

FPGA-Based Spectral Envelope Preprocessor for Power Monitoring and Control

Zachary Remscrim, James Paris, Dr. Steven B. Leeb, Dr. Steven R. Shaw,
Sabrina Neuman, Christopher Schantz, Sean Muller, Sarah Page

Abstract—Smart Grid and Smart Meter initiatives seek to enable energy providers and consumers to intelligently manage their energy needs through real-time monitoring, analysis, and control. We have developed an inexpensive FPGA implementation of a spectral envelope preprocessor. This FPGA permits cost-effective and richly detailed power consumption monitoring for individual loads or collections of loads. It permits a flexible trade-off between data transmission, storage, and computation requirements in a power monitoring or control system. The information from the FPGA can be used to coordinate the operation of power electronic controls.

I. BACKGROUND

Power electronics and power electronic controls are proliferating in consumer electronics. There is an increasing expectation that advanced power conditioning electronics will play a role in managing and coordinating power consumption not simply for a particular load, e.g., a variable speed drive in an air conditioning plant, but also in response to the dynamic needs and capability of the utility system. Loads that can respond not only to their own tasking but also to the needs of the utility are implicit in many visions of a “smart grid.”

There is a need for flexible, inexpensive metering technologies that can be deployed in many different monitoring scenarios. Individual loads may be expected to compute information about their power consumption. They may also be expected to communicate information about their power consumption through wired or wireless means. Switch gear like circuit breaker panels may eventually be expected to provide detailed submetering information for different loads on different breakers or clusters of breakers and controls. New utility meters will need to communicate bidirectionally, and may need to compute parameters of power flow not commonly assessed by most current meters.

Both vendors and consumers will likely find innumerable ways to mine information if made available in a useful form. However, appropriate sensing and information delivery systems remain a chief bottleneck for many applications, and metering hardware and access to metered information will likely limit the implementation of new electric energy conservation strategies in the near future. The U.S. Department of Energy has identified “sensing and measurement” as one of the “five fundamental technologies” essential for driving the creation of a “Smart Grid” [7]. Consumers will need “simple, accessible... , rich, useful information” to help manage their electrical consumption without interference in their lives [7].

Digital technology has been in use for over 20 years for measuring and metering power flow. A few examples from an enormous array of metering and measurement approaches for monitoring power can be found in [8]–[11], and the references

in these documents. Digital power monitoring has also made its way to the “plug” and “power strip” level, e.g., see [13]. Many different schemes for storing or communicating information are still under exploration—see [12] and its references for example. Most of these solutions deploy computation hardware that is either substantially complicated in both hardware and firmware, e.g., [10] where a DSP and a micro-controller work together to coordinate computation of real, reactive, and apparent power, or where fully integrated custom chips are specifically developed for a particular application.

The “spectral envelope” representation of observed current and voltage signals used in the non-intrusive load monitor or NILM [2], [14], [15] can be a very flexible basis for computing and tracking all sorts of useful metrics about power consumption. Spectral envelopes estimate real and reactive power consumption and harmonic contents. Even for waveforms with substantial high frequency content, the time-varying spectral envelopes can be relatively band-limited. Thus, spectral envelopes are often a natural way to “compress” useful data about load current and power consumption, easing communication requirements.

This paper describes an integer-arithmetic implementation of a spectral envelope preprocessor for an inexpensive FPGA. The spectral envelope FPGA coordinates data acquisition and computes spectral envelopes without the need for floating-point computation. Hence, the FPGA can be used in a two-IC suite (with an analog-to-digital converter) to inexpensively acquire load consumption data. This data minimizes the need for “downstream” computation later in the signal processing workflow. Of course, further computation can be used to track, trend, price, or control energy consumption. The FPGA can directly control communication as well, providing wired or wireless access, or storage on flash memory or other media. The spectral envelope FPGA is a cost-effective building block that can be used to enable a huge array of power monitoring and control applications, ranging from the individual load up through the breaker panel or utility service entry level and beyond. It can be used to provide necessary power consumption information for coordinating power electronic controls.

