

Energy Metering on Hot Swap and Power Monitor Devices

by Paul O'Sullivan

INTRODUCTION

Analog Devices, Inc., has a range of hot swap and power monitor devices, some of which include energy metering functionality. These devices include the [ADM1075](#), [ADM1276](#), [ADM1278](#), [ADM1293](#), and [ADM1294](#).

Each of these devices can measure input voltage as well as output current. The 12-bit input voltage (V_{IN}) and 12-bit output current (I_{OUT}) measurement values are multiplied together to give the input power value (P_{IN}). This multiplication is accomplished using fixed point arithmetic, and produces a 24-bit value. The result is in 12.0 format, meaning that there is no fractional part.

The PMBus™ Application Profile for Hot Swap Controllers specifies that the user can read individual power samples or use an energy metering functionality on chip. The advantage of energy metering functionality on chip is that the host processor does not have to poll the power monitor continuously to read power samples. Individual power samples are accumulated on chip; thus, the user can read from the power monitor via PMBus intermittently instead, which frees up the I²C bus for other transactions.

INSTANTANEOUS POWER MEASUREMENTS

The 24-bit power value can be read using the READ_PIN_EXT command. The 16 MSBs of the 24-bit value are used as the return value for P_{IN} , using the READ_PIN command. The input power registers return a twos complement binary value; therefore, the MSB is always 0 for a positive power measurement.

ENERGY METERING

Each time a power calculation is completed, the 24-bit power value is added to a 24-bit energy accumulator register. This 24-bit power value is a twos complement representation as well; therefore, the MSB is always 0. Each time this energy accumulator register rolls over from 0x7FFFFFFF to 0x000000, a 16-bit rollover counter is incremented. The rollover counter is straight binary, with a maximum value of 0xFFFF before it rolls over.

Note that the [ADM1293/ADM1294](#) utilize the full 24 bits of the energy accumulator and roll over at 0xFFFFFFFF rather than at 0x7FFFFFFF. Adjust the calculations for READ_EIN and READ_EIN_EXT accordingly.

A 24-bit straight binary power sample counter is also incremented by 1 each time a power value is calculated and added to the energy accumulator.

These registers can be read back using one of two commands, depending on the level of accuracy required for the energy accumulator and the desire to limit the frequency of reads from the power monitoring device.

- The READ_EIN command returns the top 16 bits of the energy accumulator, the lower 8 bits of the rollover counter, and the full 24 bits of the sample counter (see Table 1).
- The READ_EIN_EXT command returns the full 24 bits of the energy accumulator, the full 16 bits of the rollover counter, and the full 24 bits of the sample counter (see Table 2). The use of the longer rollover counter means that the time interval between reads of the device can be increased from seconds to minutes without losing any data.

A bus host can read these values, and by calculating the delta in the energy accumulated, the delta in the number of samples, and the time delta since the last read, the host can calculate the average power since the last read, as well as the energy consumed.

The time delta is calculated by the bus host based on processor time stamps, and is not provided by the Analog Devices power monitor.

To avoid loss of data, the bus host must read at a rate that ensures the rollover counter does not wrap around more than once, and if the counter does wrap around, that the next value read for P_{IN} is less than the previous one.

READ E_{IN} REGISTER

Read the energy metering registers in a single operation to ensure time consistent data.

Table 1. Bit Descriptions for READ_EIN

Bits	Bit Name	Description	Reset	Access
[47:24]	SAMPLE_COUNT	This is the total number of P _{IN} samples acquired and accumulated in the energy count accumulator. This is an unsigned 24-bit binary value. Byte 5 is the high byte, Byte 4 is the middle byte, and Byte 3 is the low byte.	0x000000	R
[23:16]	ROLLOVER_COUNT	Number of times that the energy count has rolled over from 0x7FFF to 0x0000. This is an unsigned 8-bit binary value.	0x00	R
[15:0]	ENERGY_COUNT	Energy accumulator value in PMBus direct format. Byte 1 is the high byte, and Byte 0 is the low byte. Internally, the energy accumulator is a 24-bit value, but only the most significant 16 bits are returned with this command. Use the READ_EIN_EXT register to access the nontruncated version.	0x0000	R

READ E_{IN} (EXTENDED) REGISTER

Read the extended precision energy metering registers in a single operation to ensure time consistent data.

