

Moving averager rejects noisy outlier values

[David Vincenzoni](#) - June 23, 2016

Measurements made in a noisy environment can exhibit sporadic disturbances. This Design Idea describes a digital circuit that removes outlier spikes without compromising bandwidth.

Smart moving average

The moving average is a process that continuously computes the average over N samples of data flowing through a FIFO (First In First Out) buffer. Every new sample added to the buffer will remove the oldest sample used to compute the previous mean value.

The *smart moving-average* is a variation on the theme; a digital circuit (**Figure 1**) where the idea is to collect the latest N measurements like a classical moving average, but new data will be added into the buffer (Sx) only if its value is within set limits of the actual average of the previous N samples.

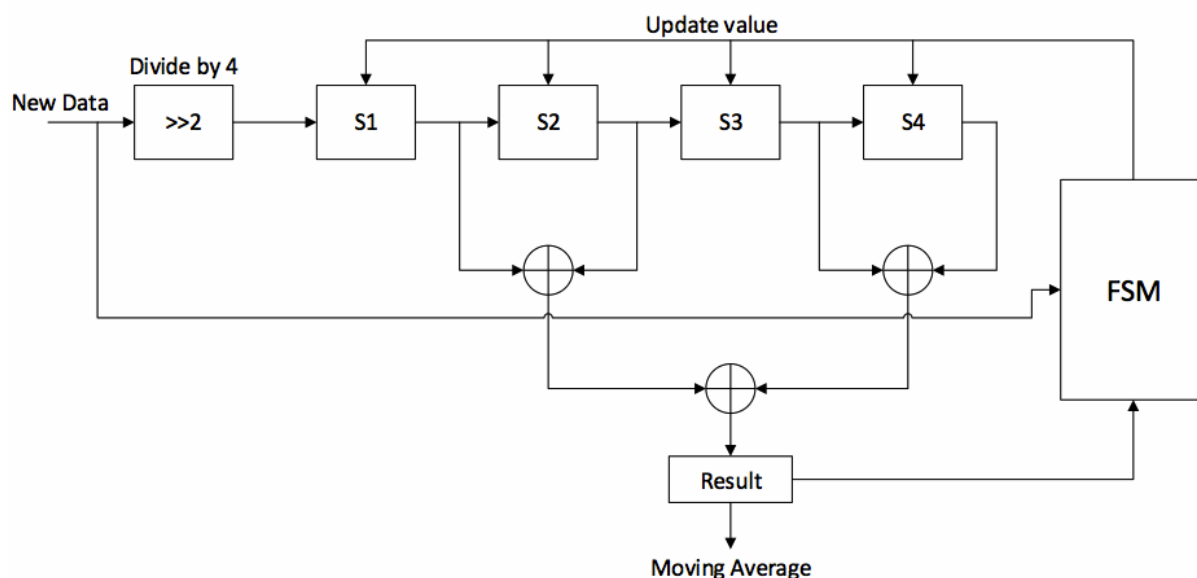


Figure 1 The smart moving-average circuit determines when to reject new data.

The Finite State Machine (FSM) of **Figure 1** manages this task. Every new data sample is compared against a maximum and minimum which depend on the current mean value. The new value is



discarded when it is beyond the set limits.

Keep the number of stored samples (S_x) to a power of two to minimize of the size of the adders and avoid a general-purpose divider. If we use 2^p samples, we can shift-right the data input by p bits performing a zero-cost division, and the adders' size is reduced by the same p bits.

In this example, we have four (2^2) samples and we right-shift the input data by 2. The samples will be of size $M-2$, where M is the bus size of input data.

The computation circuit is formed by three full adders with carry features. The result is used by the FSM for checking the new input data.

