If we want our system to be modular and expandable, how should its design accommodate components that the user might add at a later time?

For many years the approach was to provide a way to plug additional printed circuit boards into the main "motherboard" that holds the CPU, memory, and the initial collection of I/O components.

The socket on the motherboard connects the circuitry on the add-in card to the signals on the motherboard that allow the CPU to communicate with the add-in card.

These signals include power and a clock signal used to time the communication, along with the following.

* Address wires to select different communication end points on the add-in card.

The end points might include memory locations, control registers, diagnostic ports, etc.

* Data wires for transferring data to and from the CPU.

In older systems, there would many data wires to support byte- or word-width data transfers.

* Some number of control wires that tell the add-in card when a particular transfer has started and that allow the add-in card to indicate when it has responded.

If there are multiple slots for plugging in multiple add-in cards, the same signals might be connected to all the cards and the address wires would be used to sort out which transfers were intended for which cards.

Collectively these signals are referred to as the system bus.

"Bus" is system-architect jargon for a collection of wires used to transfer data using a pre-determined communication protocol.

Here's an example of how a bus transaction might work.

The CLK signal is used to time when signals are placed on the bus wires (at the assertion edge of CLK) and when they're read by the recipient (at the sample edge of the CLK).

The timing of the clock waveform is designed to allow enough time for the signals to propagate down the bus and reach valid logic levels at all the receivers.

The component initiating the transaction is called the bus master who is said to "own" the bus.

Most buses provide a mechanism for transferring ownership from one component to another.

The master sets the bus lines to indicate the desired operation (read, write, block transfer, etc.), the address of the recipient, and, in the case of a write operation, the data to be sent to the recipient.

The intended recipient, called the slave, is watching the bus lines looking for its address at each sample edge.

When it sees a transaction for itself, the slave performs the requested operation, using a bus signal to indicate when the operation is complete.

On completion it may use the data wires to return information to the master.

The bus itself may include circuitry to look for transactions where the slave isn't responding and, after an appropriate interval, generate an error response so the master can take the appropriate action.

This sort of bus architecture proved to be a very workable design for accommodating add-in cards as long as the rate of transactions wasn't too fast, say less than 50 Mhz.

But as system speeds increased, transaction rates had to increase to keep system performance at acceptable levels, so the time for each transaction got smaller.

With less time for signaling on the bus wires, various effects began loom large.

If the clock had too short a period, there wasn't enough time for the master to see the assertion edge, enable its drivers, have the signal propagate down a long bus to the intended receiver and be stable at each receiver for long enough before the sample edge.

Another problem was that the clock signal would arrive at different cards at different times.

So a card with an early-arriving clock might decide it was its turn to start driving the bus signals, while a card with a late-arriving clock might still be driving the bus from the previous cycle.

These momentary conflicts between drivers could add huge amounts of electrical noise to the system.

Another big issue is that energy would reflect off all the small impedance discontinuities caused by the bus connectors.

If there were many connectors, there would be many small echoes which would could corrupt the signal seen by various receivers.

The equations in the upper right show how much of the signal energy is transmitted and how much is reflected at each discontinuity.

The net effect was like trying to talk very fast while yelling into the Grand Canyon.

The echoes could distort the message beyond recognition unless sufficient time was allocated between words for the echoes to die away.

Eventually buses were relegated to relatively low-speed communication tasks and a different approach had to be developed for high-speed communication.