ELEC 464 : MICROCOMPUTER SYSTEM DESIGN
1996/97 WINTER SESSION TERM 1

The System Bus

This lecture describes system buses. These are buses used to interface the CPU with memory and peripherals on separate PC cards. The ISA, VME and PCI buses are used as examples.

Introduction

To increase their flexibility, most general-purpose microcomputers include a system bus that allows printed circuit boards (PCBs) containing memory or I/O devices to be connected to the CPU. This allows microcomputer systems to be customized for different applications. The use of a standard bus also allows manufacturers to produce peripherals that will work properly with other manufacturers' computer systems.

The system bus consists of a number of parallel conductors on a backplane or motherboard. There are a number of connectors or "slots" into which other PCBs containing memory and I/O devices can be plugged in.

In most cases the system bus is very similar to the processor bus. In fact, the simplest system buses simply consists of a connector that allows access to all of the pins on the CPU chip. Like the processor bus, the the system bus can be subdivided into an address bus, data bus, control bus and power supply connectors.

Some microcomputer designs place the CPU and some auxiliary circuits on a PCB (the "motherboard") along with system bus connectors. Example of this approach include most popular IBM PC-compatible systems. Other designs use a simpler passive "backplane" type of bus and place the CPU on its own PCB. This approach is used by most VME-based microcomputers. The latter approach has the advantage that the CPU card can be replaced if required. Because the bandwidth required for the RAM interface is much higher than that for most other peripherals, most modern systems use a system bus only for I/O devices and either use a proprietary interface for RAM or simply place all of the RAM memory on the same PCB as the CPU.

A bus can be described by its mechanical (size, types of connectors, etc), electrical (voltage, clock rates, etc), and protocol (read/write cycles, master/slave operation, interrupt acknowledgement, etc) characteristics.

We will look briefly at three examples of system buses. The ISA (Industrial Standard Architecture) bus is commonly used in IBM-PC compatibles and is one of the most widely-used system busses. The VME (Versa Module Europe) bus is used in many industrial computer systems and a wide range of (relatively expensive) interface cards are available for it. The VME bus includes features common to other high-performance buses such as arbitration for multiple bus masters. The PCI (Peripheral Component Interconnect) bus is a flexible high-performance peripheral bus that can efficiently interconnect peripherals and processors of widely different speeds.

Mechanical Characteristics

Low-cost consumer-grade buses use card-edge connectors to minimize the cost of the peripheral. The plug-in card has contact pads along the edges of the PCB. The motherboard has connectors on the motherboard that contact these pads. Typical examples include the ISA and PCI busses.

Some buses use connectors on both the motherboard and the card. This adds to the cost of the card (about $5-$10) but makes for more reliable contacts. A typical example is the 3-row by 32-pin (96-pin) "DIN" connector used by the VME bus.

Electrical Characteristics

The electrical characteristics include the voltage and current specifications, types of bi-directional signals and impedance matching.

Bus Drivers and Receivers

Bus signals such as clocks or data lines often drive several chips on a peripheral card. If the load presented by the card exceeds the bus specifications then buffer chips (“bus receivers”) must be used. Similarly, if the output driving capacity of a chip on the peripheral card is not enough to drive the load presented by the bus, a “bus driver” must be used to buffer the signal.

Bidirectional Buses

Some bus signals can be driven by multiple cards. For example, the data bus will be driven by all peripheral devices as well as the CPU. In this case bidirectional buffers called “bus transceivers” are used. Two common methods are open-collector and tri-state outputs. As always, care must be taken in the design to prevent multiple simultaneous outputs on the same bus line (bus conflicts).

Open-Collector (OC) Logic

OC outputs can only sink current, not source it. A pull-up resistor is used to pull that particular bus line to V_{cc}. Any device connected to that line can pull the signal low. The resistance of the pull-up resistor must be sufficiently low that the maximum voltage drop across it still provides a valid high logic level and the resistance must be high enough to limit the current through it to a value that will not damage the OC outputs.

The choice of pull-up resistance is also a compromise between rise time and power consumption.

Tri-State Logic

Tri-state logic outputs can controlled to be in a high output, a low output or in a high-impedance mode. They are used to drive bus signals that might be driven by more than one device.

For example, on a system that has multiple bus masters the address data bus must be driven by tri-state outputs and the data bus driven by tri-state bus transceivers.

Dynamic Contention

It should be remembered that buffers (receivers/drivers/transceivers) take a finite time to switch modes. A detailed timing analysis is necessary to ensure that two outputs will not be enabled simultaneously. This analysis will needs to take into account the CPU timing as well propagation delays for the circuits that control the buffer directions.

Impedance and Termination

To analyze the performance of the bus for the short rise times required in high-speed buses the bus must be treated as a transmission line.

The pulse generated by a bus driver will propagate down the bus. If the bus is not terminated on both ends by the characteristic impedance of the bus a portion of the signal will be reflected. The reflections will propagate back and forth along the bus and appear as ringing and noise. In addition, there will be reflections from each tap where a card is connected to the bus.

Typical propagation times for signals on a 10-mil (.01 inch) tracks are on the order of 6 ns/metre. If the rise time of a device is on the same order as the propagation delay there is a potential for reflections caused by improper termination to cause problems.

For the above reasons it is important that high-speed buses be terminated in the characteristic impedance of the bus. The typical characteristic impedance is on the order of 100 to 150 ohms. It is not possible to simply tie a resistor equal to the bus characteristic impedance to ground because this would pull the signal low. Instead, the resistor is connected to a low-impedance voltage source at about half of V_{cc}.

