNT Screen Examples



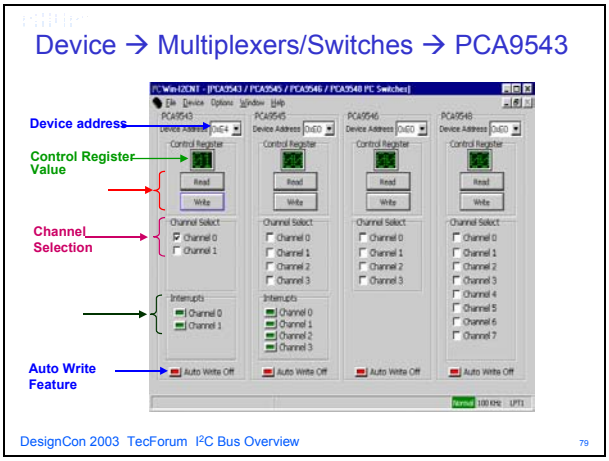
Slide 77

Slide 77 shows the start screen from which all the other screens are selected.



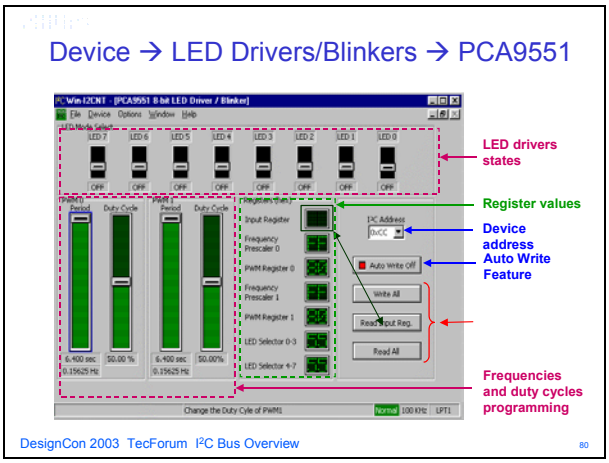
Slide 78

Slide 78 shows the 8 bit GPIO and 2 kbit EEPROM selection for the PCA9501.



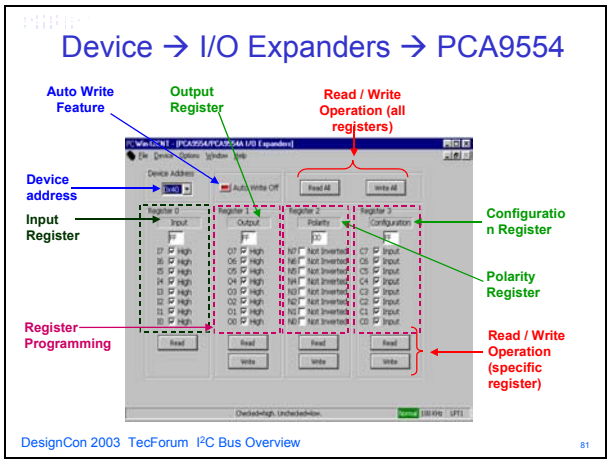
Slide 79

Slide 79 shows the selection possibilities for the PCA9543/45/46/48 switches.



Slide 80

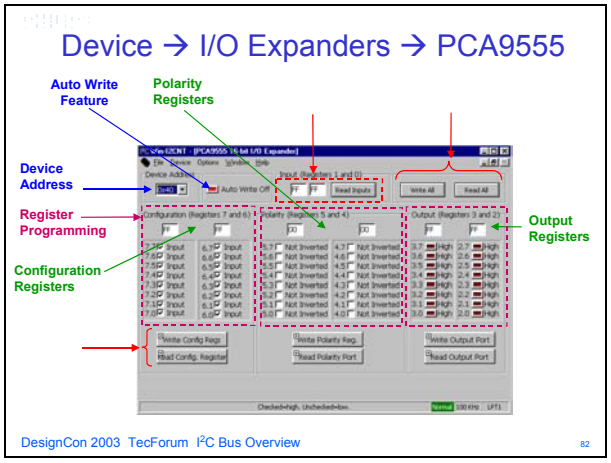
Slide 80 shows the selections for the PCA9551 8 bit LED Blinker. The PCA9551 has two PWMs and controls for each bit (ON, OFF, BLINK1 and BLINK2).



Slide 81

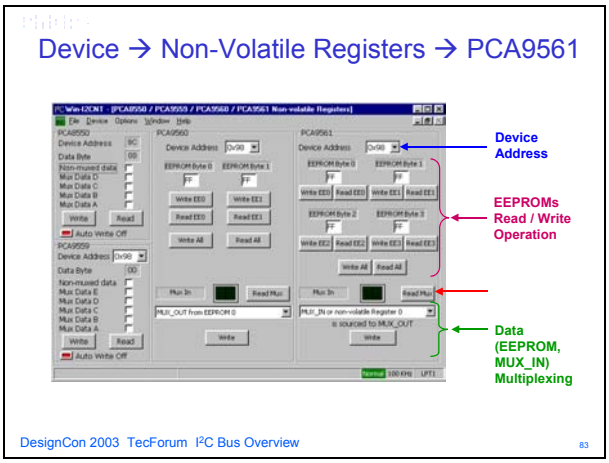
Slide 81 shows the 8 bit true output GPIO. Controls allow to:

- Program the bits a inputs or outputs
- Program the output state of output bits
- Read the logic state in each input or output pin
- Invert or not the bits that have been read



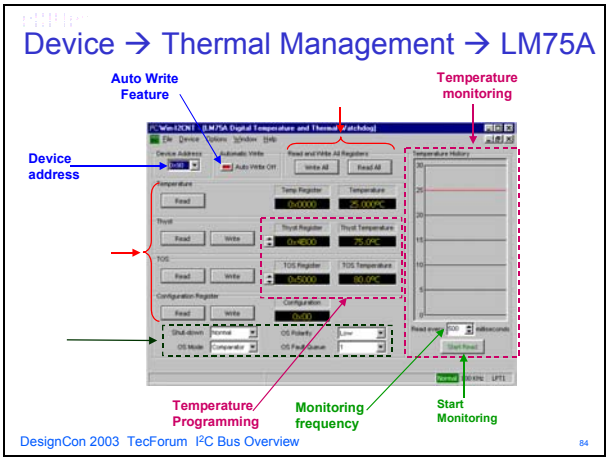
Slide 82

Slide 82 shows the 16 bit true output GPIO.



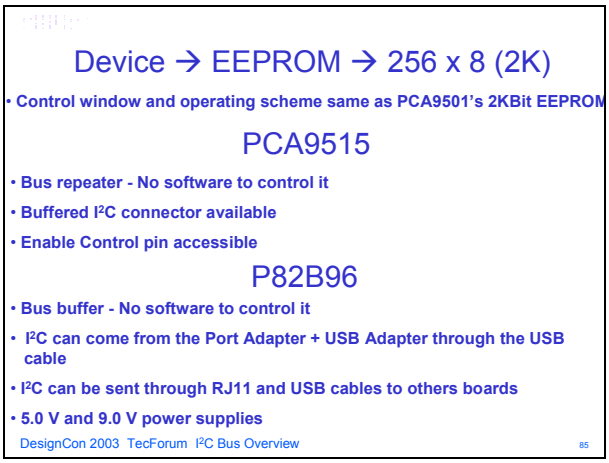
Slide 83

Slide 83 shows the PCA9561 6 bit DIP Switch along with the 4 bit PCA8550 and 6 bit PCA9559/60.



Slide 84

Slide 84 allows control of the LM75A and monitoring of the temperature on the graph.



Slide 85

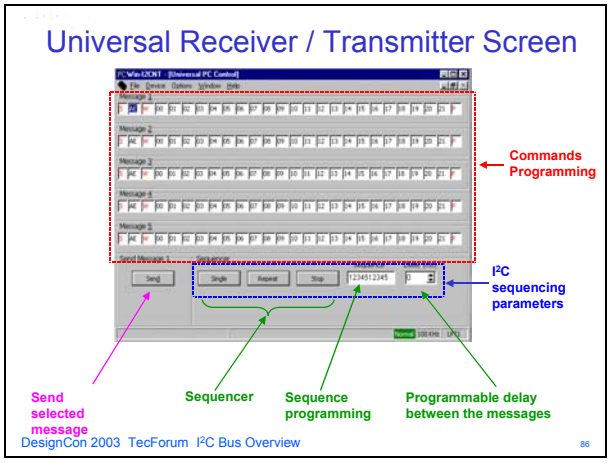
Slide 85 discusses the devices on the I2C 2002-1A evaluation board that is not controlled via the I²C bus. They just provide the possibility to expand/extend the internal 3.3V I²C bus to external devices.

PCA9515 allows connection using short wiring to another 400 pF bus having 3.3-5 V standard I²C chips.

P82B96 allows options to demonstrate:

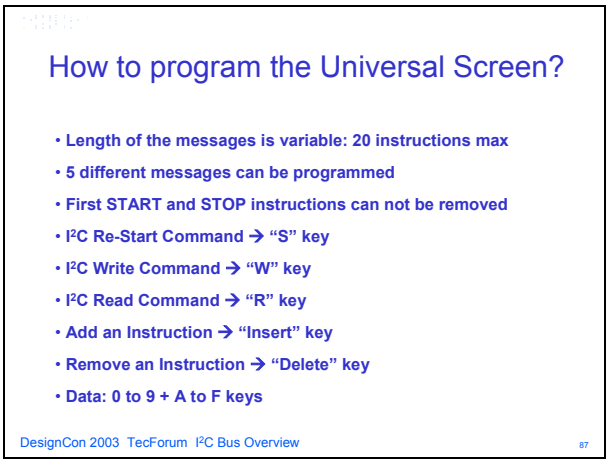
1. Linking to a second evaluation board using a USB cable to provide the power and I²C data link to it.
2. Linking two evaluation boards using a very long telephony cable, say 10 m/33 ft or even more.
3. Linking the evaluation board via a USB cable to the I2CPORT v2 adapter card. It allows a more convenient separation up to 5 m. Just include the USB adapter card.
4. Expanding to another fully standard I²C bus operating at any desired voltage from 2 V to 15 V.

See AN10146-01 for full details.



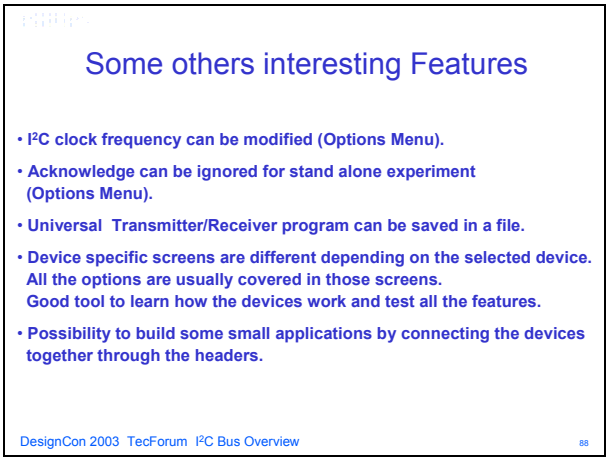
Slide 86

Slide 86 shows the universal screen where I²C command sequence can be used to program any device.



Slide 87

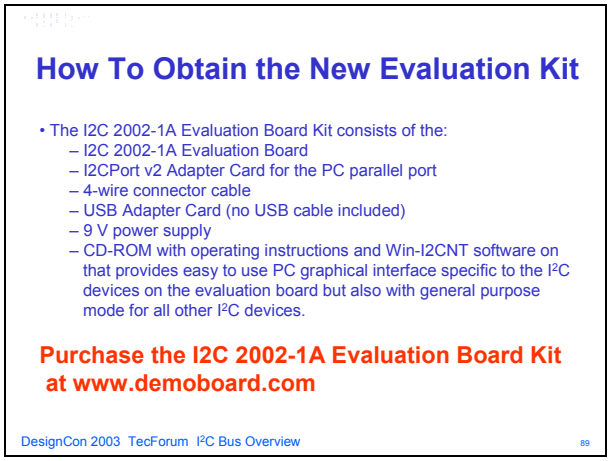
Slide 87 shows how easy it is to program the universal programming screen.



Slide 88

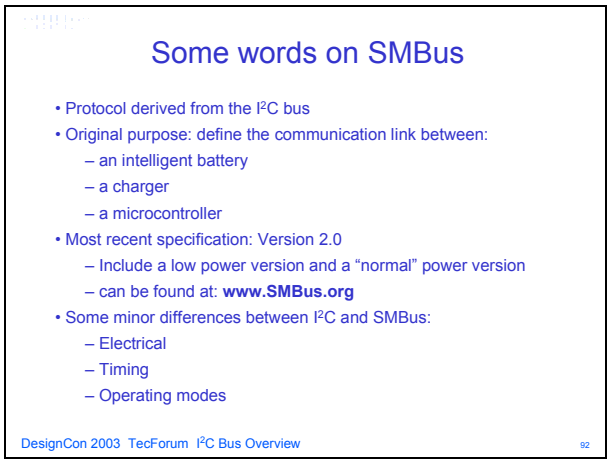
There are many interesting features in the Win-I2CNT system that can help experiment with the new I²C devices.