This paper describes the design of this key building block and shows results from a prototype. The next section describes the approach for using spectral envelopes for load identification and how this data is computed by the preprocessor. The following section presents the FPGA-based spectral envelope preprocessor and the techniques used to implement the preprocessor cost-effectively. Finally, the paper examines the performance of prototype hardware and describes further enhancements for expanded monitoring applications.

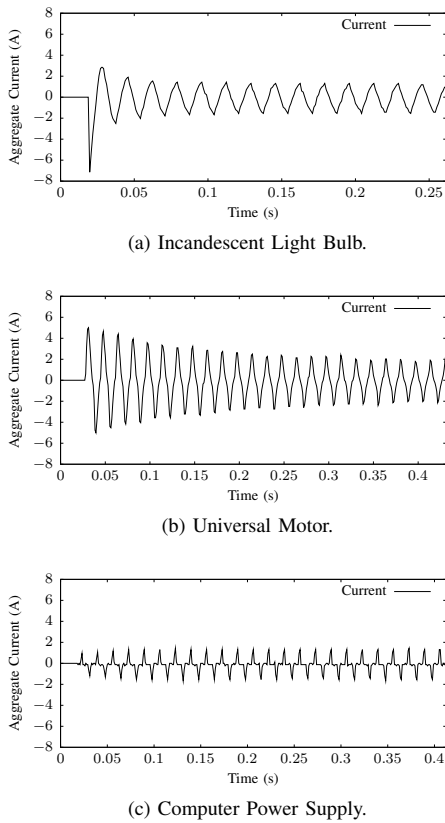


Fig. 1. Turn-on transients for an incandescent light bulb, a universal motor, and a computer power supply.

II. UTILITY OF SPECTRAL ENVELOPES

Typical turn-on current transients are shown in Fig. 1 for an incandescent lamp, a universal motor (as in a vacuum cleaner or hand tool), and a personal computer. Dynamic changes in the power and harmonic consumption of a load, e.g., during turn-on or turn-off transients, can serve as a fingerprint for identifying load operation [1]. For example, an observed turn-on transient or exemplar from a training observation produced by one of a collection of loads can be used to identify the load in an aggregate current measurement. An analogous procedure can be performed using turn-off transients. All that is needed, in principle, to determine the operating schedule of a collection of loads is to record the aggregate current drawn by those loads and then match each observed transient to the turn-on or turn-off fingerprint of a particular load in the collection.

However, direct examination of current waveforms may not be practical for many stages of some applications, including many components in energy scorekeeping, monitoring, or conservation systems. Direct operations on the current waveform require sample rates adequate to capture the highest harmonic content of interest [5]. In some metering, monitoring, and control applications it is more practical to either store data for a period of time and examine it later, or to transmit data to another location for interpretation and control. In either of these cases, it is convenient to have a useful representation of the data that avoids excessive storage or communication bandwidth requirements.

Spectral envelopes provide a useful separation between data collection and analysis. They permit a small, inexpensive system with low processing power to collect data continuously. A system with larger available processing power, potentially physically remote from the data collection front-end, can either review a storage device at a later time or continuously process a relatively low bandwidth information stream over a convenient communication channel, wired or wireless.

A. Spectral Envelope Definition

The spectral envelopes of current represent the harmonic content of the input waveform for each line-locked period of the service voltage. Given N samples $i[n]$ of a waveform $i(t)$ over one period, the samples can be expressed in terms of their spectral content by

$$i[n] = \frac{1}{N} \sum_{k=0}^{N-1} \left(a_k \sin\left(\frac{2\pi kn}{N}\right) + b_k \cos\left(\frac{2\pi kn}{N}\right) \right) \quad (1)$$

where the spectral envelopes a_k and b_k for that period are defined by

$$a_k = \sum_{n=0}^{N-1} i[n] \sin\left(\frac{2\pi kn}{N}\right) \quad (2)$$

and

$$b_k = \sum_{n=0}^{N-1} i[n] \cos\left(\frac{2\pi kn}{N}\right). \quad (3)$$

Here, k denotes the multiple of the line frequency to which a particular spectral envelope corresponds; for example, $k = 1$ corresponds to the 60 Hz component and $k = 3$ to the 180 Hz component. The values of these spectral envelopes are calculated for each period of the line voltage; the values at period m will be denoted $a_k[m]$ and $b_k[m]$. With this definition, spectral envelopes can naturally be calculated from the real and imaginary parts of the Discrete Fourier Transform (DFT) [16] of $i[n]$ over each period of the line voltage.

