Design Notes

Designing ISA Bus Expansion Cards

Ever since PCs appeared on the market in the early 1980's, their design has been based around the Industry Standard Architecture (ISA). The ISA bus consists of address, data and control lines which are connected to the ISA slots on the PC, and are used for interfacing I/O cards such as modems, sound cards, video cards etc to the host processor.

This month's Design Notes discusses the operation of the ISA bus. We look at signaling, interfacing, and a variety of modes used for transferring data between the card residing in the slot and the processor. We also discuss a typical design example and show you some tricks (and traps) to make life easier when designing an ISA interface card. Due to space limitations, we will not be discussing DMA control. This will be examined in-depth in a future edition of Design Notes.

ISA Bus Signals

An ISA bus expansion card can be designed for 8 bit or 16 bit data transfer. 8 bit cards can use up to 62 pins of the ISA connector, while 16 bit cards can use up to an additional 36 pins. The ISA bus connectors commonly found on a PC motherboard, segment the 8 bit pins from the 16 bit pins using an integral plastic rib within the connector.

ISA bus signals consist of address, data and control lines including interrupt requests, read and write strobes, clock, and wait state signals. ISA connectors also carry power and ground.

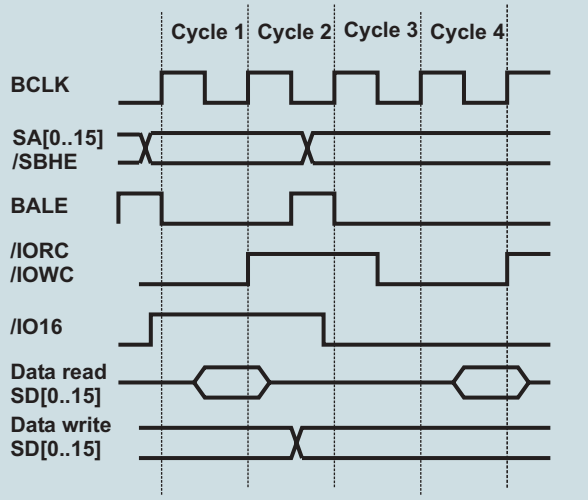

Figure 1: Access to 16 bit I/O Device

Following is a brief explanation of the function of each pin, however the reader should also consult reference (1) for further information. Let's consider the 8 bit pins first:

SA19:0 - System Address Bus

The address lines connected to the PC's processor. These can be used for address decoding on the expansion card.

BALE - Buffered Address Latch Enable

On the rising edge of this signal the address of the current bus cycle is placed on the SA19:0 bus. On the falling edge, the state of the address bus is deemed to be valid. Expansion cards should use the falling edge to latch the state of the address lines (SA19:0) and perform an address decode.

BCLK - Bus Clock

The timing clock upon which the bus' operation is synchronised. It can vary from 6MHz for early AT's to 12.5MHz for modern PCs.

SD7:0 - System Data Bus (Lower)

Data bus, lower 8 bits. The upper 8 bits are present on the 16 bit portion of the connector.

/IOWC & /IORC - I/O Write & Read Strobes

Write and read strobes for writing or reading an expansion card in I/O space (as opposed to memory space).

Electronics By Design

Suite 25, 1-7 Jordan St Gladesville 2111
Phone : (02) 9816-3965
Fax: (02) 9816-3967
Web: www.electronicsbydesign.com.au

/SMWTC & /SMRDC - System Memory Write & Read Strobes
Similar to /IOWC and /IORC but are active when accessing expansion cards in memory space below 1MB.

RESET - Reset Line
Active high reset line which is asserted on power-on reset or user reset.

+5V, -5V, +12V, -12V, 0V - Supply Lines
Power supply lines. Care must be exercised when connecting to the +5V rail as the power supply is capable of driving at least 30A in most PCs, thus a short circuit can cause big problems such as burnt boards and damaged components.

IRQ2-IRQ9 - Interrupt Requests
Interrupt request lines which are driven high when a particular interrupt is asserted. Most of these lines are committed to existing PC functions. For example IRQ4 is the serial port 1 interrupt, IRQ7 is the parallel port interrupt, etc. These lines are outputs from the expansion slot to the processor and should be driven using a tri-state buffer with a pull-down resistor. Signals are active high.