How to Order the I2C 2002-1A Evaluation Kit



Slide 89

Comparison of I²C with SMBus



Slide 92

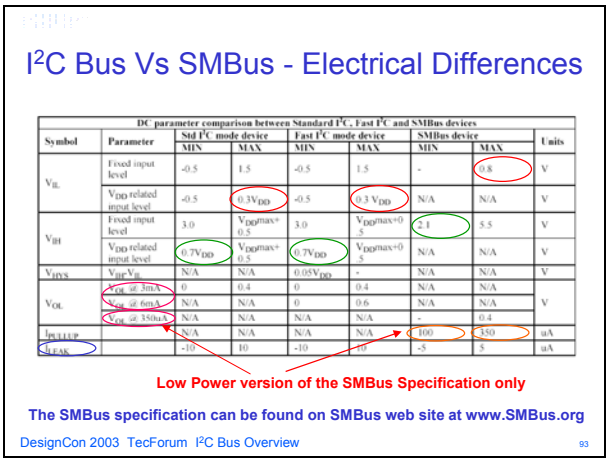
The SMBus uses I²C hardware, and I²C hardware addressing, but adds second-level software for building special systems. In particular its specifications include "Address Resolution Protocol" that can make dynamic address allocations.

"Dynamic reconfiguration: The hardware and software allow bus devices to be "hot-plugged" and used immediately, without restarting the system. The devices are recognized automatically and assigned unique addresses. This advantage results in a plug-and-play user interface." In both those protocols there is a very useful distinction made between a System Host and all the other devices in the system that can have the names, and functions, of masters or slaves.

I²C is also used as the hardware bus with some form of dynamic address assignment in the Optical network module specifications you can find at this website:

http://300pinmsa.org/document/MSA_10G_40G_TRX_I2C_Public_Document_02_19APR02.pdf

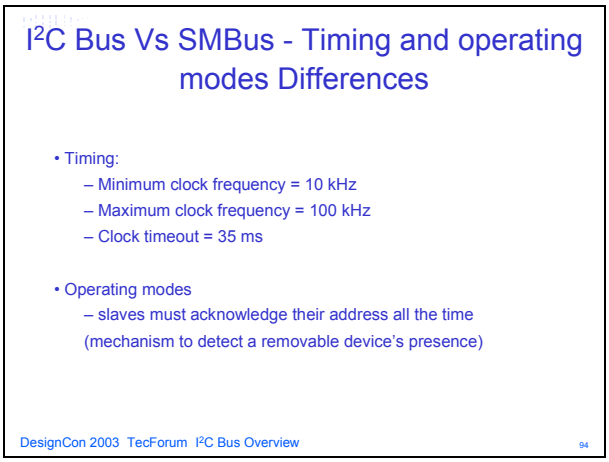
So the idea is to look to any general systems that use dynamic address allocation (even including ones that do not use I²C hardware) to find the software design ideas for building these systems.



Slide 93

SMBus is used today as a system management bus in most PCs. Developed by Intel and others in 1995, it modified some I²C electrical and software characteristics for better compatibility with the quickly decreasing power supply budget of portable equipment.

SMBus also has a "High Power" version 2.0 that includes a "4 mA sink current" version that strictly cannot be driven by I²C chips.



Slide 94

I²C/*SMBus compliancy*

SMBus and I²C protocols are basically the same: A SMBus master will be able to control I²C devices and vice-versa at the protocol level. The SMBus clock is defined from 10 kHz to 100 kHz while I²C can be a DC

bus (0 to 100 kHz, 0 to 400 kHz, 0 to 3.4 MHz). This means that an I²C bus running at a frequency lower than 10 kHz will not be SMBus compliant since the specification does not allow it.

Logic levels are slightly different also: TTL for SMBus: low = 0.8V and high = 2.1V, 30%/70% V_{DD} CMOS level for I²C. This is not a big deal if V_{DD} > 3.0 V. If the I²C device is below 3.0 V then there is a problem since the logic hi/lo levels may not be recognized.

Timeout feature: SMBus has a timeout feature, resetting the devices if a communication takes too long (thus explaining the min clock frequency at 10 kHz). I²C can be a "DC" bus meaning that a slave device stretches the master clock when performing some routine while the master is accessing it. This will notify to the master: "I'm busy right now but I do not want to loose the communication with you, so hold on a little bit and I will let you continue when I'm done" ... and a "little bit" can be an eternity, (at least lower than 10 kHz).

SMBus protocol just assumes that if something takes too long, then it means that there is a problem in the bus and that everybody must reset in order to clear this mode. Slave devices are not then allowed to hold the clock low too long.

Differences SMBus 1.0 and SMBus 2.0

Here is the statement from the SMBus 2.0 document: This specification defines two classes of electrical characteristics, low power and high power. The first class, originally defined in the SMBus 1.0 and 1.1 specifications, was designed primarily with Smart Batteries in mind, but could be used with other low-power devices.

This 2.0 version of the specification introduces an alternative higher power set of electrical characteristics. This class is appropriate for use when higher drive capability is required, for example with SMBus devices on PCI add-in cards and for connecting such cards across the PCI connector between each other and to SMBus devices on the system board.

Devices may be powered by the bus V_{DD} or by another power source, VBus, (as with, for example, Smart Batteries) and will inter-operate as long as they adhere to the SMBus electrical specifications for their class.

Philips devices have a higher power set of electrical characteristics than SMBus1.0.

Main parameter is the current sink capability with Vol = 0.4V.
- SMBus low power = 350 uA
- SMBus high power = 4 mA
- I²C = 3 mA

Philips SMBus “high power” devices are also electrically compatible with I²C specifications but SMBus devices from others may not always be compatible with I²C. Philips I²C devices are electrically compatible with low power SMBus specifications but will not normally conform to all its software features like time-out.

Example for a typical I²C slave device, the PCA9552. It will be SMBus compliant if:
- 10 kHz < Fclock < 100 kHz
- It the device works in a 3.3V or higher environment

Note: the PCA9552 will not be able to reset itself if the bus communication time is higher than the timeout value. That is pretty much the case for all Philips devices. Often the time-out feature can be added for a few cents in discrete hardware. See Slide 57.

Intelligent Platform Management Interface (IPMI)

Intel initiative in conjunction with hp, NEC and Dell and consists of three specifications:
- IPMI for software extensions
- Intelligent Platform Management Bus (IPMB) for intra-chassis (in side the box) extensions
- Inter Chassis Management Bus (ICMB) for inter-chassis (outside of the box) extensions

Needed since as the complexity of systems increase, MTBF decreases. IPMI defines a standardized, abstracted, message-based interface to intelligent platform management hardware are defines standardized records for describing platform management devices and their characteristics. IPMI provides a self monitoring capability increasing reliability of the systems

IPMI
Provides a self monitoring capability increasing reliability of the systems
Monitor server physical health characteristics:
- Temperatures
- Voltages
- Fans
- Chassis intrusion

General system management:
- Automatic alerting
- Automatic system shutdown and re-start
- Remote re-start
- Power control

More information:
www.intel.com/design/servers/ipmi/ipmi.htm

Standardized bus and protocol for extending management control, monitoring, and event delivery within the chassis:

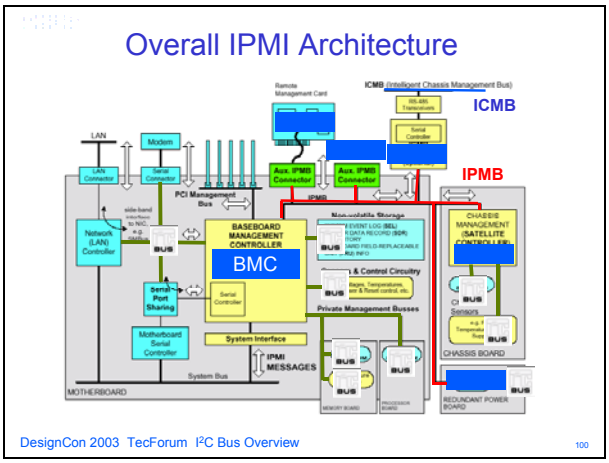
- I²C based
- Multi-master
- Simple Request/Response Protocol
- Uses IPMI Command sets
- Supports non-IPMI devices
- Physically I²C but write only (master capable devices), hot swap not required.
- Enables the Baseboard Management Controller (BMC) to accept IPMI request messages from other management controllers in the system.
- Allows non-intelligent devices as well as management controllers on the bus.
- BMC serves as a controller to give system software access to IPMB

Defines a standardized interface to intelligent platform management

Hardware

- Prediction and early monitoring of hardware failures
- Diagnosis of hardware problems
- Automatic recovery and restoration measures after failure
- Permanent availability management
- Facilitate management and recovery
- Autonomous Management Functions: Monitoring, Event Logging, Platform Inventory, Remote Recovery

Implemented using Autonomous Management Hardware:
Designed for Microcontrollers based implementations
Hardware implementation is isolated from software implementation
New sensors and events can then be added without any software changes



Slide 100

Where IPMI is being used

Intel Server Management

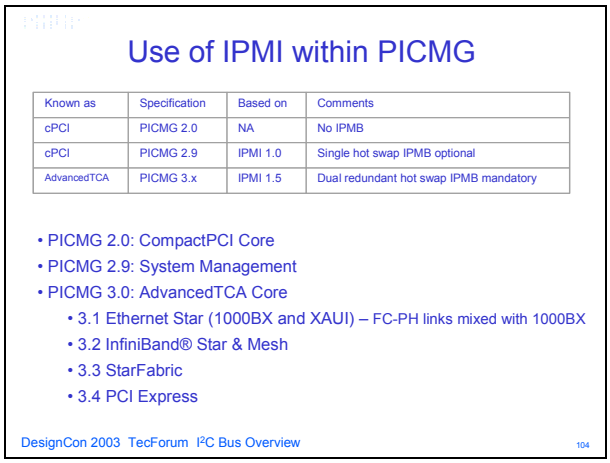
Servers today run mission-critical applications. There is literally no time for downtime. That is why Intel created Intel® Server Management – a set of hardware and software technologies built right into most Intel® sever boards that monitors and diagnoses server health. Intel Server Management helps give you and your customers more server uptime, increased peace of mind, lower support costs, and new revenue opportunities.

More information:
http://program.intel.com/shared/products/servers/boards/server_management

PICMG

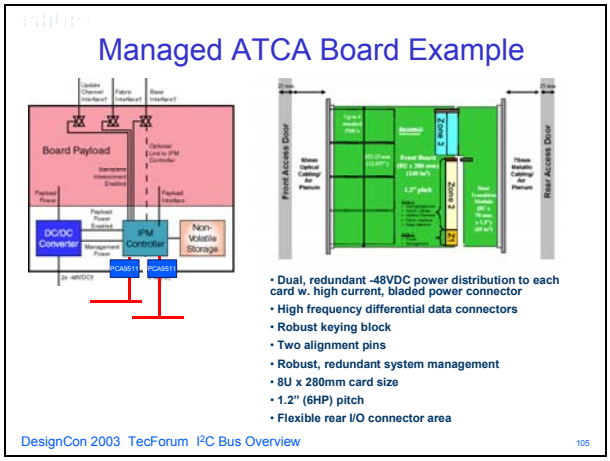
PICMG (PCI Industrial Computer Manufacturers Group) is a consortium of over 600 companies who collaboratively develop open specifications for high performance telecommunications and industrial computing applications. PICMG specifications include CompactPCI® for Eurocard, rack mount applications and PCI/ISA for passive backplane, standard format cards. Recently, PICMG announced it was beginning development of a new series of specifications, called AdvancedTCA™, for next-generation telecommunications equipment, with a new form factor and based on switched fabric architectures.

More information can be found at:
http://www.picmg.org



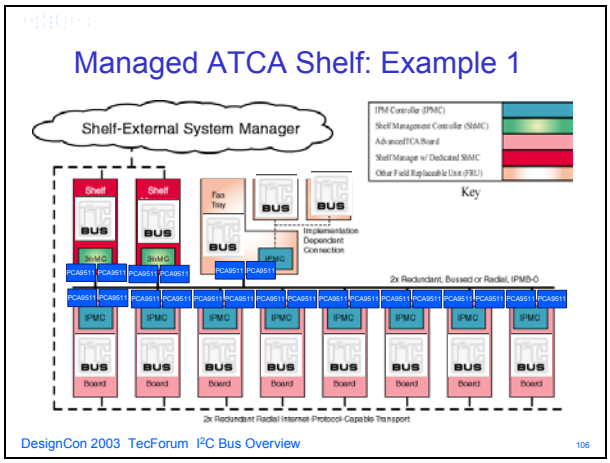
Slide 104

IPMI with additional extension is used as the basis for PICMG 2.9 and PICMG 3.x.