B. Spectral Envelope Compression

Given the DFT coefficients over one period of the service voltage, it is possible to exactly reconstruct or preserve all of the information in the raw current samples over that period. Of course, simply recording all of the DFT coefficients will not reduce the data handling requirements—if there are N samples of current, the DFT will produce N DFT coefficients. While it may appear that storing the complex DFT coefficients would take twice as much space as storing the raw current samples, because they are complex numbers, note that the DFT will be symmetric (because the raw current samples are real), so only $\frac{N}{2}$ of the N complex numbers need to be stored. The data requirements for storing all meaningful DFT coefficients is the same as storing all raw current samples at the same level of precision.

In situations where the significant or relevant current drawn by an electrical load consists predominantly of the fundamental frequency (the frequency of the service voltage, for

example, 60 Hz) and a small collection of the line frequency harmonics (such as the 3rd, 5th, 7th), it is reasonable to record, for each period of the service voltage, only a few DFT coefficients [2]. These relatively few DFT coefficients can be used to reconstruct the original current samples with a relatively small error. Furthermore, the time varying values of the DFT coefficients themselves can be used directly as fingerprint signatures for the loads, or to track important quantities associated with load operation, with reasonable accuracy.

Because only a few DFT coefficients may be needed to accurately represent the current waveforms, this “spectral” approach to the representation of the waveforms serves as a form of compression. Consider current and voltage samples that are collected at a 7.68 KHz sample rate. The sampling rate is chosen to be sufficiently high to provide adequate anti-aliasing without filtering effects and to provide accurate detection of voltage zero crossings to enable line-locked data collection. This corresponds to 128 samples per 60 Hz line-cycle ($N = 128$), and so 64 meaningful complex DFT coefficients need to be stored to perfectly reconstruct arbitrary current samples. However, for many applications, including load monitoring for diagnostics, only a limited number of DFT coefficients need be stored. In the prototype system discussed in this paper, just 4 coefficients, or 6.25% of the full set of already compact harmonically-related DFT coefficients, were needed.

Of course, other reductions of the data could be applied, e.g., simply recording average aggregate real power once per second (where the average is taken over each second interval), leads to further compression. Such data would not reflect the detailed short term variations that would occur in real power, nor would it reflect any of the behavior of the higher harmonics. Time-varying DFT coefficients or spectral envelopes strike a balance between the need to store or transmit as little data as possible and the need to maintain sufficiently detailed data to perform load monitoring.

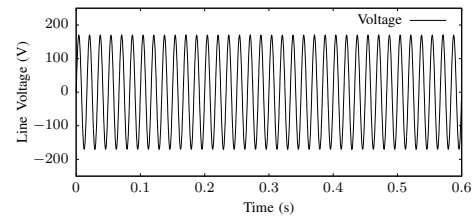
C. Physical Interpretation of Spectral Envelopes

Spectral envelopes can be directly interpreted as other meaningful physical quantities under some conditions. In situations where the utility voltage is relatively harmonic free and “stiff” (with constant peak amplitude), the real part of the first DFT coefficient of the current waveform, or fundamental frequency spectral envelope a_1 , scales to “real power” in steady-state. Similarly, the imaginary part b_1 scales to “reactive power.”

