Unibus Electrical Characteristics Limit Optimum System Performance

Accumulated capacitive, resistive, and inductive effects on the Unibus reduce its reliability and throughput, but solutions to these problems exist

by John Jones
Setasi Research And Development

The Unibus is a parallel, bidirectional, asynchronous bus composed of 56 signal lines and provides the computer's major data highway, performing all transfers among the system peripherals, memory, and CPU.

Due to its asynchronous design, it handles devices of widely varying data transfer rates without increasing system overhead. Because each device runs at its fastest possible rate, replacing an older, slower device with a newer, faster one means the system runs faster with no other hardware or software changes.

Full direct memory access (DMA) is inherent in the structure with byte, word, or multiple word transfer capability, and is available to all DMA devices simultaneously.

Memory is the primary speed control device, and with MOS memories operating at 650–700 nsec, the Unibus can handle a 2-byte data transfer in approximately 900 nsec. This translates to more than a 2.2 Mbyte/sec. transfer rate that is available to any device on the system.

Its modular design allows maximum ease of configuration as virtually any device can be connected where there is physically enough room (Figure 1). This scheme also implements automatic priority selection and nonpolled high speed vectored interrupts. Power fail and auto restart are supported with either core memory or battery backed up MOS memory systems.

The Unibus architecture is also supported on many CPUs that aren't limited by the 256-Kbyte memory addressing scheme. These CPUs access their memory on a separate bus that the Unibus can also access; they can have main memory capacities up to 64 Mbytes.

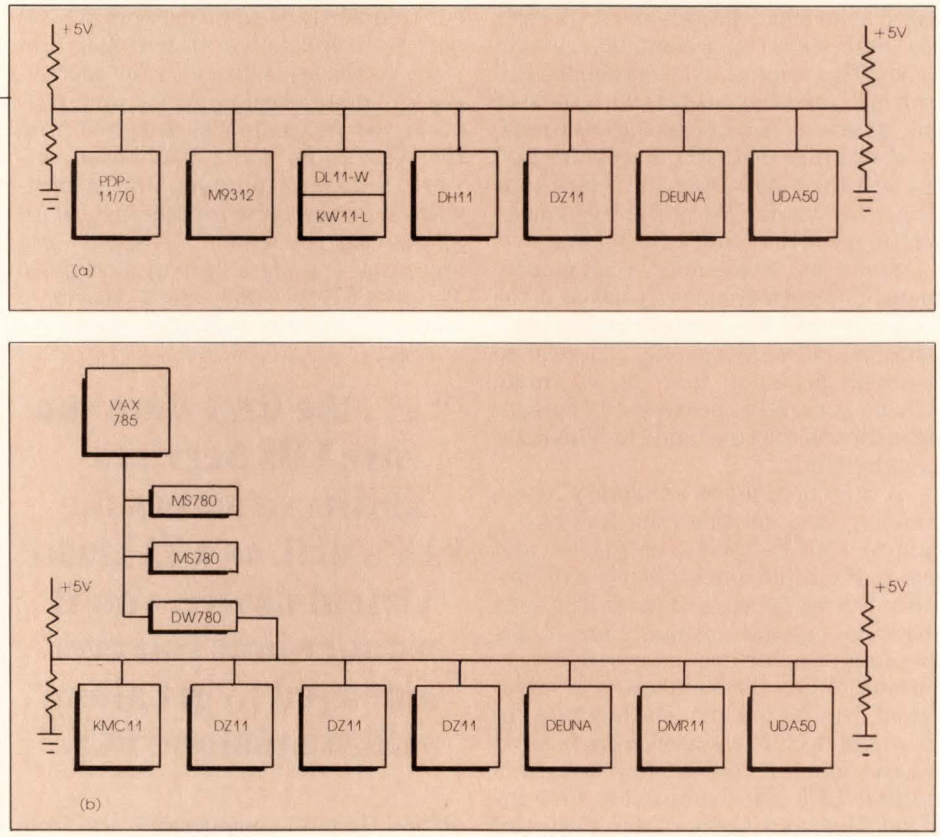


Figure 1—*The Unibus supports both standard PDP-11* (a) *and VAX* (b) *architectures.*

Unibus Protocol

Unibus protocol consists of handling 56 signal lines grouped as address, data, control, arbitration, and failure reporting. These lines are asserted low (<=.8V) true, high (>=2.5V) false, with the five grant lines vice versa.