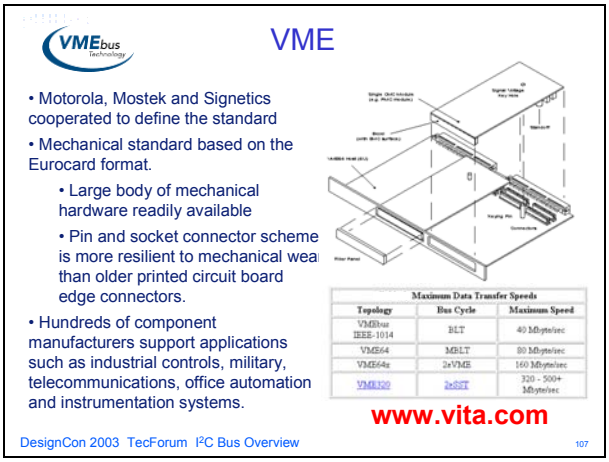
Slide 105

Slide 105 shows how IPMI is used within an AdvancedTCA card.



Slide 106

Slide 106 shows one of the two redundant buses that would interface through the PCA9511 or PCA9512/13/14.



Slide 107

VMEbus

VMEbus is a computer architecture. The term 'VME' stands for VERSAmodule Eurocard and was first coined in 1980 by the group of manufacturers who defined it. This group was composed of people from Motorola, Mostek and Signetics corporations who were cooperating to define the standard. The term 'bus' is a generic term describing a computer data path, hence the name VMEbus. Actually, the origin of the term 'VME' has never been formally defined. Other widely used definitions are VERSAbus-E, VERSAmodule Europe and VERSAmodule European. However, the term 'Eurocard' tends to fit better, as VMEbus was originally a combination of the VERSAbus electrical standard, and the Eurocard mechanical form factor.

VERSAbus was first defined by Motorola Corporation in 1979 for its 68000 microprocessor. Initially, it competed with other buses such as Multibus™, STD Bus, S-100 and Q-bus. However, it is rarely used anymore. The microcomputer bus industry began with the advent of the microprocessor, and in 1980 many buses were showing their age. Most worked well with only one or two types of microprocessors, had a small addressing range and were rather slow. The VMEbus architects were charged with defining a new bus that would be microprocessor independent, easily upgraded from 16 to 32-bit data paths, implement a reliable mechanical standard and allow independent vendors to build compatible products. No proprietary rights were assigned to the new bus, which helped stimulate third party product development. Anyone can make VMEbus products without any royalty fees or licenses. Since much work was already done on VERSAbus it was used as a framework for the new standard.

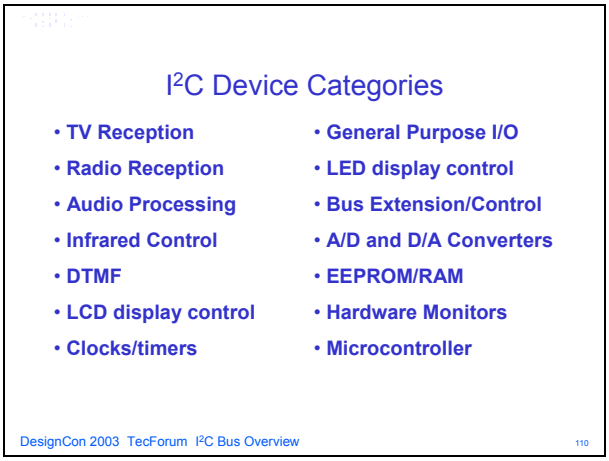
In addition, a mechanical standard based on the Eurocard format was chosen. Eurocard is a term that loosely describes a family of products based around the DIN 41612 and IEC 603-2 connector standards, the IEEE 1101 PC board standards and the DIN 41494 and IEC 297-3 rack standards. When VMEbus was first developed, the Eurocard format had been well established in Europe for several years. A large body of mechanical hardware such as card cages, connectors and sub-racks were readily available. The pin and socket connector scheme is more resilient to mechanical wear than older printed circuit board edge connectors.

The marriage of the VERSAbus electrical specification and the Eurocard format resulted in VMEbus Revision A. It was released in 1981. The VMEbus specification has since been refined through revisions B, C, C.1, IEC 821, IEEE 1014-1987 and ANSI/VITA 1-1994. The ANSI, VITA, IEC and IEEE standards are important because they make VMEbus a publicly defined specification. Since no proprietary rights are assigned to it, vendors and users need not worry that their products will become obsolete at the whim of any single manufacturer. Since its introduction, VMEbus has generated thousands of products and attracted hundreds of manufacturers of boards, mechanical hardware, software and bus interface chips. It continues to grow and support diverse applications such as industrial controls, military, telecommunications, office automation and instrumentation systems.

Use of IPMI in VME Architecture

New VME draft standard indirectly calls for IPMI over I²C for the system management protocol since there was nothing to be gained by reinventing a different form of system management for VME. The only change from the PICMG 2.9 system management specification is to redefine the backplane pins used for the I²C bus and to redefine the capacitance that a VME board can present on the I²C bus. The pin change was required because the VME backplane connectors are different from cPCI. The capacitance change was required because cPCI can have a maximum of 8 slots and VME can have a maximum of 21 slots. System Management for VME Draft Standard VITA 38 – 200x Draft 0.5 9 May 02 draft at:
http://www.vita.com/vso/draftstd/vita38.d0.5.pdf
provides more information.

I²C Device Overview

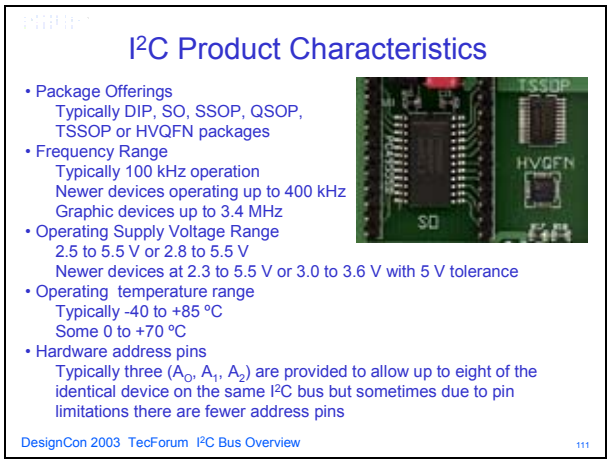


Slide 110

I²C devices can be broken down into different categories.

- TV reception: Provides TV tuning and reception
- Radio reception: Provides radio tuning and reception
- Audio Processing
- Infra-Red control
- DTMF: Dual Tone Multiple Frequency
- LCD display control: Provides power to segments of an LCD that are controlled via I²C bus
- LED display control: Provides power to segments of an LED that are controlled via I²C bus
- Real time clocks and event counters: counting the passage of time, chronometer, periodic alarms for safety applications, system energy conservation, time and date stamp for point of sales terminals or bank machines.
- General Purpose Digital Input/Output (I/O): monitoring of 'YES' or 'NO' information, such as whether or not a switch is closed or a tank overflows; or controlling a contact, turning on an LED, turning off a relay, starting or stopping a motor, or reading a digital number presented at the port (via a DIP switch, for example).
- Bus Extension/Control: expends the I²C bus beyond the 400 pF limit, allows different voltage devices on the same I²C bus or allows devices with the same I²C address to be selectively addressed on the I²C bus.
- Analog/digital conversion: measurement of the size of a physical quantity (temperature, pressure…), proportional control; transformation of physical analog values into numerical values for calculation.
- Digital/analog conversion: creation of particular control voltages to control DC motors or LCD contrast.
- RAM: Random Access Memory

- EEPROM: Electrically Erasable Programmable Read Only Memory, retains digital information even when powered down
- Hardware Monitors: monitoring of the temperature and voltage of systems
- Microprocessors: Provides the brains behind the I²C bus operation.



Slide 111

The frequency range of most of the newer I²C devices is up to 400 kHz and we are moving to 3.4 MHz for future devices where typical uses would be in consumer electronics where a DSP is the master and the designer wants to rapidly send out the I²C information and then move on to other processing needs.

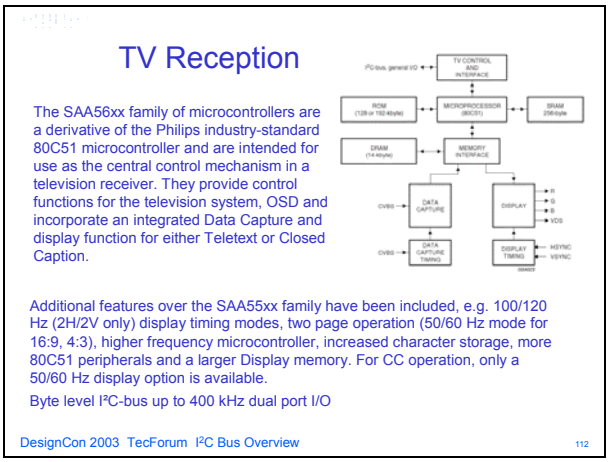
The operating range of most of the newer CMOS devices is 2.3 to 5 V to allow operation at the 2.5, 3.3 and 5V nodes. Some processes restrict the voltage range to the 3.3 V node. Most customers have moved from 5 V and are now at 3.3 V but several are moving rapidly to 2.5 V and even 1.8 V in the near future. We are working on next generation general purpose devices to support 1.8 V operation and currently have some LCD display drivers that operate down to 1 V.

The operating temperature range is typically specified at the industrial temperature range but again depending on process or application, the range may be specified higher or lower. The automotive, military and aviation industries have expressed more interest in I²C devices due to the low cost and simplicity of operation so future devices temperature ranges may be expanded to meet their needs.

I²C devices were typically offered in either DIP or SO and limited their use in equipment where space is at a premium. Newer I²C devices are typically offered in SO, TSSOP or near chip scale HVQFN packages.

I²C devices are designed in the process that allows best electrical and ESD performance and are manufactured in Philips or third party fabs through out the world. Philips has taken the initiative to offer the same process in multiple internal fabs to provide redundancy and continuation of supply in any market condition.

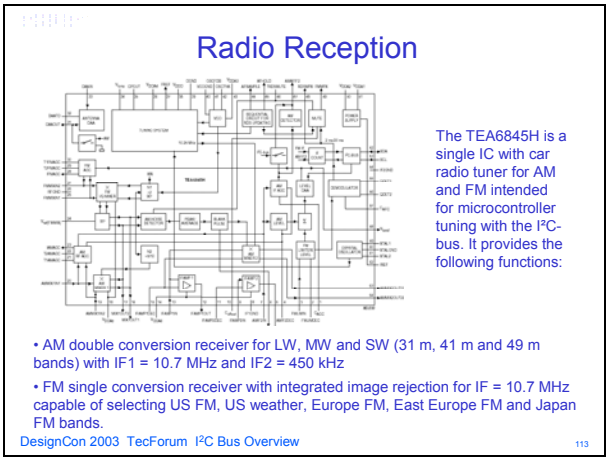
TV Reception



Slide 112

The I²C bus is used as a means to easily move control or status information on and off the devices. The SA56xx is given as an example of this type of device.

Radio Reception

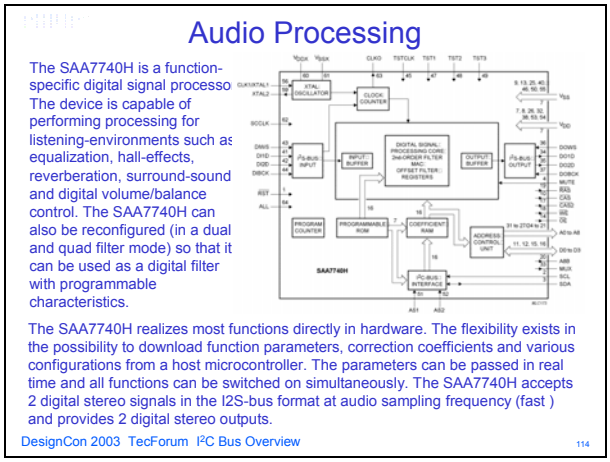


Slide 113

Again, the I²C bus is used to control frequency selection or control the audio sound control and interface with the microcontroller. Special software programs, applied by connecting to the I²C bus during factory testing, automatically perform the alignment of the RF sections of the receiver, eliminating the need for manual or mechanical adjustments. The alignment information will be stored in some non-volatile memory

chip and re-sent to the receiver chip, where it is stored in R.A.M., each time power is applied to the receiver.