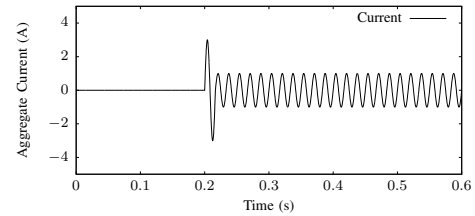
If the voltage is not stiff or harmonically pure, then it is still possible to accurately estimate real and reactive power by also storing a corresponding set of DFT coefficients of the voltage. That is, if the 1st, 3rd, 5th, and 7th DFT coefficients of current are stored, then these same DFT coefficients of voltage should also be stored. For a discrete-time system, real power is given by

$$P = \frac{1}{N} \sum_{n=0}^{N-1} i[n]v[n],$$

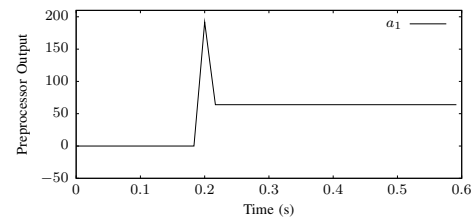
where $i[n]$ and $v[n]$ are the samples of current and voltage, respectively, over one period of length N . Let I_k and V_k denote



(a) Line Voltage.



(b) Aggregate Current.



(c) Preprocessor Output.

Fig. 2. Sample preprocessor output. The first two plots depict the line voltage and aggregate current while a simulated device is being turned on. The third plot shows the corresponding preprocessor output.

the (complex) amplitudes of the k^{th} harmonics of current and voltage, respectively. Using the Plancherel Theorem, real power can then be expressed in terms of the DFT of current and voltage,

$$P = \frac{1}{N^2} \sum_{k=0}^{N-1} I_k V_k^*.$$

Reactive power can be calculated in an analogous fashion. Thus, if only a small number of DFT coefficients of current are not approximately zero, then an accurate approximation of real and reactive power could be obtained by storing only the few significant DFT coefficients of current, and the same set of DFT coefficients of voltage.

Figure 2 shows the line voltage, aggregate current, and preprocessor output during the turn-on of a device that draws exclusively real power. The only non-zero preprocessor envelope is the envelope a_1 that corresponds to real power. It is important to note that, because the only non-zero DFT coefficient of current is the 1st coefficient (fundamental), this envelope is a scaled version of real power, regardless of the harmonic content of the voltage waveform. The only harmonics of the voltage waveform that affect real and reactive power are those harmonics that are also present in current.

III. FPGA-BASED SPECTRAL ENVELOPE PREPROCESSOR

To calculate, store, and communicate a relevant subset of DFT coefficients for power monitoring and energy scorekeep-

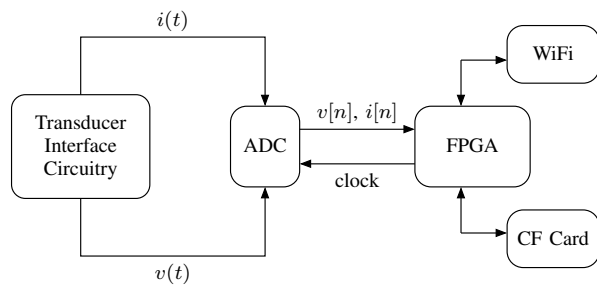


Fig. 3. Spectral envelope preprocessor block diagram. An ADC samples the line voltage and the aggregate current, which are then used by the FPGA to produce spectral envelope coefficients. These coefficients can be stored in a Compact Flash card and also transmitted via WiFi on demand.

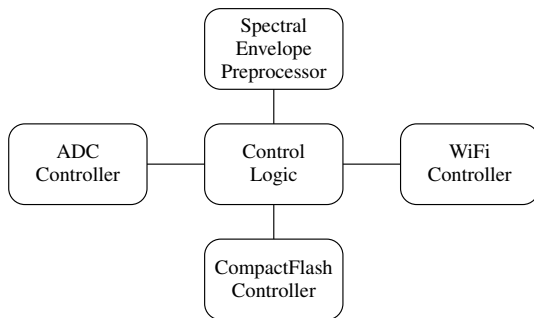


Fig. 4. FPGA block diagram. The preprocessor calculates spectral envelope coefficients. The ADC controller controls the sampling scheme of the ADC. The CF controller interfaces with a Compact Flash card to enable spectral envelopes to be stored and later recalled. The WiFi controller interfaces with an IEEE 802.11 WiFi transceiver to transmit spectral envelope data.

