PMIC Digital Control Made Simple Using a Single Wire

By Tom Karpus, Semtech Corporation

Digital control has permeated all aspects of the integrated circuit business, including the analog power management sector. Inter-Integrated Circuit (I^2C), System Peripheral Interface (SPI), Power Management Bus (PMBus), and other interfaces have been adapted to the needs of power management chips (PMIC), at the expense of requiring at least three extra pins and a large chunk of die area to implement a digital bus that is in many cases overkill for a simple regulator chip. In many cases, simple power management ICs don't require the read-write capability provided by these bi-directional data buses. A simple write-only protocol would be sufficient to program these devices without the need for read-back capability.

Simple Pulse - Counting

While no widely accepted industry standard write-only interconnect has emerged, the need has been addressed by vendors who have developed a variety of single-wire interfaces to fill the void.

Several IC companies have developed simple interface protocols to perform this function. The simplest of these interfaces employs pulse counting as a way to determine what control register bits should be set. In the typical pulse counting scheme, the device monitors an input pin and counts how many rising edges it receives before the pulsing is paused.

Once a count is determined, the chip compares the count to a look-up table of settings and puts the device in the desired state. This system works very well and is somewhat error resistant, but it is also extremely limited in the number of possible states that can be practically set. For example, consider a programmable regulator with 32 output settings. Each setting would need to have a specific pulse count assigned to it. If an additional output were added to this device, another 32 pulse counts would be needed.

Note that any additional functionality forces the available pulse count to increase. If each pulse requires a fixed amount of time, the total time to make a single setting could get too large to be practical if multiple functions are included in a single chip. An interface that avoids this serial counting approach is needed if multiple functions are to be integrated into a single part.

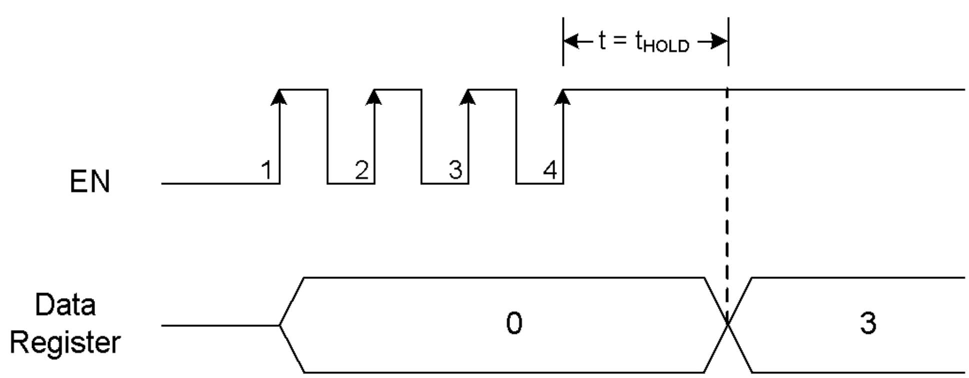

Figure 1 – Timing Diagram for Simple Pulse Counting Systems

Other Interfaces

Some IC manufacturers have developed more intelligent single-wire interfaces that allow multiple functions to be controlled in a single chip. These interfaces typically provide more functionality than simple pulse counting systems while employing pulse counting to maintain the simplicity of implementation. Two sets of pulses are typically used – one to pick a register location and one to determine the setting for that register. This approach is theoretically more robust than simple pulse counting, but there are some limitations inherent to these systems as well.

In one such interface protocol, the number of possible settings is limited to 16 so that the total number of pulses counted is not excessively long. It also uses pulse counts greater than 16 to determine the address that is being written. The resulting data transmission for such a system requires the host processor to transmit at least 17 pulses for the shortest word transmission.

In a similar system, pulses are counted to determine an address and data word. This system requires fewer pulse transmissions than a strict pulse counting system, but it is limited to two address bits (four total addresses) and six data bits. This is adequate for most simple power management devices, but it falls short when multi-function integration combines several power management functions into a single chip.

Another concern for these systems involves timing requirements. Many mobile systems employ system processors that are fully utilized while a device is activated. These systems depend on interrupts to perform high priority tasks such as answering a page in a mobile phone application. In these systems, the interrupt can occur during a pulse transmission to the peripheral device, causing the pulse transmission to remain at the same level until the interrupt is complete and the processor can get back to the pulse transmission routine. If this interrupt is longer than the interface's hold time, the pulse transmission can be ignored or, worse yet, register an erroneous value. In these cases, multiple software writes to add redundancy are needed to avoid such errors.