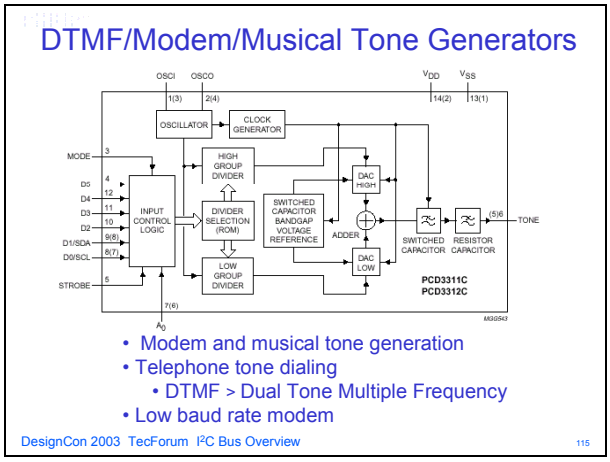
Audio Processing



Slide 114

The I²C bus is used to control the audio and sound balance.

Dual Tone Multi-Frequency (DTMF)



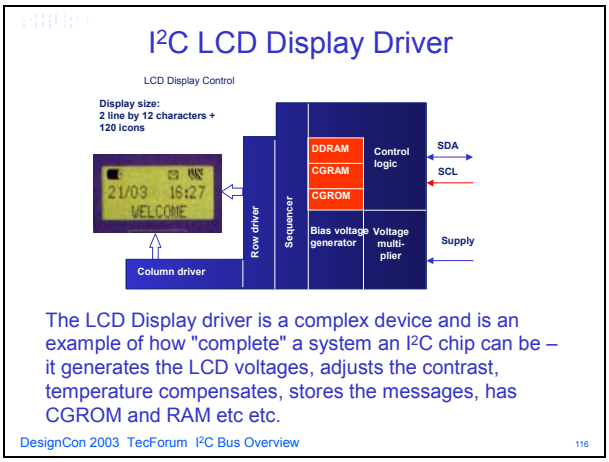
Slide 115

The PCD3311C and PCD3312C are single-chip silicon gate CMOS integrated circuits. They are intended principally for use in telephone sets to provide the dual-tone multi-frequency (DTMF) combinations required for tone dialing systems. The various audio output frequencies are generated from an on-chip 3.58 MHz quartz crystal-controlled oscillator. A separate crystal is used, and a separate microcontroller is required to control the devices.

Both the devices can interface to I²C bus compatible microcontrollers for serial input. The PCD3311C can also interface directly to all standard microcontrollers, accepting a binary coded parallel input. With their on-chip voltage reference the PCD3311C and PCD3312C provide constant output amplitudes that are independent of the operating supply voltage and ambient temperature. An on-chip filtering system assures a very low total harmonic distortion in accordance with CEPT recommendations. In addition to the standard DTMF frequencies the devices can also provide:

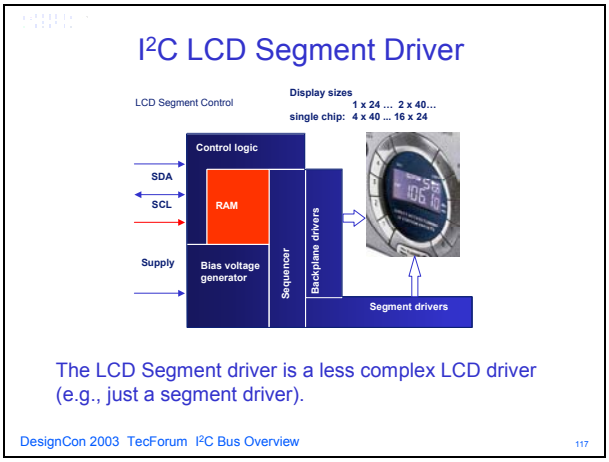
- Twelve standard frequencies used in simplex modem applications for data rates from 300 to 1200 bits per second
- Two octaves of musical scales in steps of semitones.

LCD Display Driver



Slide 116

The LCD display driver is a complex LCD driver and is an example of how "complete" a system an I²C chip can be - generates the LCD voltages, adjusts the contrast, temperature compensates, stores the messages, has CGROM and RAM etc.



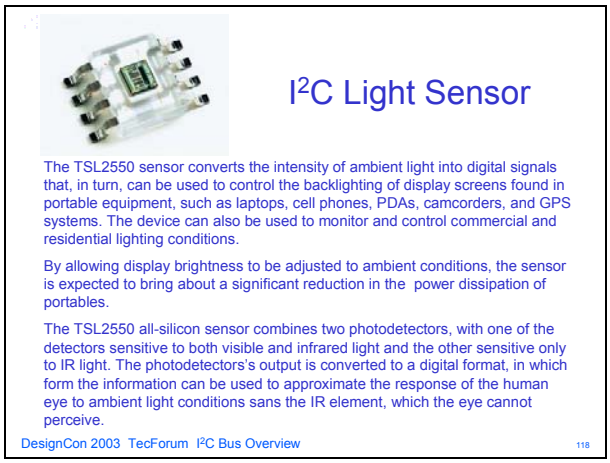
Slide 117

The LCD segment driver is a less complex LCD driver (e.g., just a segment driver). Philips focus is for large

volume consumer display apps, which is right now B&W and color STN LCD displays and in near future it will be TFT and OLED (organic LED displays). The OLED drivers will most probably not be useable with conventional LEDs.

VGA is beyond our current roadmap that stretches only up to about 1/4 VGA. This is simply because of the requirements that we see in the mobile telecomm market, our main focus. We find already that I²C does not give us enough transmission rate for display data so serial bus is mainly intended for control and text overlay signals in such displays.

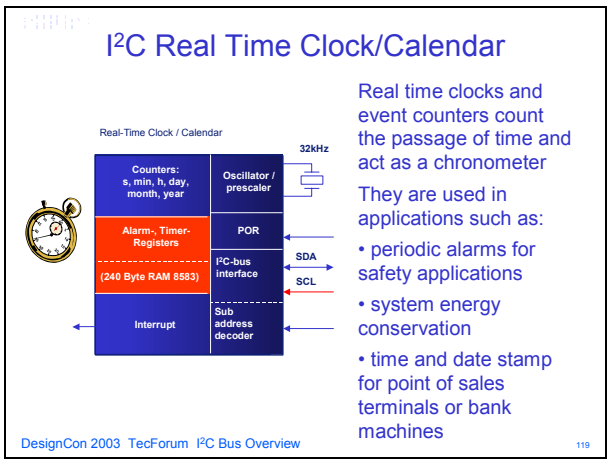
Light Sensor



Slide 118

Slide 118 shows a new innovation in light detectors that uses the I²C bus to transfer information to and from the sensor.

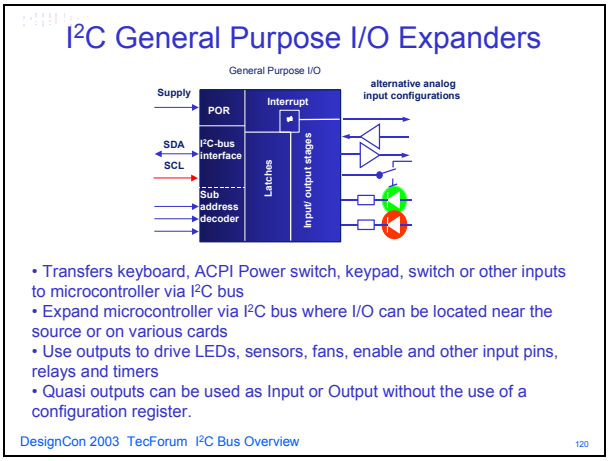
Real Time Clock/Calendar



Slide 119

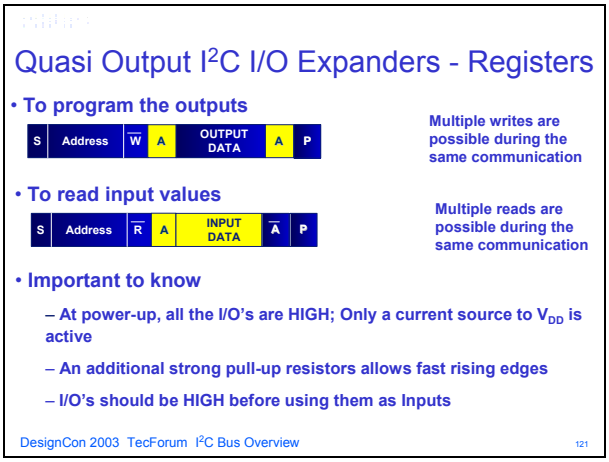
Philips offers four RTCs, these are PCF8593, PCF8583, PCF8573 and PCF8563. The PCF8563 is the most recently developed and is technically the most advanced. The RTCs have one interrupt output and do not track the exact year. This must be done in software by the customer. They do use a 4-year calendar base and can count 255 years. PCF8583 has the added advantage of 240 bytes of RAM integrated with the RTC. This could be important if such small RAM is required then we replace two chips with one.

General Purpose I/O Expanders



Slide 120

Let's talk about some of the newer devices, such as these new general-purpose input and output (GPIO) expansion for the I²C/SMBus.

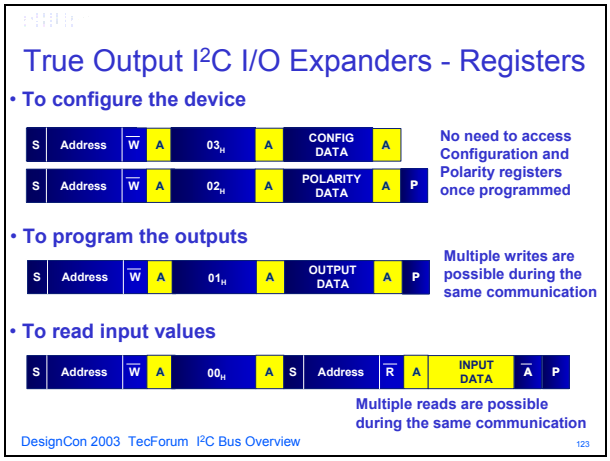


Slide 121

The PCF8574 and PCA8575 are well known general purpose I/O expanders. The PCA9500 is a combination of the PCF8574 with a 2K EEPROM. The interrupt pin is replaced by the EEPROM write protect (WP). The EEPROM has a different fixed I²C address then the GPIO. The PCA9501 is a combination of the PCF8574 with a 2K EEPROM. The device is offered in a 20-pin TSSOP package and the four extra pins allow the

interrupt output to be included in addition to the WP. The extra three pins are then used to offer a total of six address pins allowing up to 64 of these devices to share the same I²C bus. The PTN devices are design for telecom maintenance and control applications.

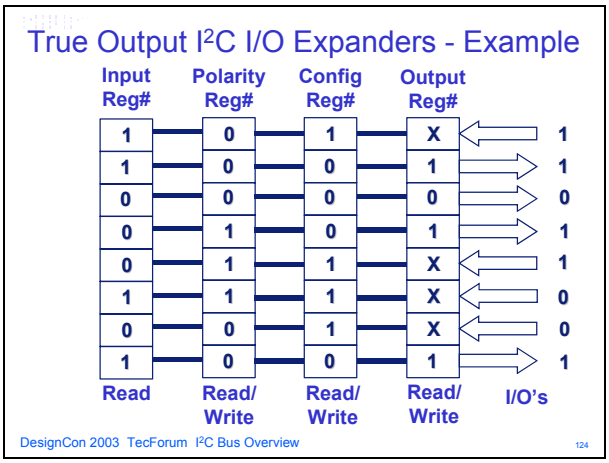
The PCA9558 is a combination of the PCA9557 with a 2K EEPROM and 5-bit DIP Switch.



Slide 123

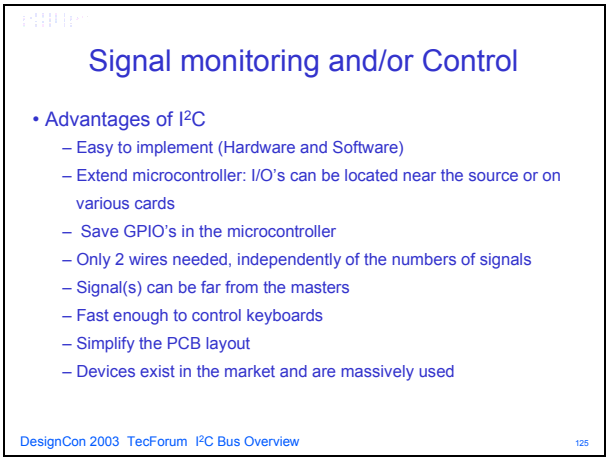
These newer device's true outputs provide active source and sink current sources and does not rely upon a pull up resistor to provide the source current. The four sets of registers within the true outputs devices are programmable and provide for: Configuration (Input or Output) control, Input (value), Output (value) or Polarity (active high or low).