ISA Bus

The bus used for expansion cards by the original IBM PC was designed to support expansion memory and peripherals such as video displays and parallel printer ports. It had the same 8-bit data bus and 20-bit address space as the Intel 8088 processor in the XT. The subsequent PC/AT (ISA) bus used a second card-edge connector to extended the address space to 24 bits and the data bus to 16 bits.

The PC cards are about 4 inches high and up to 13 inches long and use one (PC) or two (AT) card-edge connectors.

Intel chips have a separate I/O address space and the ISA bus includes MEMR* (memory read) and MEMW* (memory write) and IOR* (I/O read) and IOW* (I/O write) strobes.

The PC bus is synchronous. The CPU performs fixed-length read and write cycles although a WAIT signal is available on the bus so that slow peripherals can request wait states.

The PC bus has 6 active-low interrupt request lines (IRQ2* to IRQ7*). The PC motherboard has a programmable interrupt controller chip (intel 8259) that arbitrates different levels of interrupt requests and generates an interrupt number in response to the processor's interrupt acknowledge cycle.

The bus also has 3 DMA request (DRQ*) and acknowledge (DACK) lines that can be used by peripherals with DMA capabilities.

The PC/AT (ISA) bus (but not the PC/XT bus) allows for a limited form of external bus mastering.

VME Bus

Background

The VME system bus was developed by Motorola specifically for 68000-based systems. It has also been adapted to other processors. Unlike the ISA bus the VME bus standard is precisely specified. It includes advanced features such as complex arbitration mechanisms for multiple bus masters and interrupt sources.

Mechanical

VME bus cards can be single- or dual-height cards. The dual-height (160x233mm) cards use two 96-connector "DIN" connectors (P1 and P2) to connect to the backplane. Single-height cards (160x100mm) use only one connector (P1). The P1 connector contains a 24-bit address bus, a 16-bit data bus and most of the control bus. Connector P2 contains additional address bus bits (extending it to 32 bits) and data bus bits (extending it to 32 bits).

Like the ISA bus, the VME bus uses TTL voltage levels.

VME Functional Modules

The VME bus supports the standard read, write and interrupt acknowledge cycles as well as block read and write cycles that allow the CPU to read/write a contiguous block of memory. The VME bus also allows a read-modify-write cycle that allows atomic (un-interruptible) memory updates.

The operation of the bus is defined in terms of the operation of functional blocks that may be present on cards. A particular card will contain only those functional blocks that are relevant to its operation. Among the possible functional modules are:

A *master* initiates transfers of data over the DTB (data transfer bus). A *slave* responds to data transfers initiated by a bus master. A *location monitor* decodes addresses to determine when a slave is being accessed. A *interrupter* generates interrupts and responds to interrupt acknowledge cycles. A *requester* is capable of requesting use of the bus. An *arbiter* determines which master will get control of the bus.

Exercise: Which functional modules do you think would be found on a disk controller card? On a memory card? On a CPU card?

VME Data Transfer Bus

The VME bus is very similar to the 68000 processor bus. It uses 31 address lines, and DS0*, DS1* and LWORD* strobes in the same fashion as the 68000 family. The address modifier lines AM0 to AM5 indicate the type of address space (supervisor, interrupt acknowledge, etc) in a similar way as the FC* lines are used on the 68000. There is a 32 bit data bus and unaligned read and write operations are allowed.

The VME DTB is asynchronous and a DTACK* signal is used (as in the 68000 processor bus) to indicate completion of a read/write cycle.

Bus Master and Interrupt Arbitration

The VME bus allows for sophisticated arbitration of multiple requests for bus ownership. A number of possible arbitration mechanism are defined in the bus specification with the implementation depending on the needs of the application. In addition, it is possible for different bus masters to respond to different interrupts by first obtaining bus ownership if required.

PCI Bus

The PCI bus is a high-performance peripheral (system) bus. The PCI bus maximizes CPU performance by decoupling the CPU and peripheral buses as much as possible. This decoupling also makes the bus relatively independent of the host CPU architecture.

The PCI bus is connected to the CPU bus by means of a *bridge*. This is an interface circuit that multiplexes address and data signals from the CPU, buffers (caches) data transferred between the CPU and the cards on the PCI bus, and optimizing access to sequential memory locations.

The PCI bus has multiplexed 32-bit data and address buses and runs at 33 MHz. A read cycle takes 3 clock cycles and a write cycle takes 2 clock cycles. Thus the bus bandwidth is 44 or 66 MB/s.

For example, if consecutive memory locations are being accessed the PCI bus allows sequential read or write cycles to proceed without explicitly putting the address on the bus. In this case on 32-bit word can be transferred in each clock cycle for a bus bandwidth of 132 MB/s.

The buffering provided by the PCI bridge allows the CPU to write data to a PCI peripherals without incurring wait states. The CPU can continue executing at full speed while the bridge takes care of the handshaking with the peripheral. The PCI bridge can also pre-fetch data to improve performance on reads.

It is also possible to use a second bridge between the PCI bus and another type of bus, for example, an ISA or VME bus. This allows slow peripherals to be efficiently interfaced to the CPU because this second bridge takes care of the details of interfacing the PCI bus and the slower bus.

The basic PCI bus signals include:

Signal	Purpose
AD0 to AD31	multiplexed address and data
FRAME*	indicates start of transfer
C/BE0 to C/BE3	type of cycle and byte enables
IRDY*, TRDY*	ready (to generate wait states)
CLK	clock
REQ*/GNT*	bus request/grant

Like the VME bus, the PCI bus allows DMA by allowing different cards to acquire control over the bus. Each card has bus REQuest* and bus GraNT* signals that are used to communicate with a bus arbitration circuit.

There is a 64-bit extension to the PCI bus that increases the maximum bus bandwidth to $33 \times 8 = 266$ MB/s.