The signals of primary importance are the control signals, as these are effective immediately upon assertion or de-assertion, and are not de-skewed (as are the address and data). Another important item in an asynchronous bus is the fact that leading edges (asserting) and trailing edges (de-asserting) have

John Jones, president of Setasi Research and Development (Hollywood, Fla.), holds a Ph.D in computer science from the University of Massachusetts. He and his partner, Bud DeFore, worked at Digital Equipment Corp. for many years as a customer troubleshooting team.

equal importance in the successful completion of a transfer.

An example of a simple Unibus transfer best explains this. Consider that the CPU needs data from memory and is ready to use the bus. The following events occur (Figure 2).

- The CPU checks to see if any higher priority device is requesting use of the bus.
- If not, the CPU, now bus master, asserts bus busy (BBSY) to notify all other devices the bus is in use.
- The CPU then asserts the address of the location in memory on address lines A0-A17 and control lines C0 and C1 to indicate a read word operation.
- The CPU waits 150-180 nsec for the lines to settle (de-skew), then asserts master sync (MSYN) to notify all devices on the bus that a command is waiting for the slave identified by the address.
- The slave device—memory—decodes that it is to respond to this address, does an internal select, drives the selected data onto the data lines, then asserts slave sync (SSYN) to indicate to the master that its request has been fulfilled.
- The CPU, upon receipt of SSYN, latches the data on the data lines, then de-asserts the two lines commanding the bus, BBSY and MSYN.
- Memory, upon the de-assertion of MSYN, knows the transaction is complete, performs an internal de-select, and de-asserts drive on both the data lines and slave sync.
- The cycle is now complete and the bus is again ready for use.

While this transfer was occurring, the next bus cycle was being primed in the bus arbiter, and commences as soon as BBSY is de-asserted. The new bus master then asserts BBSY and the protocol implemented for its cycle—either interrupt request or DMA.

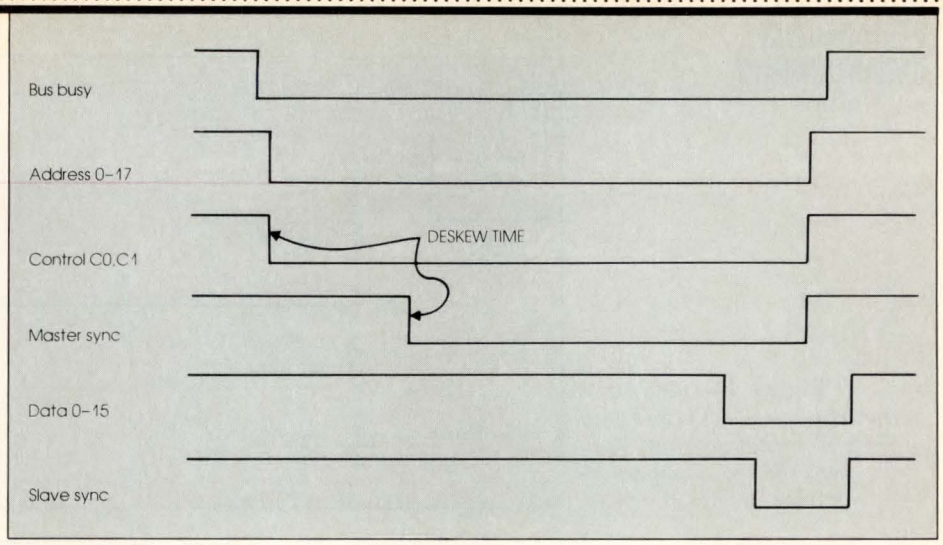
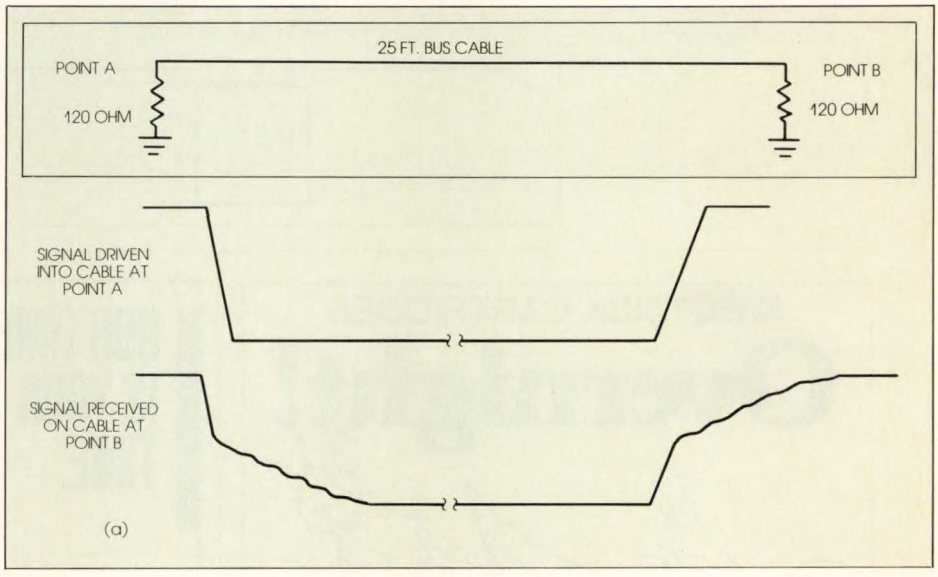