The PCA9554/54A/55 devices have an interrupt output and the 8 or 16 I/O pins can be configured for interrupt inputs. These newly released devices have the same I²C address and footprint as the PCF8574/74A/75 but require some software modifications due to the different I/O registers. The PCA9554 and PCA9555 have the same I²C address while the PCA9554A has a slightly different fixed address allowing 16 devices (eight 54A and eight 54/55 in any combination) to be on the same I²C/SMBus. The PCA9556/57 feature a Hardware Reset pin instead of the interrupt output that allows the device to be reset remotely should the I²C bus become hung up. The PCA9557 is an improved version of the PCA9556 that has the electrical characteristics of the PCA9554/54A. Information on GPIO selection is contained within application note AN469.



Slide 124

Slide 124 shows an example of how the PCA9554/54A/57 is programmable.



Slide 125

Signal Monitoring and/or Control first approach is to use GPIO's of the master(s) controlling the application. In some applications, use of these GPIO's is not the best approach.

Reasons can be the following:

- Number of signals to monitor/control is too important and requires a big amount of the master's GPIO's.
- Signals can be in a remote location implying a more complex PCB layout, with a lot of long traces (making the design more sensitive to noise)
- Upgrade (more signals to monitor/control) requires a total re-layout of the PCB and is limited to the number of GPIO's still available in the master.

Signal monitoring and/or Control

• Proposed devices

# of Outputs	Interrupt and POR	POR and 2K EEPROM	Interrupt, POR and 2K EEPROM
Quasi Output (20-25 ma sink and 100 uA source)			
8	PCF8574/74A	PCA9500/58	PCA9501
16	PCF8575/75C	-	-

# of Outputs	Reset and POR	Interrupt and POR
True Output (20-25 ma sink and 10 mA source)		
8	PCA9556/57	PCA9534/54/54A
16	-	PCA9535/55

• Advantages
- Number of I/O scalable
- Programmable I²C address allowing more than one device in the bus
- Interrupt output to monitor changes in the inputs
- Software controlling the device(s) easy to implement

DesignCon 2003 TecForum I²C Bus Overview 126

Slide 126

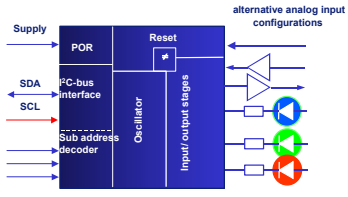
The I²C GPIO device approach provides an elegant solution with minimum hardware and software changes:

- The device(s) can be plugged to an existing I²C bus in the application
- Minor software change is required to control the new device(s)
- Easily upgradeable (by adding more I²C GPIO devices)
- Remote signals can be easily controlled (requires only a longer I²C bus trace - 2 wires only)
- Changes in the monitored input signals can be propagated to the master through a single Interrupt line. The master can be easily interrogate the I²C GPIO to determine which input(s) generated the Interrupt

See Application Note AN469 for more information on GPIOs.

LED Dimmers and Blinkers

I²C LED Dimmers and Blinkers



• I²C/SMBus is not tied up by sending repeated transmissions to turn LEDs on and then off to "blink" LEDs.
• Frees up the micro's timer
• Continues to blink LEDs even when no longer connected to bus master
• Can be used to cycle relays and timers
• Higher frequency rate allows LEDs to be dimmed by varying the duty cycle for Red/Green/Blue color mixing applications.

DesignCon 2003 TecForum I²C Bus Overview 127

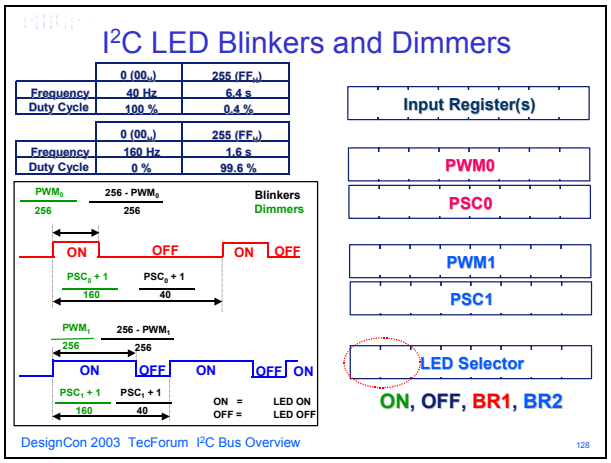
Slide 127

These new devices are useful for LED driving and blinking. The I²C/SMBus or the micro controller is not tied up by sending repeated transmissions to blink LEDs as is currently done when a GPIO is used. The PCA9530/31/32/33 and the PCA9550/51/52/53 provide the same amount of electrical sink capability as the PCA9554/55/57 but have a built in oscillator and two I²C programmable blink rates.

Two user definable blink rates and duty cycles programmed via the I²C/SMBus. These are programmed during the initial set up and can range between 160 Hz and every 1.6 seconds for the LED Dimmers and between 40 Hz and every 6.4 seconds for the LED Blinkers. Thereafter only a single transmission is required to turn individual LEDs: on, off or blink at one of the two programmable blink rates. The duty cycle can be used to 'dim' the LEDs using the LED Dimmers by setting the blink rate to 160 Hz (faster than the eye can see the blinking) and then changing the average current through the LED by changing the duty cycle.

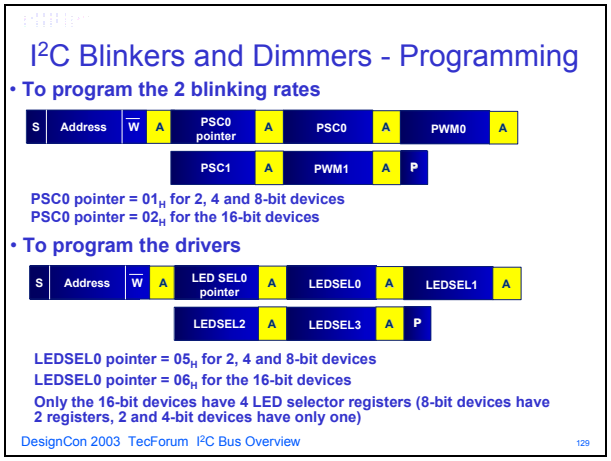
The internal oscillator is regulated to +/- 10% accuracy and no external components are required. The +/- 10% tolerance was recommended by human factor engineers. These devices allow you to program two specific blink rates and then command a LED to blink at one of these rates without sending any further I²C commands. If you use normal GPIOs to blink LEDs, you must send an ON command followed by an OFF command followed by an ON command for the duration of the blink. This is OK if you do not have many LEDs to blink or much traffic on the I²C bus, or have microcontroller overhead to burn, but if you do this for many LEDs you will tie up the I²C bus and your micro controller. Hence the need for dedicated LED blinkers as a stand alone part option. Unused pins can be used as normal GP input or output, but since they are open-drain, a pull up resistor will be needed for logic high outputs.

A Hardware Reset pin is included, allowing the LED blinker to be reset independently from the rest of the I²C/SMBus or higher level system. Each open drain output can sink 25 mA of current with total package sinking capacity limited to 100 mA for the 2, 4 and 8 bit devices and 200 mA for the 16 bit device (100 mA for each byte). Typical LEDs take 10-25 mA of current when in operation.



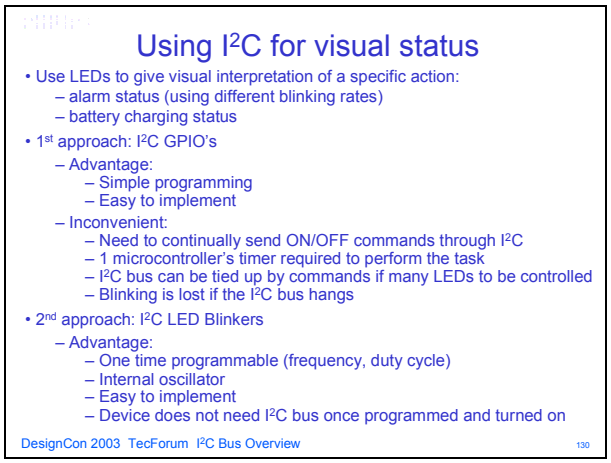
Slide 128

Slide 128 shows the register configuration of the LED Blinkers and Dimmers.



Slide 129

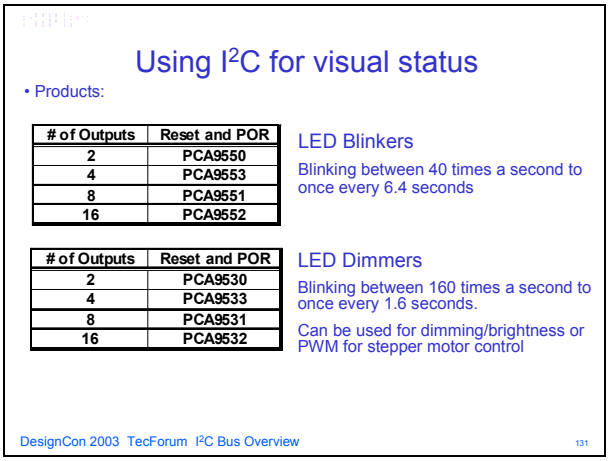
Slide 129 shows the programming sequence for the LED Dimmers and Blinkers.



Slide 130

I²C GPIO's can be used to control LEDs in order to visual status, like for example blink slowly when in normal condition, blink faster in an alarm mode. The main disadvantages of this method are the following:

- ON/OFF commands need to be sent all the time by the master
- I²C bus can be tied by sending the ON/OFF commands when a lot of LEDs needs to be controlled
- At least one timer in the master needs to be dedicated for this purpose
- Blinking is lost if the I²C bus hangs or if the master fails



Slide 131

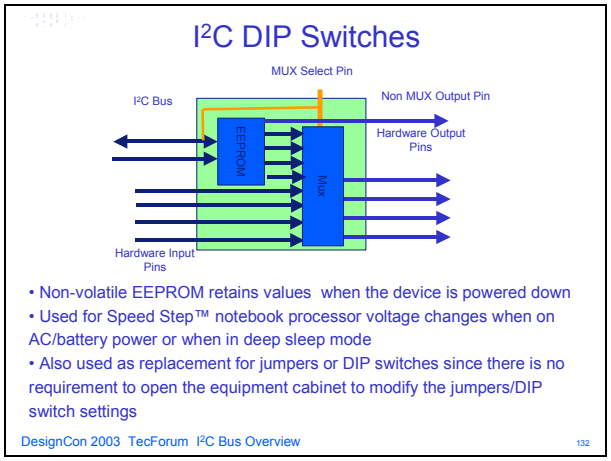
I²C LED blinkers provide an elegant autonomous solution:

- They have an built-in accurate oscillator requiring no external components
- They can be programmed in one I²C access (2 selectable fully programmable blinking rates)
- Output state (Blinking rate 1, Blinking rate 2, Permanently ON, Permanently OFF) is programmed in one I²C access anytime.

Blinking is not lost, once the device is programmed, in case the bus hangs or the master fails.

See Application Note AN264 for more information on the LED Dimmers/Blinkers.

DIP Switch



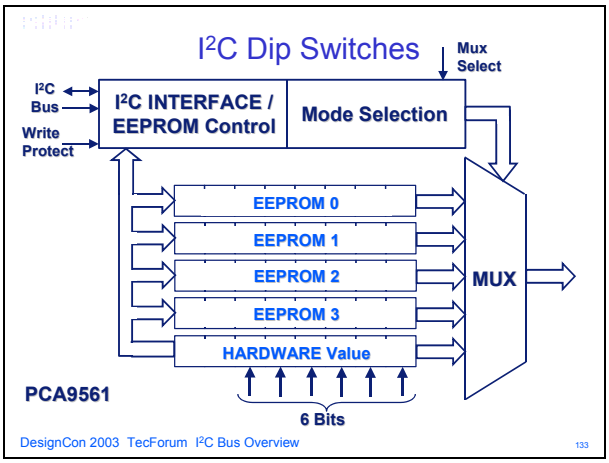
Slide 132

These devices were designed for use with Intel® processors to implement the Speed Step™ technology for notebook computers (selects different processor voltages when connected to AC power, the battery or in a deep sleep/deeper sleep mode), Dual BIOS selection (select different operating systems during start-up).

Designers have however found other uses for these devices such as; VGA/Tuner cards (select the appropriate transmission standard), in inkjet printers and are being used as replacement for jumpers or dip switches since the I²C controlled integrated EEPROM and Multiplexer eliminates the need to open equipment to modify the settings by hand, making it easier to change settings and less likely to damage the equipment.