/NOWS - No Wait States
An ISA bus memory or I/O bus cycle will have a specified amount of wait states as it's default scenario. This can typically be 1, 2, 3, or 4 depending on the cycle type. In applications where speed is very important, the /NOWS line can be asserted by the expansion card to reduce (or eliminate) the number of wait states.

CHRDY - Channel Ready
This signal operates in the opposite way to /NOWS. If an expansion card requires more wait states than the default amount, it asserts this line for as long as required and the PC processor will insert wait states until CHRDY is deasserted.

/REFRESH - DRAM refresh Signal
Asserted when the motherboard is refreshing DRAM memory.

OSC - Oscillator
A 14.31818MHz free running oscillator available for use by the expansion card.

/CHCHK - Channel or I/O check
Asserted by the ISA expansion card when an error condition occurs which warrants a non maskable interrupt (NMI) being generated. This can include a brown out or power fail condition.

The following signals are in addition to the above, and are used only for 16 bit cards:

LA23:17 - Latched Address Bus
These signals are buffered versions of the upper bits of the address bus. They present the address of the next bus cycle prior to the beginning of the actual cycle. They can be used for performing an early address decode for high speed expansion cards.

/MWTC & /MRDC - Memory Write & Read Strobes
Used for strobing 16 bit memory hardware on the expansion card.

/M16 - Memory Size 16
This signal is asserted by an ISA expansion card to indicate it recognises it is being addressed as a 16 bit card, and can communicate over both data paths.

/IO16 - I/O Size 16
Similar to /M16 but it is asserted during a 16 bit I/O space access.

IRQ10-IRQ15 - Interrupt Requests
Similar to IRQ2-IRQ9.

The remaining signals, namely DRQx, DAKx, TC, AEN, and /MASTER16 are used for DMA data transfer and will not be discussed further.

ISA Bus Operation
Like most Intel based processor architectures, ISA bus uses separate memory and I/O address spaces. The bus cycle for both types of accesses is similar. The only major difference are the read and write strobe signals. Memory space uses /SMWTC and /SMRDC, while I/O uses /IOWC and /IORC.

One limitation of I/O access is it cannot address space greater than 64k byte. This is not usually a problem when communicating with an ISA expansion card, because there are plenty of address locations to map as many expansion

cards as one wishes. When a designer chooses an I/O location for mapping, one must ensure it doesn't clash with other peripherals. These include other ISA cards, serial ports, parallel ports, video cards, floppy disk controllers and many others. The designer should find out which I/O addresses are reserved and avoid them. Reference (1) provides a great lookup table for this purpose. An ISA card should have jumpers for mapping the card in a choice of I/O locations to minimise the risk of bus conflict with other locations.

ISA bus clock speed varies depending on the PC. Recent PCs operate at 12.5MHz, while older PCs will run slower, typically down to 6MHz. The designer should ensure the expansion card will operate at the fastest possible clock speed for which it is intended.

When designing an ISA expansion card particular attention must be paid to the power-on reset state of the card. The hardware interfacing to the ISA bus, must appear as a high impedance on power up, otherwise it will crowbar the ISA signals and prevent the PC from booting.

ISA expansion card hardware must use strong drivers for all outputs, typically 24mA bus specification drivers. This includes address, data and control lines, in other words, everything should be buffered.

ISA Bus Transfer

ISA bus transfers which do not involve DMA can be 8 or 16 bit, either of which can be memory or I/O transfers. To accommodate slow hardware, the bus can also accommodate wait stating.

As an example we will review a typical bus cycle as shown in figure 1. This is an example of a 16 bit transfer in I/O space. The transfer takes four clock cycles. The synchronous clock signal is BCLK which is available on the ISA connector.

During cycle 1 and halfway into cycle 2, the address lines are asserted. The address latch signal BALE, is asserted in the middle of the second cycle and de-asserted at the end of the second cycle. The ISA expansion card hardware should use this signal to latch the state of the address lines on the falling edge of BALE, because it is at this point they are correct and stable. This then allows the card to address decode the signals. The address decoding must be fast so the chip select of the expansion card hardware is asserted before the upcoming read or write strobe, with sufficient time to satisfy all timing delays including setup and propagation delays.