Figure 2—*The Unibus waits a de-skew time of 150-180 nsec for the bus to settle down before asserting the master sync signal to notify all devices on the bus that a command is waiting for the slave identified by the address.*



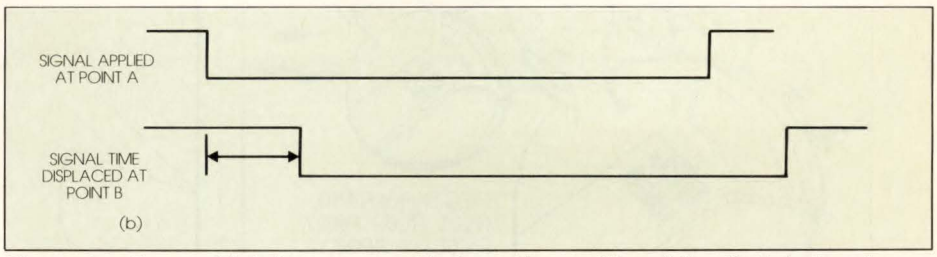


Figure 3—*Due to high frequency cable loss, the quality of the digital signals on the Unibus deteriorates as the signal moves through the cable* (a). *Propagation delay is another problem, and is defined as the difference in time between when a signal occurs and when it is received by the affected component* (b).

Transmission Line Effects

Transmission line effects can be grouped into several categories. Each imposes a different affect on the transmission.

High frequency cable loss—This is typically referred to as "skin effect," and occurs when a wire is driven with a fast-changing signal such as encountered in digital signals on the leading or trailing edge. It's desirable that the signal change from one state to the other as rapidly as possible. However, as this occurs, the wire stops carrying energy through its entire cross section and current flow is confined to the very outer perimeter of the conductor. Thus the resistance of the cable during the rapid transition appears to be much higher than the actual DC resistance of the conductor. This changes what was a sharp edge at the point of drive into the wire, into a type of stepped pulse at the other end (Figure 3a).

Propagation delay—This is the time required for a signal to travel from one point to another through a conductor. Contrary to popular belief, electricity does not flow through a conductor at the speed of light, but travels more typically somewhere between 50-90% of it, dependent on the materials used in the cable. The result is a very distinct dif-

ference in time between when the signal is received at the end of the cable and when it was transmitted. When using a cable with more than one conductor, or transitioning through a connector block, some of the conductors can become different lengths, resulting in a skew of times between transmission and reception (Figure 3b).