I²C commands and/or hardware pins are used to select between the default values or the setting programmed from the I²C bus and stored in the onboard I²C EEPROM register. These onboard values can be changed at any time via the I²C bus. The non-volatile I²C EEPROM register values stay resident even when the device is powered down. The devices power up with either the hardware pin inputs or the EEPROM0 register retained value on the hardware output pins depending on the position (H or L) of the Mux select pins.

The PCA9560 is footprint identical to the PCA9559 but has two internal EEPROM registers to allow for three preprogrammed setting (e.g., AC power/battery power, deep sleep or deeper sleep mode).

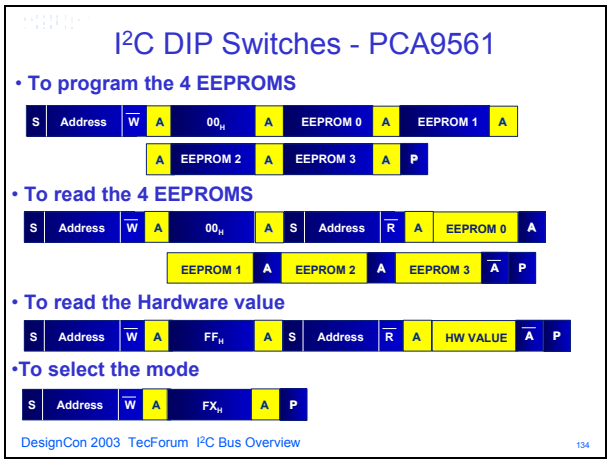


Slide 133

The PCA9561 shown in Slide 133 is unique in that it has 6 hardware input pins and four internal 6-bit EEPROM registers. Output selection is possible between any one of these five 6-bit values at any time via the I²C bus. The EEPROMs have a 10 year memory retention and are rated for 3000 write cycles in the data sheet but have been tested to 50,000 cycles with no failures.

The hardware pins may not be used at all or may be used for a default manufacturing address. At manufacturing, the I²C address of the targeted device may be the one given by the default EEPROM values (all Zero's). If the customer wants to change the I²C address, he has to Address the Multiplexed/Latched EEPROM device (PCA8550, PCA9559, PCA9560 or PCA9561) and program the EEPROM to the new value they want.

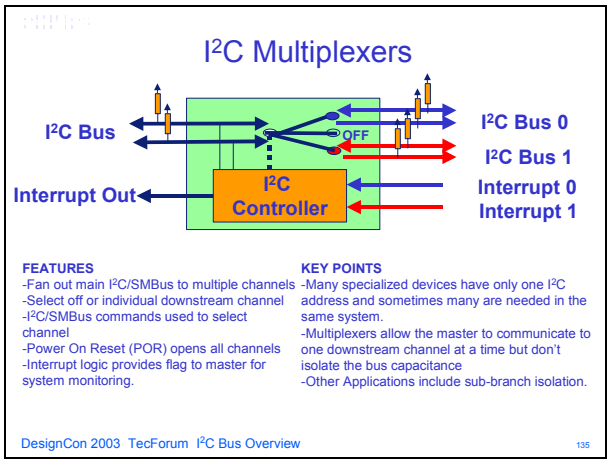
If they use the PCA9560 or PCA9561, 2 or 4 different values can be already pre-programmed. Put the right logic level(s) on the Mux_select pin(s) if necessary (to select the EEPROM values at the Mux input and propagate them to the outputs (connected to the Address pins of the targeted I²C device). Address the targeted I²C device (programmed with the new I²C address). Nice thing about using Multiplexed/Latched EEPROM is that the configuration is not lost each time supply is powered down.



Slide 134

Side 134 shows the typical program sequence for the PCA9561. See Application Note AN250 for more information on the DIP Switches.

Multiplexers and Switches



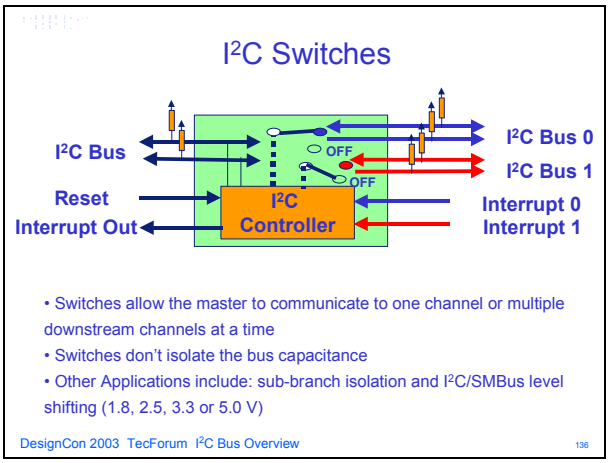
Slide 135

The multiplexer allows multiplexing multiple I²C devices with the same I²C address. The I²C SCL/SDA upstream channel to fan out to multiple SCx/SDx channels that are selected by the programmable control register. The I²C command is sent via the main I²C bus and is used to select or deselect the downstream channels. The Multiplexers can select none or only one SCx/SDx channels at a time since they were designed primarily for address conflict resolution such as when multiple devices with the same I²C address need to be attached to the same I²C bus and you can only talk to one of the devices at a time.

These devices are used in video projectors and server applications. Other applications include:
Address conflict resolution (e.g., SPD EEPROMs on DIMMs).

- I²C sub-branch isolation
- I²C bus level shifting (e.g., each individual SCx/SDx channel can be operated at 1.8 V, 2.5 V, 3.3 V or 5.0 V if the device is powered at 2.5 V).

Interrupt logic inputs for each channel and a combined output are included on every multiplexer and provide a flag to the master for system monitoring. These devices do not isolate the capacitive loading on either side of the device so the designer must take into account all trace and device capacitance on both sides of the device (any active channels). Pull up resistors must be used on all channels.



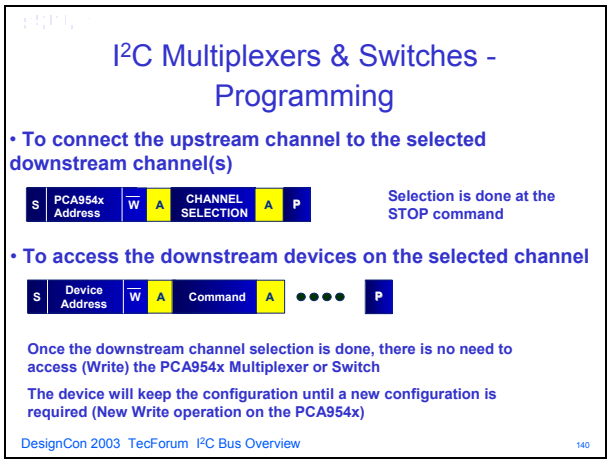
Slide 136

The Switches allow multiplexing but also allow multiple downstream channels to be active at the same time that allows voltage level translation or load sharing applications. The I²C SCL/SDA upstream channel to fan out to multiple SCx/SDx channels that are selected by the programmable control register. The Switches can select individual SCx/SDx channels one at a time, all at once or in any combination through I²C commands and very primary designed for sub-branch isolation and level shifting but also work fine for address conflict resolution (Just make sure you do not select two channels at the same time). Applications are the same as for the multiplexers but since multiple channels can be selected at the same time the switches are really great for I²C bus level shifting (e.g., individual SCx/SDx channels at 1.8 V, 2.5 V, 3.3 V or 5.0 V if the device is powered at 2.5 V).

A hardware reset pin has been added to all the switches. It provides a means of resetting the bus should it hang up, without rebooting the entire system and is very useful in server applications where it is impractical to reset the entire system when the I²C bus hangs up. The switches reset to no channels selected.

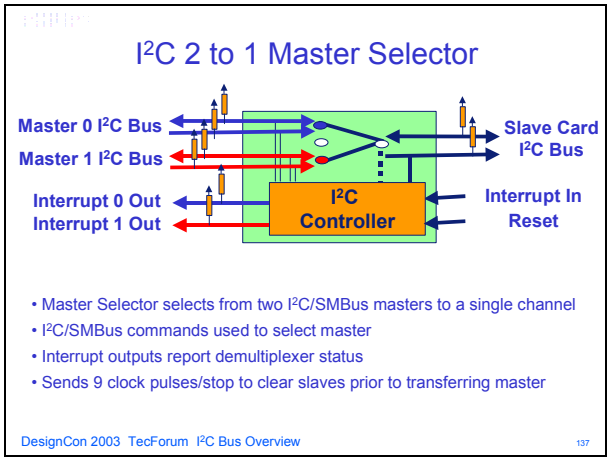
Interrupt logic inputs and output are available on the PCA9543 and PCA9545 and provide a flag to the master for system monitoring. The PCA9546 is a lower cost version of the PCA9545 without Interrupt Logic. The PCA9548 provides eight channels and are more convenient to use then dual 4 channel devices since the device address does not have to shift.

These devices do not isolate the capacitive loading on either side of the device so the designer must take into account all trace and device capacitance on both sides of the device (active channels only). Pull up resistors must be used on all channels.



Slide 137

Slide 137 shows a typical programming sequence. See Application Note AN262 for more information on the switch/multiplexers.



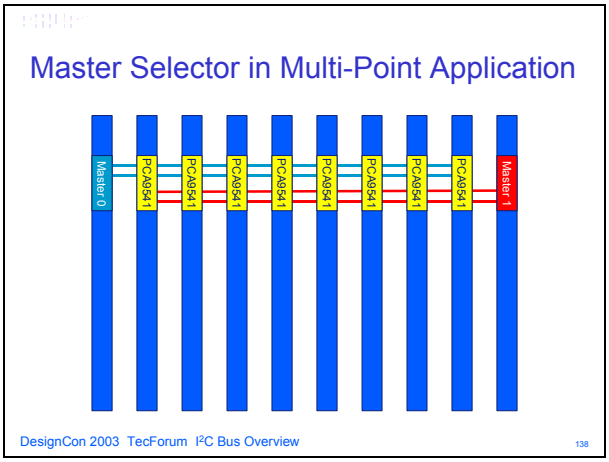
Slide 138

The PCA9541 is designed for applications where there are two bus masters controlling the same slaves and the masters need to be isolated for redundancy.

The PCA9541/01 defaults to channel 0 on start up/reset. The device was designed for a company that wanted the device to connect master 0 to shared resources at start up so they wouldn't have to send any commands.

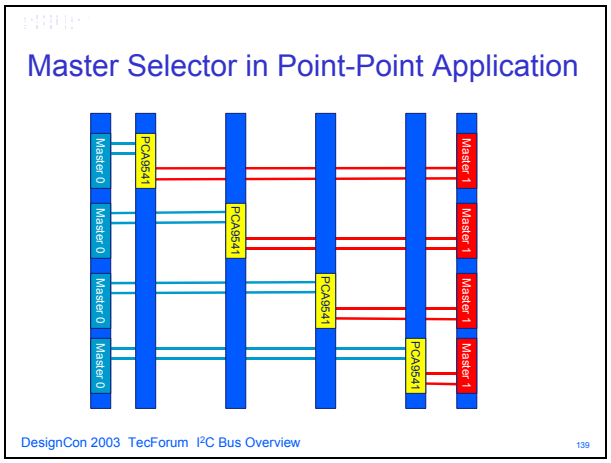
The PCA9541/02 defaults to channel 0 on start up/reset only after it has seen a stop command on bus 0. This is our hot swap version, a requirement the company using the PCA9541/01 didn't have (since they power down the system before cards are inserted or removed). This feature on the PCA9541/02 allows you to insert and remove cards without confusing the slave devices on the card by them being caught midway into an I²C transmission if there is an active transmission on the backplane/main bus.

The PCA9541/03 defaults to no channels selected on start up/reset and one of the masters needs to command the PCA9541/03 to select bus 0 or 1. We had some customers interested in not connecting any bus until the master was ready. This feature also allows the PCA9541/03 to be used as a 'gatekeeper" multiplexer as described in the data sheet specific applications section.



Slide 139

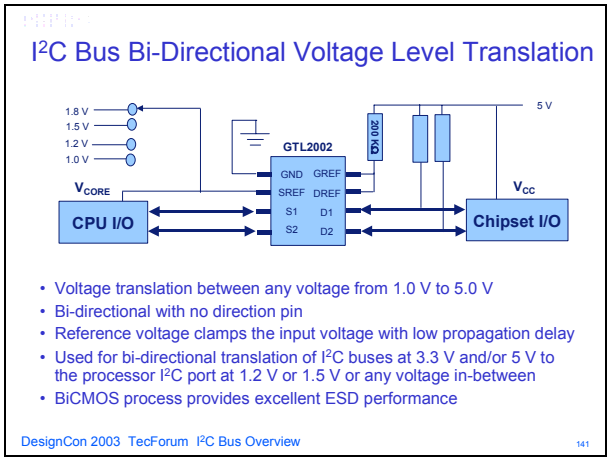
PCA9541 in a multi-point application were all cards use the same two buses. Master 0 is the primary master and master 1 is the back up master.