Table 2. Bit Descriptions for READ_EIN_EXT

Bits	Bit Name	Description	Reset	Access
[63:40]	SAMPLE_COUNT	This is the total number of P _{IN} samples acquired and accumulated in the energy count accumulator. This is an unsigned 24-bit binary value. Byte 7 is the high byte, Byte 6 is the middle byte, and Byte 5 is the low byte.	0x000000	R
[39:24]	ROLLOVER_EXT	Number of times that the energy count has rolled over from 0x7FFFFFFF to 0x000000. This is an unsigned 16-bit binary value. Byte 4 is the high byte, and Byte 3 is the low byte.	0x0000	R
[23:0]	ENERGY_EXT	Extended precision energy accumulator value in PMBus direct format. Byte 2 is the high byte, Byte 1 is the middle byte, and Byte 0 is the low byte.	0x000000	R

POLLING TIME EXAMPLE

The analog-to-digital converter (ADC) conversion time can be calculated from the data sheet specifications. For example, on the [ADM1278](#), the typical conversion time for one sample of I_{OUT} is 144 μs and one sample of V_{IN} is 64 μs. Therefore, the typical ADC conversion time for the [ADM1278](#) with I_{OUT} and V_{IN} sampling enabled is 208 μs with averaging disabled. With 128× averaging, the typical ADC conversion time increases to 26.624 ms (that is, 208 μs × 128).

For the purposes of this example, assume that there is an average power reading of 700 W. In direct format code, the power is returned as

$$Power = 6123 \times R_{SENSE} \times 700 \text{ W} \times 10^{-2}$$

See the PMBus direct format equations in the [ADM1278](#) data sheet.

Assuming that R_{SENSE} = 0.25 mΩ,

$$Power = 6123 \times 0.25 \text{ m}\Omega \times 700 \text{ W} \times 10^{-2} = 10,715.25 = 10,715$$

The energy accumulator is a 24-bit, twos complement value. The READ_EIN command returns the 16 MSBs of the energy accumulator register (ENERGY_COUNT). The MSB is always 0 for positive power values. Therefore, ENERGY_COUNT rolls over after the accumulated power reaches $2^{(16-1)} = 32,768$ (direct format code). For the given example, the energy accumulator rolls over after $32,768/10,715 = 3.058$ ADC samples.

The rollover counter is an 8-bit value for the READ_EIN command. This 8-bit value is a straight binary number corresponding to a rollover value of $2^8 = 256$. The rollover counter increments once each time the energy accumulator rolls over. Assuming the READ_EIN command is used, the number of ADC samples required for the rollover counter to wrap around is $256 \times 3.058 = 782.8$ samples, which gives a total rollover time of $782.8 \times 208 \mu\text{s} = 0.16$ seconds for 1× averaging and $782.8 \times 26.624 \text{ ms} = 20.84$ seconds for 128× averaging.

The polling time can be greatly increased (by a factor of 256) by using the READ_EIN_EXT command instead, which incorporates a 16-bit rollover counter. The rollover times for the examples calculated above increase to 41.7 seconds for 1× averaging and 5335.4 seconds = 88.9 minutes for 128× averaging.

In summary, the rollover times (in seconds) can be calculated with the following equations:

$$READ_EIN \text{ Rollover} = \frac{2^{15}}{ADC \text{ Power Code}} \times ADC \text{ Conversion Time} \times 2^8$$

$$READ_EIN_EXT \text{ Rollover} = READ_EIN \text{ Rollover} \times 256$$

INDEPENDENT POWER AVERAGING

The examples shown in the Polling Time Example section assume that power averaging is configured the same as voltage and current (VI) averaging in the PMON_CONFIG register. However, this may not always be the case because the [ADM1278](#) and [ADM1293/ADM1294](#) allow the user to set power averaging independently.

In effect, the VI averaging sets how many ADC samples are taken with a single ADC trigger. Because the power value is a calculation, the power averaging figure determines how this calculation is performed and when the result is added to the output registers. For example, if VI averaging = 128× and power averaging = 1×, a single ADC trigger initiates 128 VI samples. The average voltage and current values over the 128 samples are returned to the READ_VIN and READ_IOUT registers, but all 128 power measurements are added to the energy accumulator and the sample count increments by 128.

By contrast, if power averaging is also set to 128×, only a single averaged power value is added to the energy accumulator at the end of the 128 samples and the sample count increments by 1.

Therefore, the power averaging value determines how often a power value is added to the energy accumulator and impacts how often it overflows. Setting a larger number of power averages means the polling time between READ_EIN commands can be increased. The cost of increasing the power averaging is a small decrease in measurement accuracy and resolution.

In summary, the required polling time to ensure no data is lost is primarily dependent on the magnitude of the power value being measured, the power averaging value, and whether the READ_EIN or READ_EIN_EXT command is used.

I²C refers to a communications protocol originally developed by Philips Semiconductors (now NXP Semiconductors).