Impedance matching—This is a technique used to control the AC reactance of a line and thereby ensure that the energy put into the transmission is received at the other end. It's usually implemented by using a cable with a signal conductor and a ground conductor running parallel a particular distance from each other over the cable's length. This creates a characteristic impedance. Since this impedance is only true, considering no other losses, if the wire is of infinite length, it's further augmented by resistor termination (in the value of the impedance chosen) at both ends of the cable segment (Figure 4).

Cabled resistance and connector losses—These are due to the material used to make a conductor. A piece of wire has a small amount of resistance per length, and as lengths become long enough to be useful, there's enough resistance to impact the amplitude of the signal received at the end due to the voltage drop on the cable. There's also resistance added at any point of mechanical connection, such as edgecard connectors or backplanes (Figure 5).

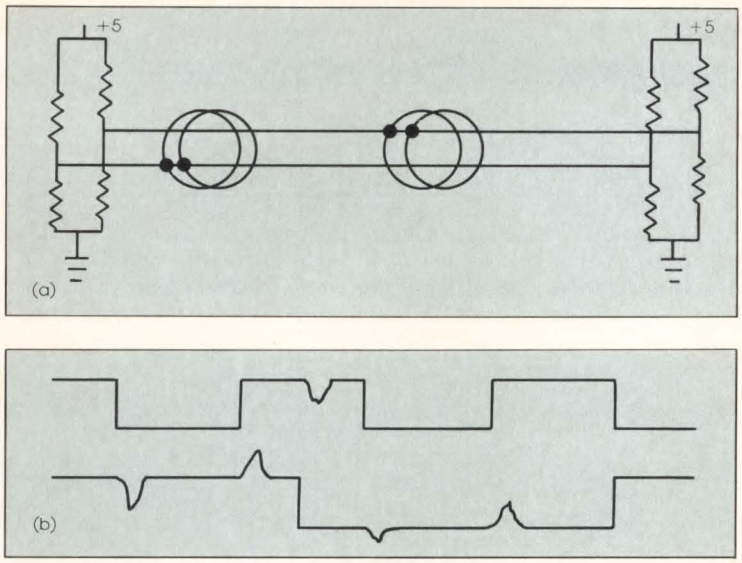


Figure 6—*Crosstalk occurs when lines of flux cut across conductors* (a). *The resulting waveform shapes in the conductors are distorted* (b).

Cross talk—This occurs in a multiple conductor cable when many lines change simultaneously. The changes cause magnetic flux to develop around the changing conductors and can couple, with a transformer action, into another line in the cable that wasn't supposed to change (Figure 6).

Stubs and inductance—Stubs are connections to the transmission line to

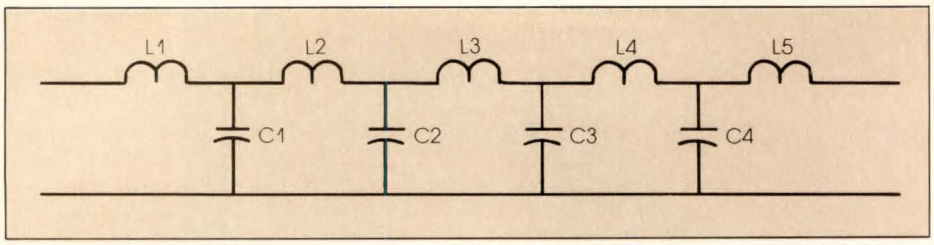


Figure 7—*Wire stubs act as capacitors when connected to a transmission line. The cable impedance caused by the segmenting of the bus is represented by the five inductors, and the stub equivalent capacitances are represented by the capacitors.*