Slide 140

PCA9541 in a point to point application where there are two dedicated buses to each slave card for even higher redundancy, such as a bent pin would not disable all the cards.

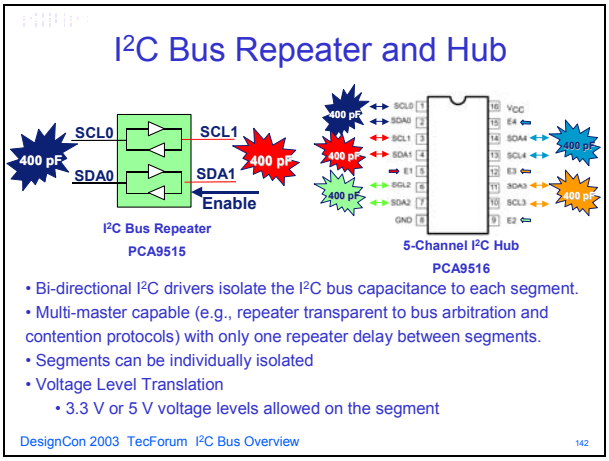
Voltage Level Translators



Slide 141

These devices are very useful in translation of I²C bus voltages as a lower and lower core voltages are used. The GTL2000 is 22 bits wide, the GTL2002 is 2 bits wide and the GTL2010 is 10 bits wide. See Application Note AN10145 for more information.

Bus Repeaters and Hubs



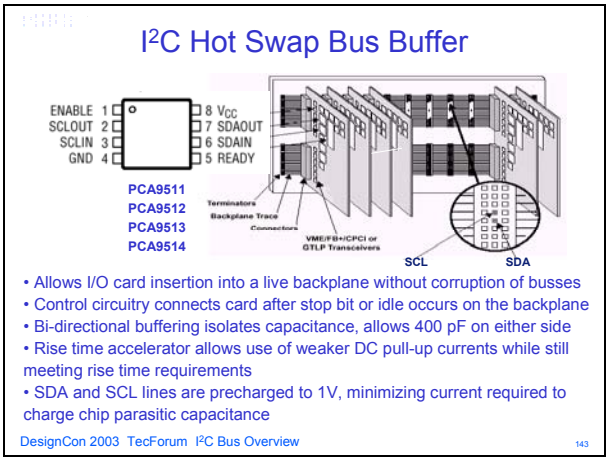
Slide 142

These bi-directional I²C drivers enable designers to isolate the I²C bus capacitance into smaller sections, accommodating more I²C devices or a longer bus length. The I²C specification only allows 400 pF load on the I²C bus and these devices can break the I²C bus into multiple 400 pF segments.

PCA9515 and PCA9516 applications include supporting the PCI management bus, > 8 PCI slots, isolating SMBus to hot plug PCI slots and driving I²C to multiple system boards. Either 3.3 V or 5 V voltages are allowed on each segment to allow devices with different voltages ranges to be used on the same bus. The devices are transparent to bus arbitration and contention protocols in a multi-master environment.

The PCA9518 expandable hub is designed to allow more multiple groups of 4 downstream channels.

Hot Swap Bus Buffers



Slide 143

The PCA9511 hot swappable 2-wire bus buffer allows I/O card insertion into a live backplane without corruption of the data and clock busses. Control circuitry prevents the backplane from being connected to the card until a stop bit or bus idle occurs on the backplane without bus contention on the card. When the connection is made, the PCA9511 provides bi-directional buffering, keeping the backplane and card capacitances isolated. Rise time accelerator circuitry allows the use of weaker DC pull-up currents while still meeting rise time requirements.

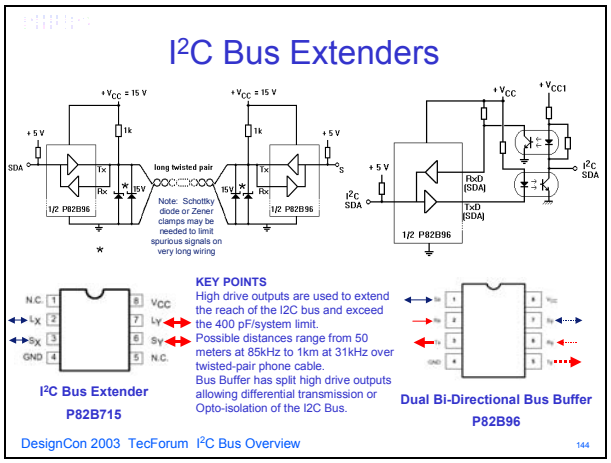
During insertion, the SDA and SCL lines are precharged to 1 V to minimize the current required to charge the parasitic capacitance of the chip.

The PCA9511 incorporates a digital ENABLE input pin, which forces the part into a low current mode when asserted low, and an open drain READY output pin, which indicates that the backplane and card sides are connected together.

The PCA9512/13/14 are variants on the PCA9511.

The PCA9511DP is an alternate source for the Linear Tech LTC4300-1I and the PCA9512DP is an alternate source for the Linear Tech LTC4300-2I.

Bus Extenders



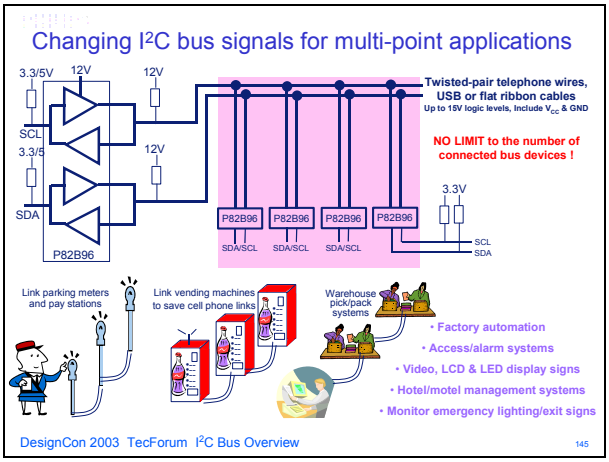
Slide 144

Applications requiring opto-isolation of the I²C bus (P82B96 only):

- Digital telephone answering machines (Philips PCD6001), Fax machines, feature phones and security system auto-dialers are connected to the phone line and often powered from the 110/230 V mains via double-insulated 'plug-pack' DC power packs. Many use Microcontrollers (e.g. PCD33xx), and some will already have I²C buses. Any other interfaces, e.g. connecting to the product's internal I²C bus, will require safety isolation.
- Medical equipment requires safety isolation of the patient connections. Any power for the isolated circuitry must be passed via isolating transformers. The data paths are sometimes transformer coupled using carrier tones, but they could also be via opto-isolated I²C.
- Lamp dimmers and switches can be controlled over I²C data links.
- Each light in a disco or live stage production could have its own identity and be individually computer controlled from a control desk or computer via I²C. Dimming (phase control) can be done with small micros or TCA280B from IES. Putting the phase controller inside each lamp will make it easier to meet EMC rules - lower power wiring radiation.

Applications requiring extension of the I²C bus (both P82B715 and P82B96):

- Almost any application where a remote control needs to be located some distance from the main equipment cabinet, e.g. in medical or industrial applications. Some safe distances the P82B715 or P82B96 can transmit I²C signals are:
  - P82B715: 50 Ω coax cable or twisted-pair cables - 50 meters, 85 kHz
  - P82B96: Telephone cable pairs or Flat Ribbon Cable - 100 meters at 71 kHz or 1 kilometer at 31 kHz



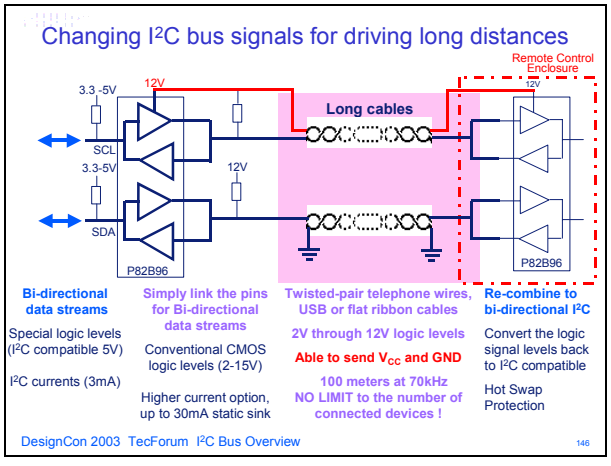
Slide 145

The buffered 12V bus has exactly the same multi-drop characteristic as a standard I²C but the restriction to 400 pF has been removed so there is no longer any restriction on the number of connected devices. P82B96 alone can sink at least 30 mA (static specification, > 60 mA dynamic) and there is no theoretical limitation to providing further amplification. Just adding a simple 2N2907A emitter-follower enables 500 mA bus sink capability.

With large sink currents it is possible to drive a special type of low impedance "I²C" bus - say at 500 Ω, or even down to 50 Ω. With the ability to use logic voltages up to 15 V it is possible to drive hundreds of meters of cable, providing the clock rate is decreased to allow time for the signals to travel the long distances. It's possible to run 100 meters with at least 70 kHz and 1kilometer at 30 kHz. That beats CAN bus, based on useful byte rate!

Note the special bus formed when the P82B96 Tx and Rx outputs are linked has all the usual properties of an I²C bus -- it IS an I²C bus, but with some of the limitations removed. So it is a 'multi-drop' bus that can support ANY NUMBER of physical connection nodes. Of course the method of addressing of individual nodes must be designed but it's easy with microcontrollers, and possible using hardware, to achieve sub-addressing.

Application examples: Parking meters and vehicle sensors are linked to a pay station, some have credit card and pay-by-phone options. Groups of vending machines can be linked so only one in a group needs a cell phone link for payment facility or reporting the stock/sales/faults situation. Warehouse systems transmit requirements to workstations, print labels, have real-time visibility of work status. Motel systems control access, air-con, messages via teletext on TV screen, report room status.



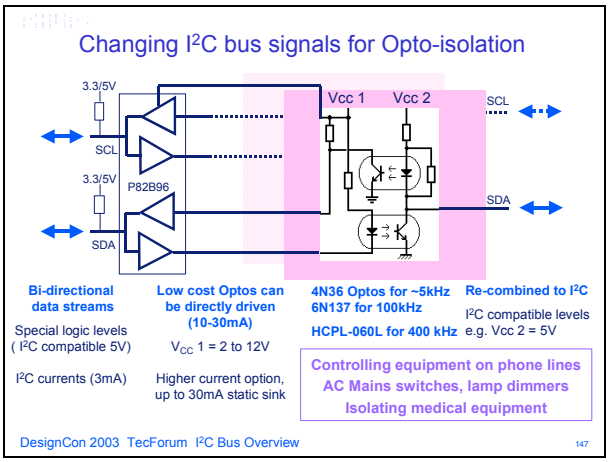
Slide 146

It is allowed to simply join the two unidirectional logic pins Tx and Rx to form a bi-directional bus with all the same features as I²C but with freedom to choose different logic voltages and sink larger currents than the 3 mA limitation of the I²C specifications.
P82B96 alone can sink at least 30 mA (static specification, >60 mA dynamic) and there is no theoretical limitation to providing further amplification.

Just adding a simple 2N2907A emitter-follower enables 500 mA sink capability.

This allows longer distance communication on the I²C bus. See Application Note AN255 for more information.

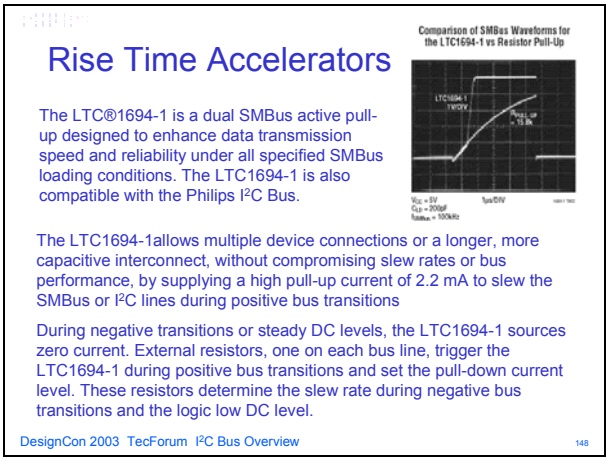
Electro-Optical Isolation



Slide 147

Here the 30mA drive capability at Tx is used to directly drive low cost opto-couplers to achieve isolation of the I²C bus signals. This allows I²C nodes in industrial applications (e.g. factory automation) to have their grounds at different potentials. It allows I²C chips inside telephones to interface to external devices that need to be grounded, for example to a PC to log Faxed information. It allows driving I²C chips connected to the AC power mains with a safety isolation barrier. The P82B96 allows operation up to 400 kHz.