ing, a prototype FPGA (Field Programmable Gate Array) was constructed to implement a spectral envelope preprocessor. This system makes use of a low-cost FPGA (Altera Cyclone I, EP1C3T100C8). The spectral preprocessor consists of four subsystems: the first, to obtain current and voltage samples; second, to compute spectral envelope coefficients; third, to store computed spectral envelope coefficients; and fourth, to transmit the spectral envelope coefficients to another computation or display platform for further analysis. Each of these subsystems will be considered in detail. Figure 3 shows the overall block diagram of the system.

Data flows through the system as follows. The transducer interface circuitry first measures the line voltage and aggregate current, producing the signals $v(t)$ and $i(t)$. These signals are sampled and quantized by an analog-to-digital converter (ADC) that produces the samples $v[n]$ and $i[n]$. The FPGA processes these samples to compute spectral envelopes. The spectral envelope coefficients can be stored in a Compact Flash card for later use. The system also includes an IEEE 802.11b/g WiFi transceiver that allows any collection of the spectral envelope coefficients to be transmitted to another computation device for analysis. The FPGA provides control logic for each of the subsystems. Figure 4 shows a block diagram of the system implemented in the FPGA.

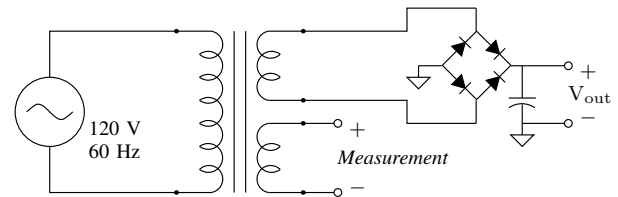


Fig. 5. Power supply and measurement schematic. The FPGA system receives both line voltage measurement and low-voltage supply through a transformer with dual secondary coils.

A. Current and Voltage Measurement

Current and voltage measurements from at least one voltage channel and at least one current channel are used to compute spectral envelopes. The system is easily expanded to measure more channels, supporting three-phase electrical services, for example. The prototype system uses an LEM LA 55 current transducer [18] to measure aggregate current and a transformer to measure the line voltage. A transformer with dual secondary coils is used in the prototype. This provides one coil for measurement, and a second coil for powering the preprocessor. The two coil arrangement provides a voltage sense with very little phase distortion, ensuring accurate calculation of in-phase and quadrature spectral components. Figure 5 illustrates this utility connection.

B. ADC Controller

In many signal processing applications, a computationally efficient algorithm like the Fast Fourier Transform (FFT) computes the complete spectral analysis of a sampled waveform. However, in situations like power monitoring, where a relatively small number of spectral coefficients may contain all or most needed information, needed spectral coefficients can be computed more efficiently by a traditional DFT implementation, i.e., by mixing observed waveform samples directly with the stored samples of basis sinusoids. In this approach, basis sinusoids are stored in a memory and multiplied by observed samples of a waveform. If there are N samples stored in memory for each basis sinusoid, then it is necessary to acquire N samples of the current and voltage waveforms for each line voltage period.

The FPGA coordinates the operation of the ADC to obtain the samples $i[n]$ and $v[n]$ of the current and voltage waveforms $i(t)$ and $v(t)$. To provide a known number of waveform samples per line period, the FPGA “locks” to the line voltage waveform. That is, the FPGA varies the sample rate to track with variations in the line voltage frequency. The sampling clock is derived from the output of a digital phase-locked loop (PLL) on the FPGA that tracks the line voltage frequency.