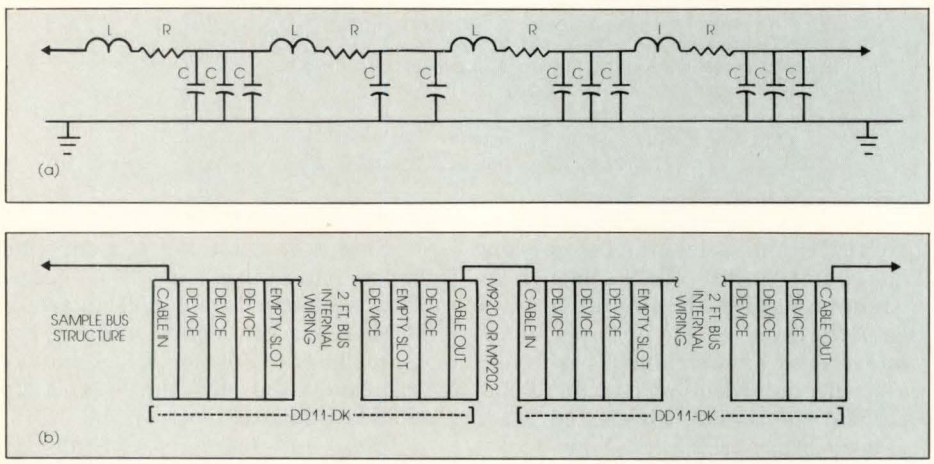


Figure 8—*The electrical representation* (a) *of a sample system bus structure* (b) *shows the complex RLC make-up of an actual system bus.*

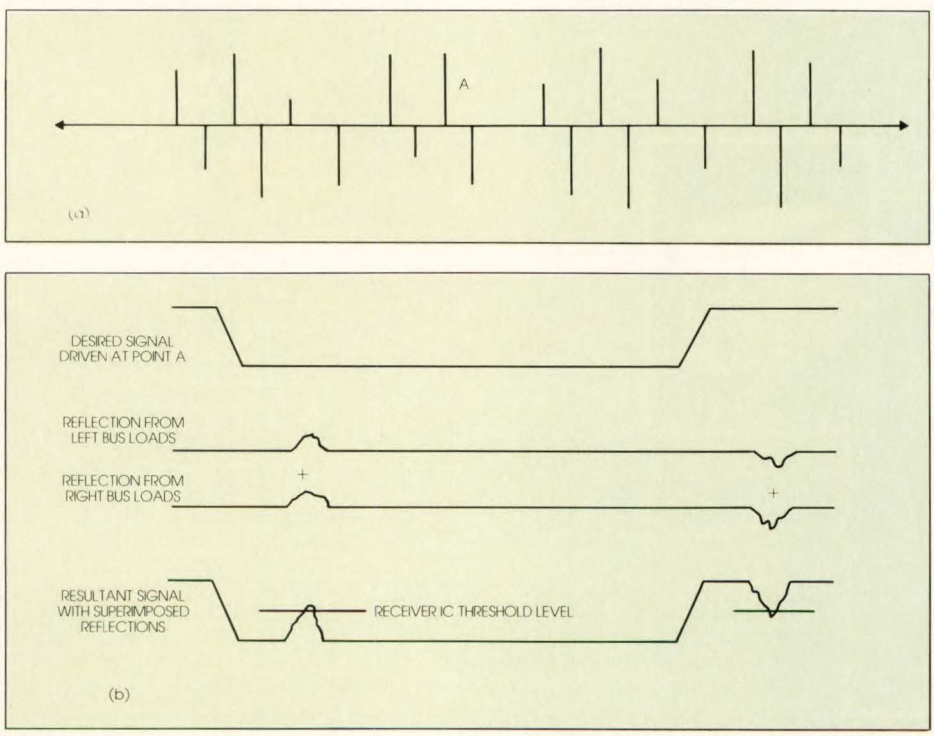


Figure 9—*A complex stub layout* (a) *is the reality behind an actual system bus. These stubs have a substantial effect on bus signals* (b).

link the information on the line to a device that needs to communicate over the line. They are formed of short pieces of wire that connect electrically from the main transmission path to some point on a module, and have the effect of connecting a capacitor to the line. The amount of capacitance is related to the length and mechanical position of the connecting wire. The inductance of cable runs between and within transition blocks, which can have variations, depending on the cable's proximity to metal chassis and racks, needs to be considered in an analysis of signal transitions (Figure 7).