Rise Time Accelerators

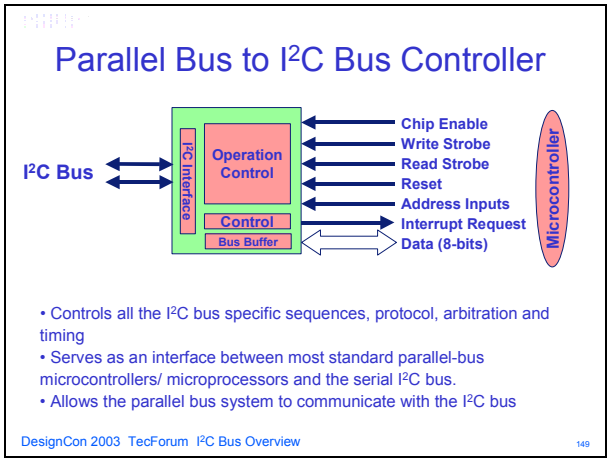


Slide 148

Rise time accelerators like the LTC1694 and LCT1694-1 are used to help control the rise time of the I²C bus.

See Application Note AN255 Appendix 6 for differences between the LTC1694 and LCT1694-1.

Parallel Bus to I²C Bus Controller



Slide 149

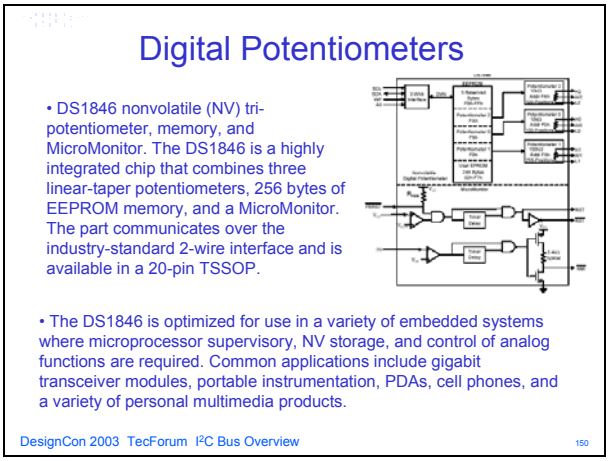
The PCF8584 and PCA9564 serve as an interface between most standard parallel-bus microcontrollers/ microprocessors and the serial I^2C bus and allow the parallel bus system to communicate bi-directionally with the I^2C bus. This commonly is referred as the bus master. Communication with the I^2C bus is carried out on a byte-wise basis using interrupt or polled handshake. It controls all the I^2C bus specific sequences, protocol, arbitration and timing.

The PCA9564 is similar to the PCF8584 but operates at 2.3 to 3.6 V V_{CC} and up to 400 kHz (slave mode) with various enhancements added that were requested by engineers.

	PCA9564	PCF8584	Comments
1. Voltage range	2.3-3.6V	4.5-5.5V	PCA9564 is 5V tolerant
2. Max I^2C freq.	360 kHz	90 kHz	Faster I^2C
3. Clock source flexible	Internal	External	Less expensive/more
4. Parallel interface processors	Fast	Slow	Compatible with faster

In addition, the PCA9564 has been made very similar to the Philips standard 80C51 microcontroller I^2C hardware so existing code can be utilized with a few modifications.

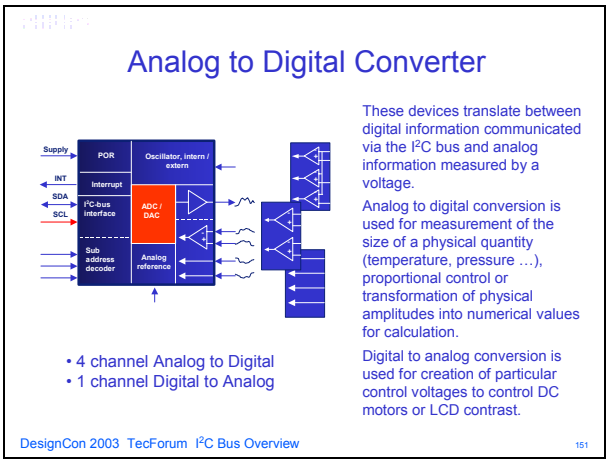
Digital Potentiometers



Slide 150

Digital potentiometers are similar to the potentiometers you used to adjust with the screwdriver but these are adjusted via the I^2C bus. Some digital potentiometers include onboard EEPROM so that settings are retained with the device is powered down.

Analog to Digital Converters

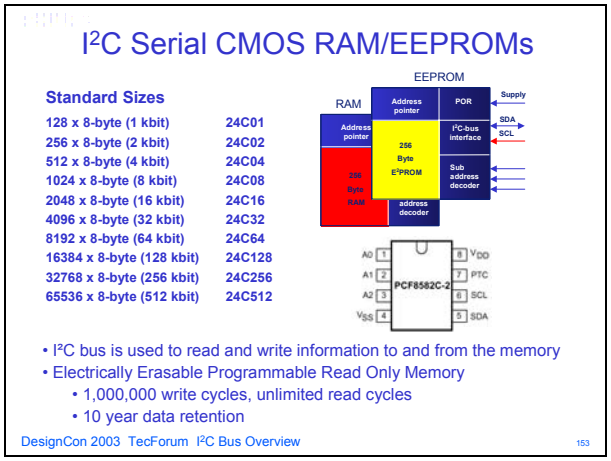


Slide 151

The PCF8591 is capable of converting four different analog voltages to the digital values for processing in the microcontroller. It can also generate one analog voltage by converting an 8-bit digital value provided by the microcontroller.

Several kinds of analog information in your applications, such as temperature, pressure, battery level, signal strength, etc can be processed by such a device. These are digitally processed and can be subsequently displayed, used to control contacts, switches, relay, etc. for example using the previously discussed I/O expander PCA9554. The D/A output is useful for such jobs as LCD contrast control.

Serial RAM/EEPROM



Slide 153

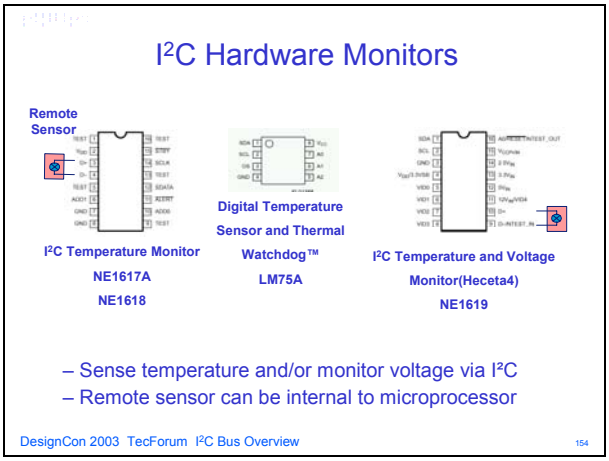
There are different kinds of memories in the line of I²C bus compatible components such as: RAM, EEPROM, video memories and Flash memories.

- RAM is Random Access Memory
- EEPROM is Electrically Erasable Programmable Read Only Memory
- Common small serial memories (RAM and EEPROM) are often used in applications. EEPROMs are particularly useful in applications where data retention during power-off is essential (for example: meter readings, electronic key, product identification number, etc).
- A single pinning is used for these ICs because they are very similar and their pinouts have been intentionally designed for interchangeability.
- EEPROMs store data (2kbits organized in 256 x 8 in the PCF8582C-2 for example), including set points, temperature, alarms, and more, for a guaranteed minimum storage time of ten years in the absence of power. EEPROMs change values 100,000 to 1,000,000 times and have an infinite number of read cycles, while consuming only 10 micro amperes of current.

For example, the PCA8581 is organized as 128 words of 8-bytes. Addresses and data are transferred serially via a two-line bi-directional bus (I^2C bus). The built-in word address register is incremented automatically after each data byte is written or read. All bytes can be read in a single addressing operation. Up to 8 bytes can be written in one operation, reducing the total write time per byte.

The PCA8582C-2 is pin and address compatible to: PCF8570, PCF8571, PCF8572 and PCF8581. The PCF85102C-2 is identical to the PCF8582C-2 with pin 7 (Programming time control output) as a 'no connect' to allow it to be used in competitors sockets since PTC should be left floating or held at V_{CC}. The PCF85103C-2 is identical to the PCF85102C-2 except that the fixed I^2C address is different, allowing up to eight of each device to be used on the same I^2C bus.

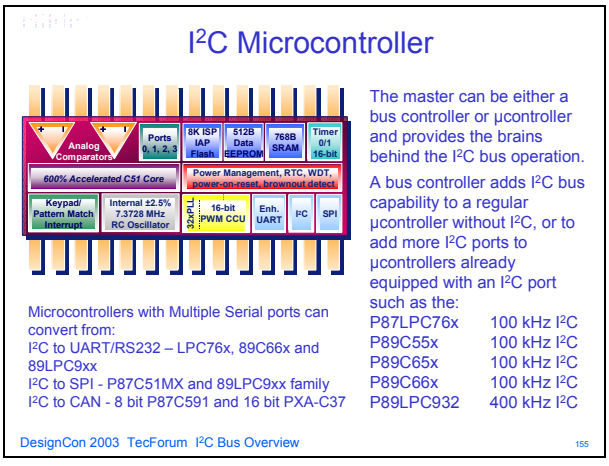
Hardware Monitors/Temp & Voltage Sensors



Slide 154

Hardware monitors such as the NE1617A, NE1618, NE1619 and LM75A use the I²C bus to report temperature and/or voltage. Some of the temperature monitors include hardware pins that allow external transistors/diodes to be located in external components (e.g., processors) that sense the temperature much more accurately then if the sensor was mounted externally on the package. The test pins are used at the factory to calibrate/set the temperature sensor and are left floating by the customer.

Microcontrollers



Slide 155

Microcontrollers are the brains behind the I^2C bus operation. More and more micros include at least one I^2C port if not more to allow multiple I^2C buses to be controlled from the same microcontroller.

I²C Patent and Legal Information

The I²C bus is protected by patents held by Philips. Licensed IC manufacturers that sell devices incorporating the technology already have secured the rights to use these devices, relieving the burden from the purchaser. A license is required for implementing an I²C interface on a chip (IC, ASIC, FPGA, etc).

It is Philips's position that all chips that can talk to the I²C bus must be licensed. It does not matter how this interface is implemented. The licensed manufacturer may use its own know how, purchased IP cores, or whatever. This also applies to FPGAs. However, since the FPGAs are programmed by the user, the user is considered a company that builds an I²C-IC and would need to obtain the license from Philips.

Apply for a license or text of the Philips I²C Standard License Agreement

- US and Canadian companies: contact Mr. Piotrowski (pc.mb.svl@philips.com)
- All other companies: contact Mr. Hesselmann (pc.mb.svl@philips.com)

ADDITIONAL INFORMATION

The latest datasheets for both released and sampling general purpose I²C devices and other Specialty Logic products can be found at the Philips Logic Product Group website: http://www.philipslogic.com/i2c

Datasheets for all released Philips Semiconductors I²C devices can be found at the Philips Semiconductors website: http://www.semiconductors.philips.com/i2c

More information or technical support on I²C devices can be provided by e-mail: pc.mb.svl@philips.com

APPLICATION NOTES

AN168	**Theory and Practical Consideration using PCF84Cxx and PCD33xx Microcontrollers**
AN250	**PCA8550 4-Bit Multiplexed/1-Bit Latched 5-Bit I²C E2PROM**
AN255	**I²C/SMBus Repeaters, Hubs, and Expanders**
AN256	**PCA9500/01 Provides Simple Card Maintenance and Control Using I²C**
AN262	**PCA954X Family OF I²C/SMBus Multiplexers and Switches**
AN264	**I²C Devices for LED Display Control**
AN444	**Using the P82B715 I²C Extender on Long Cables**
AN460	**Using the P82B96 for Bus Interface**
AN469	**I²C I/O Ports**
AN10145	**Bi-directional Low Voltage Translators GTL2000, GTL2002, GTL2010**
AN10146	**I2C 2002-1 Evaluation Board**
AN95068	**C Routines for the PCx8584**
AN96119	**I²C with the XA-G3**
AN97055	**Bi-Directional Level Shifter for I²C-Bus and Other Systems**
ANP82B96	**Introducing the P82B96 I²C Bus Buffer**

ANZ96003 Using the PCF8584 with Non-Specified Timings and Other Frequently Asked Questions