The phase of the sampling is set such that the first voltage and first current samples are taken at the negative to positive zero crossing of the line voltage. The goal is to multiply each entry in the basis sinusoid by the value of the waveform to be analyzed at the corresponding point in time. It is essential that the entire process is line locked to the line frequency in order for the estimated spectral envelope coefficients of current to correspond to “in-phase” and “quadrature” components

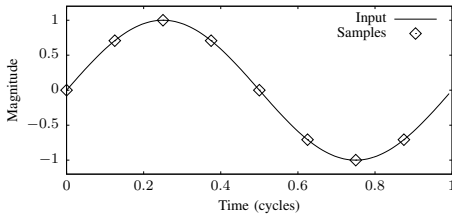


Fig. 6. Sampling scheme for the line voltage and current. Samples are taken at points in time that correspond to the samples stored in the basis sinusoid memory. For clarity, this figure only depicts 8 sample points per cycle, while the prototype system actually uses 128 sample points per cycle.

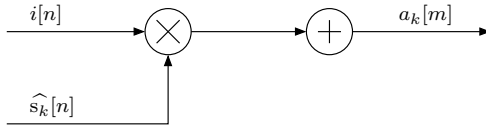


Fig. 7. Signal flow graph for the calculation of a spectral envelope. Raw current values $i[n]$ are multiplied by the appropriate elements of each basis sinusoid \hat{s}_k , and the results are accumulated over one period to produce each spectral envelope value.

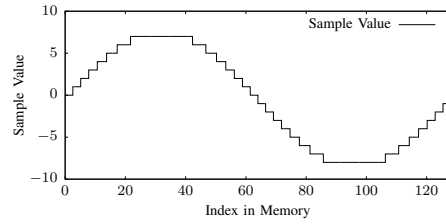
of current with respect to the fundamental of the voltage waveform. This sampling scheme is illustrated in Figure 6. The sample times and values are indicated by diamonds.

C. Envelope Preprocessor

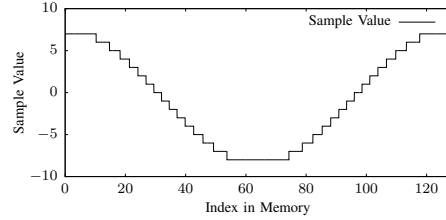
References [8]–[12] and their associated references describe various metering schemes that compute real, reactive, and apparent power, and also harmonic distortion in one form or another. References [1]–[4] describe specific implementations. In [1], multiple phase-locked loops, analog multipliers and integrators were used to estimate spectral envelope coefficients. In [2], a design using multiplying digital-to-analog converters, low-pass filters, and a single phase-locked loop was presented. In [3], a expensive digital signal processing board was used to perform the calculations. In [4], the processing power of a personal computer was used for spectral envelope coefficient estimation.

All of these systems can provide accurate estimates of spectral envelope coefficients or related quantities. They serve as essential building blocks of various types of metering systems. They are often expensive and dedicated. The FPGA-based system discussed in this section is an inexpensive single-chip solution that can estimate spectral envelope coefficients for stand-alone use or as part of a turn-key building block in more complex systems. The FPGA computes spectral envelopes using integer arithmetic on stored basis waveforms and observed waveform samples.

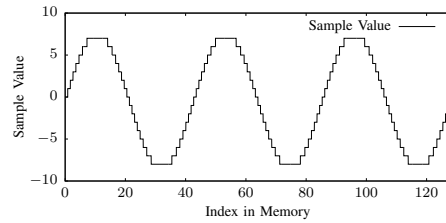
The FPGA-based spectral envelope preprocessor calculates the spectral envelopes of current, $a_1[m], b_1[m], a_3[m], \dots$, where m indexes the periods of the line voltage. Figure 7 shows the computation performed to produce estimates of a single spectral envelope coefficient, $a_k[m]$. The system multiplies $i[n]$, the samples of current, with samples of a basis sinusoid, and sums the result over a single period of the line voltage. Specifically, if N denotes the number of current



(a) Basis sinusoid \hat{s}_1 used to compute a_1



(b) Basis sinusoid \hat{c}_1 used to compute b_1



(c) Basis sinusoid \hat{s}_3 used to compute a_3

Fig. 8. Basis sinusoids. This figure depicts three examples of basis sinusoids, used to calculate real spectral envelopes of in-phase fundamental frequency content (a_1), quadrature fundamental frequency content (b_1), and in-phase third harmonic content (a_3). The basis sinusoids shown here are sampled at 4 bits for illustration—the actual prototype spectral envelope preprocessor uses 10 bit samples.

