dress, an
perform
the syste
other words, it can "turn off" the system's master processor, take control of all (or most) of its lines, and perform temporarily as the system's master processor.

As microcomputer technology evolved, it became apparent to the standard's designers that the S-100 bus was capable of "lending" control for more than just memory access operations. In fact, any "temporary" master processor could take over the bus for any kind of bus access at all (within limitations, of course) and freely drive any passive system components (called bus slaves) that it desired. To indicate that this broader form of temporary master access was available with the S-100 bus, the name TMA (Temporary Master Access) was coined, to replace the less descriptive name DMA.

The designers of the 696 standard further enhanced the TMA process by adding four new lines to the S-100 bus that allowed up to 16 temporary masters to vie for control of the system bus at the same time. These four lines, called TMA0*-TMA3*, allow prioritized arbitration among any temporary masters that simultaneously request bus access. This method of arbitration requires that each temporary master have its own arbitration controller[1] in order to establish (and assert) its priority. In addition, each temporary master must have a unique priority.

Basically, what happens is this: Any temporary master wishing to perform TMA will place its priority code on the TMA bus by placing the complement of its priority code on the four TMA lines TMO*-TMA3*. These TMA bus lines (being negative-logic) are normally pulled up to a logic one, so that each line can be made ACTIVE by pulling it down to a logic zero. In this manner, each of the four lines is WIRE OR'ed to each temporary master, so that more than one temporary master may assert its priority on the TMA bus at the same time. Once a "requesting" temporary master has asserted its code on the TMA bus, it READS the TMA bus, then compares what it sees with its own priority code. If there is no difference, then it assumes that it has "won" the arbitration. If there is a difference, then it must compare the bits of the TMA bus with its own priority code to determine if there is a higher-priority temporary master asserted on the TMA bus. If the temporary master determines that it has "lost" the arbitration, then it will continue comparing the TMA bus. If the temporary master determines that it has "lost" the arbitration, then it will continue comparing the TMA bus with its priority code, and wait until it has the highest asserted priority before it will attempt TMA. The ac-