When the ISA card has done this and hence detected it is being addressed, it should assert /IO16. This indicates to the processor that the card wants to transfer data over both data paths, namely SD15:0.

In the middle of the third cycle, the processor asserts the read or write strobe as appropriate. This will be either /IORC or /IOWC. This is normally used to strobe a latch or RAM chip in the expansion card. Data transfer can then take place over the data bus.

If the processor writes data to the expansion card, it places the data on the data bus and waggles the write strobe line /IOWC from high to low and high again, thus allowing the expansion card to latch the data on the rising edge of this signal.

If a read cycle is executed, the expansion card should place the data on the bus upon the falling edge of the read strobe /IORC. The processor will then waggle /IORC and the data will be latched on the rising edge of /IORC.

An ISA bus transfer can be wait stated. Some transfer cycles (such as the one we have just discussed), already have default wait states inserted, however the expansion card hardware can insert further wait states using the signal "channel ready" (CHRDY). This signal is active high but can be pulled low by the hardware before the second last clock cycle of the transfer in order to insert an indefinite amount of wait states. As long as it's low, wait states will be inserted.

If a transfer has a default number of wait states and the expansion card hardware wishes to eliminate them to speed things up, it can assert

the signal “no wait state” (/NOWS). This has the opposite effect of CHRDY.

Typical Design Example

Following is a design example which illustrates some of the points we’ve discussed. This is a real design which Electronics By Design undertook earlier this year.

A customer required an ISA expansion card which would interface to his target hardware and allow it to be tested and configured after it rolled off the production line. The card could also be used in a service environment to check if the target hardware was operating correctly, and configure various setup parameters. Due to confidentiality requirements, we cannot divulge anything specific about the target hardware or the application, however we can still illustrate how the design of the interface card was approached and executed.

The interface card had to provide the following:

1) Digital I/O which could be controlled by software.
2) Current consumption measurement of the target hardware.
3) Serial port.
4) Card access mapped in I/O address space.
5) Interrupt request lines, jumper selectable to one of four IRQ lines.

There were two possible approaches to implement the design. It could be done using glue logic, or through the use of an FPGA. For versatility the latter solution was chosen and a Xilinx XC4003 FPGA was used. The FPGA was interfaced to the address lines, data lines and control lines. In fact any ISA signals that were needed were connected directly to it. The FPGA performed address decoding for the digital I/O latches which interfaced to the target hardware. The FPGA also performed address decoding for the UART. An A/D converter was used to measure current consumption of the target hardware across a current sensing resistor. It required various functions to be managed, including starting the A/D conversion, detecting end of conversion and reading the data out of the converter and into the PCs processor. All these functions were implemented in the FPGA.

The design used an on-board EPROM to store the configuration bitstream of the FPGA for loading on power up.

The design was wire wrapped on a prefabricated blank ISA bus board as per the customer’s requirements.

The FPGA mapped the ISA card into one of eight possible base address locations in blocks of 16 bytes. The required base address was selected using on board jumpers. An alternative method of defining the address mapping would be to use a pre-programmed serial EEPROM which contained the required location, and it could be read on power up by the FPGA.

It can’t be emphasized enough, that careful consideration must be given to the behavior of the card on power up. Since the FPGA’s operation is undefined until it is configured, it must be isolated from the ISA bus until configuration is complete. This was done using tri state buffers to isolate the address lines of the FPGA, and connect them to the configuration EPROM used for programming.

An interrupt into the PC processor was also required for the serial port residing on the expansion card. Jumpers were used for this purpose to select one of four possible IRQ lines. Care was used to ensure the IRQ lines were not being used for other peripherals inside the PC.

The design was implemented as an 8 bit interface card and was memory mapped in I/O space, clocking from BCLK. Test software was written to check the card’s operation and performance. A low cost PC was also used to test the card’s operation. The design worked very effectively and required minimal effort to debug, mainly thanks to the versatility of the FPGA.

References and Acknowledgments

(1) ISA System Architecture
3rd edition
Tom Shanley / Don Anderson
(2) ISA, Xilinx, IBM, are registered trademarks.