Transmission Line Effects

Figure 8 shows a sample system bus structure and its electrical representation when transmission line effects are considered. The Unibus is a very complex RLC (resistor, inductor, capacitor) network. Fortunately, however, just one item is responsible for the majority of problems encountered.

The bus is no longer a clean piece of wire strung between two terminators with only one driver and receiver. Figure 9a is a representation of the stub layout for the sample bus structure of Figure 8. The stubs are the short pieces of wire connecting the IC's on the modules to the bus conductor.

"Systems using the Unibus range from the low end PDP-11/05 to the VAX 8000 series."

The major effect on the transmitted signal is produced by the stubs. They "de-tune" the transmission line. This occurs because they act as an energy storage point that dumps or absorbs energy during a transition, and as a varying resistance during a transition that causes impedance mismatches.

As shown in Figure 9b, when a signal is asserted (high to low) on the bus, it propagates down the conductor with a stepped wavefront. As this front encounters a stub (capacitive load) that was previously charged to the quiescent voltage of the bus, energy stored in the capacitor is discharged onto the bus. This discharge causes the signal level to rise above the lower level of the wavefront that was initially transmitted, and reflect back toward the transmitting device. If the capacitive load stored more energy than the transmitting device could instantaneously dissipate, its output level also rises. Since this reflection is directed back at the transmitting device from both directions on the bus, the waveform at the driver is composed of the two reflections superimposed. Thus added, it results in a level rise that may cross the

some years ago, the 780 needed to be upgraded to two interleaved memory controllers to handle it. On the surface this seems contradictory, but in reality, while the bus could handle the transfer rate, the devices connected to it could not. With system overhead and the disk transferring simultaneously, this became a problem.

The synchronous architecture of the VAX-11/780 SBI can transfer from 1 to 8 bytes/cycle time. One full cycle typically requires seven or eight 200 nsec cell times, or about 1.4 μsec, which results in a .7 Mbyte/sec. to approximately 5 Mbyte/sec. average throughput, depending on the devices using the bus. The architecture uses the same amount of time to handle a fast device as it does a slow one. In typical applications, approximately a 3 Mbyte/sec. throughput is more realistic.

This in no way degrades the 780 architecture; it is simply mentioned to distinguish theoretical from realistic values.

The Unibus is an asynchronous bus, which means its cycle time varies dependent upon the device using it. The primary speed controlling device is memory. Most MOS memories used require approximately 650 nsec access time and 250 nsec bus overhead time to complete a cycle. This translates to a bandwidth of about 1.1 MHz and, whether doing a 1- or 2-byte transfer, either 1.1 Mbyte/sec. or 2.2 Mbyte/sec. throughput rate. This maximum rate is available to any DMA device on the bus, since DMA is the highest priority for bus use.

In both the Unibus and the SBIbus, interrupts were not considered in the throughput analysis. While they are a very necessary part of the overall operation, they make up only a small portion of total bus activity. An asynchronous bus also wins here, as it will usually handle an interrupt twice as fast as its synchronous counterpart.

Some other major differences between synchronous and asynchronous busses should also be considered.

A synchronous bus gets around transmission line effects by adding deskew to each of its clock cell times, resulting in the time for the operation plus about 100 nsec/clock cell as a minimum. This severely limits the length of the bus. Asynchronous busses must handle these effects only once per cycle, not seven or eight times.

On synchronous systems, degradation in speed of a device very rapidly results in system failures; thus fixes are required that help keep devices operating near peak performance. Conversely, asynchronous device slowdowns in response simply cause the system to slow down to accommodate them. These may take the form of slowed slave responses, illegal interrupts that waste CPU time, or passive bus releases.

Running at its standard DMA transfer rate, a Unibus can move enough data to fill a 700-Mbyte system disk in about 5 min. That is, if the system can accumulate that much data that fast, and if the disk can handle that transfer rate. Since that's more data than is usually handled in a day, it should be of little concern.

The Unibus will move enough data to support virtually any standard application, and most high speed applications. This assumes, however, that the bus is configured to attain its maximum throughput.