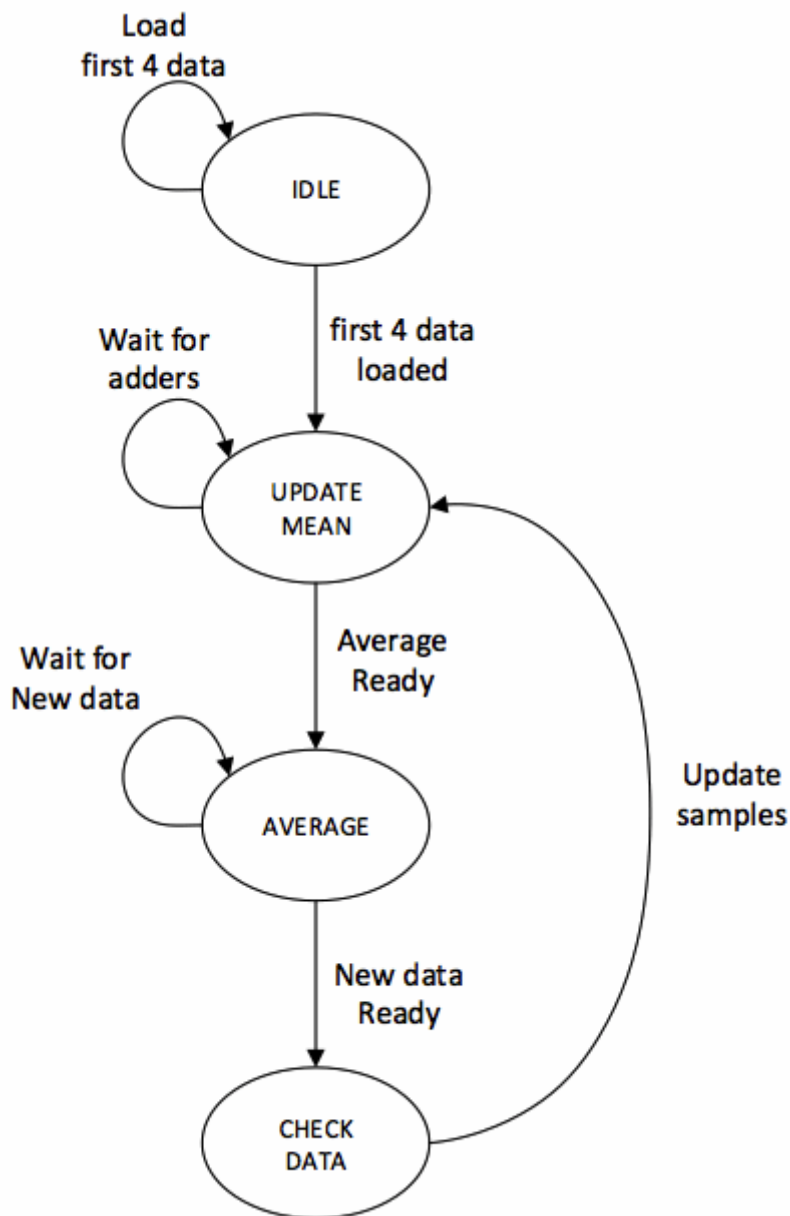


Figure 2 FSM state diagram

At startup, since there is no mean value to be used as a comparison point, the first data are stored in the Sx buffers; then the mean value is computed. The FSM steady state is *AVERAGE*, shown in **Figure 2**. Here the FSM is waiting for new data input that will be checked against the mean value on the next state: *CHECK DATA*. This state's implementation depends on what we are going to measure. In the case of temperature data, there is relatively slow variation; we do not expect new data to be very dissimilar from previous, nor from the mean. For this specific case, the data comparison can be done in the following way:


1.  Only the most-significant bits of the mean and input (MSb) will be compared.
2. If the values are within a defined difference range, the new data will be stored and the mean value updated.
3. If the values have a greater difference, the new data will be discarded and the mean value kept.

Figure 3 shows an example comparison table. The range of possible values for the new data input is divided into four rows; only two MSb are used in the comparison (for some conditions the third bit is also checked). For example, when the new data MSb are "00", it is accepted if the mean value of the previous four samples has MSb "00" or "01". Otherwise, the data is discarded.

Mean Data	00	01	10	11
00				
01				
10				
11				

Figure 3 Comparison scheme

The number of elements used to compute the moving average can depend on the occurrence of incorrect data. In the case of sporadic events, this can be low (e.g., four); if the occurrence is high, then it will be necessary to increase the buffer size to eight or 16 elements.

—[David Vincenzoni](#) is R&D Design Manager at STMicroelectronics, responsible for the design and verification of new chips for Broadband Power Line Modems and for new families of devices for industrial applications.