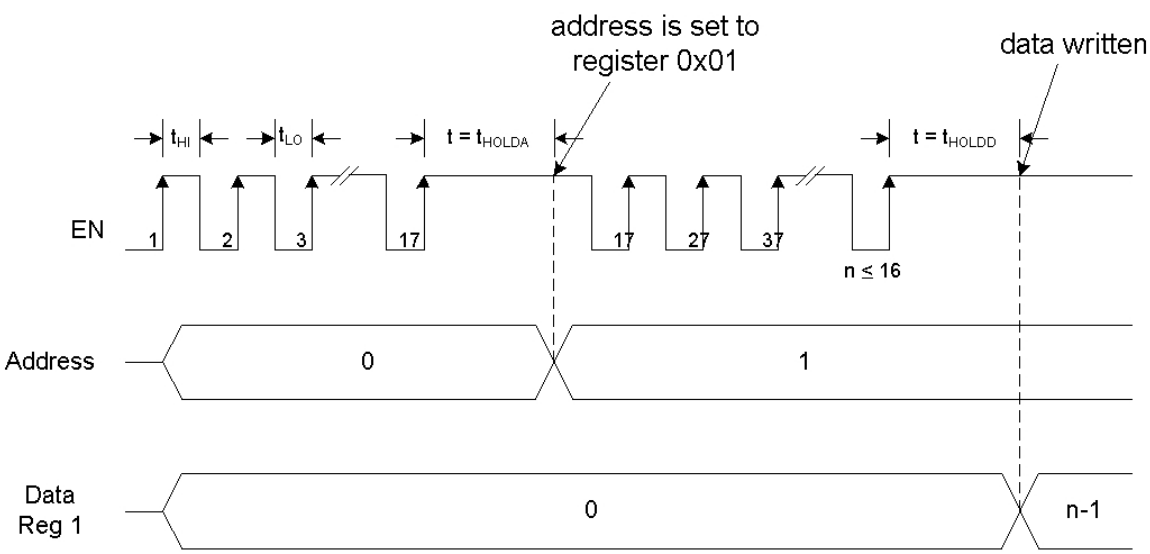

Figure 2 – Timing Diagram for Pulse Counting
Interface with Address and Data

Hybrid Pulse Counting-Data Register approach

More recently, a new approach has emerged that combines the benefits of pulse counting and the flexibility of data registers as used in the multi-wire interface protocols. This single-wire interface is designed to write data to control registers using a form of pulse counting to determine the data to be written. One example of this is the SemPulse® interface from Semtech. SemPulse splits the pulse count into two distinct values. The first count is used to determine the register number. In its current form, SemPulse can accommodate up to 32 registers (this could be expanded by increasing the maximum count number in future). After the first set of pulses is sent, the controller pauses for 5ms before sending a second set of pulses that provide the data to be written to the designated register. Each register has a six-bit length, so the total number of pulses in this set can reach a total of 64. The pulse width minimum is 0.75µs (or 1.5µs for the high and low time), so the total time to transmit the worst case address (32 counts) and data (64 counts) with the hold time between the two sets (5ms) is less than 5.5ms.

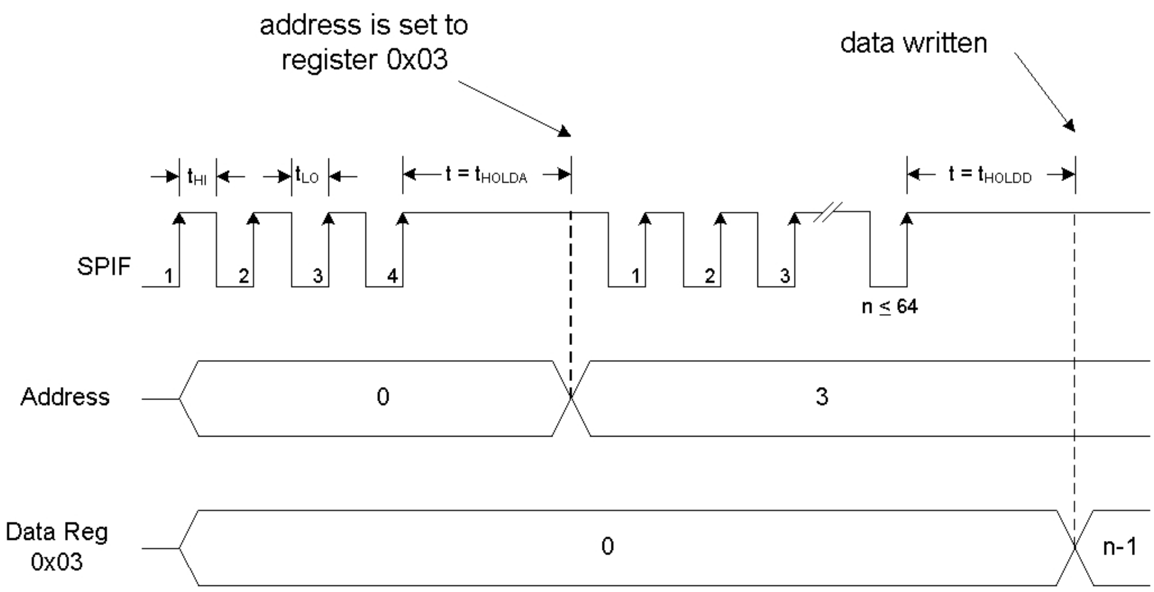

Figure 3 – SemPulse Timing Diagram Example

Conclusion

The drive to implement digital controls in PMICs has led to a variety of single-wire programming interfaces to provide communication without the overhead of three-wire industry standard interfaces. The key to the right interface is to combine the design simplicity with the features needed to enable power management IC expandability. The capability to include multiple registers and six bits of programmability per register make an ideal combination for new power management units that integrate multiple functions into a single device. Programming simplicity is not compromised as simple pulse counting is the basic format of the data transmission on this bus.

Tom Karpus is Director of Handheld Systems and Applications Engineering for Semtech Corporation based in Morrisville NC. He received a BSEE degree from Michigan Technological University and an MSEE degree from Syracuse University. He is also a graduate of the GE Advanced Course in Engineering and Edison Engineering programs.

Semtech Corporation
200 Flynn Road, Camarillo, CA 93012
Phone (805) 498-2111 Fax: (805) 498-3804 Web: www.semtech.com