samples $i[n]$ over one line period, then

$$a_k[m] = \sum_{j=0}^{N-1} i[mN + j] \cdot \hat{s}_k[mN + j] \quad (4)$$

$$b_k[m] = \sum_{j=0}^{N-1} i[mN + j] \cdot \hat{c}_k[mN + j]. \quad (5)$$

Here \hat{s}_k and \hat{c}_k represent the stored integer basis sinusoids, which are pre-calculated to b bits of precision as

$$\hat{s}_k[n] = \text{floor} \left(\sin \left(\frac{2\pi kn}{N} \right) \cdot 2^{b-1} \right) \quad (6)$$

$$\hat{c}_k[n] = \text{floor} \left(\cos \left(\frac{2\pi kn}{N} \right) \cdot 2^{b-1} \right) \quad (7)$$

Each spectral envelope coefficient has a different basis sinusoid associated with it; for example, calculation of $a_3[m]$ involves multiplying $i[n]$ by \hat{s}_3 , which consists of discrete-time samples of a sinusoid at three times the line frequency, with its phase locked to the line voltage.

Figure 8 depicts examples of basis sinusoids. For illustration purposes, these sinusoids are sampled at only 4 bits, while the prototype system makes use of 10 bit samples.

Figure 9 shows a block diagram of the FPGA-based spectral envelope preprocessor. The preprocessor takes as input the

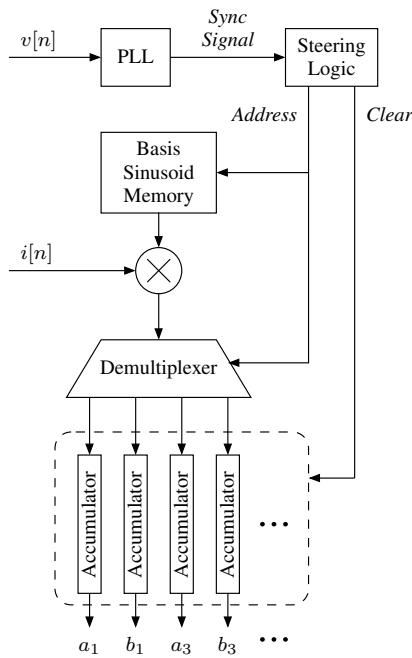


Fig. 9. Preprocessor block diagram.

discrete-time samples of $i[n]$ and $v[n]$ and produces estimates of the spectral envelope coefficients a_k and b_k of the current $i[n]$.

The voltage samples $v[n]$ are used as input to a phase-locked loop (PLL), which synchronizes the entire computation to the line voltage. As noted earlier, the computation process is synchronized to the line voltage so that the calculated spectral envelope coefficients correspond to meaningful physical quantities in steady-state operation (real power, reactive power, etc.). The output of the PLL is sent to a block of steering logic on the FPGA that produces the address for the basis sinusoid memory, as well as a clear signal for the accumulators. The basis sinusoid memory consists of the samples of the various basis sinusoids. The address produced by the steering logic specifies a single sample time of a single basis sinusoid. The sample of the basis sinusoid that is retrieved from the basis sinusoid memory is then multiplied by the current sample $i[n]$. The result of this multiplication is then passed through a demultiplexer which sends the result to the appropriate accumulator by using the address produced by steering logic to determine which spectral envelope coefficient is currently being calculated.

There is one accumulator for each estimated spectral envelope coefficient. The accumulators are all cleared at the end of each period of the line voltage through the use of the clear signal produced by the steering logic. For every sample $i[n]$, the address produced by the steering logic will select each of the basis sinusoids in turn, so that every sample of current is multiplied by the appropriate sample of each of the basis sinusoids.

This FPGA-based implementation provides a great deal of flexibility. For example, the subset of spectral envelope coefficients that are being estimated can be changed by altering

the entries in the memory to correspond to a different set of basis sinusoids. This implementation is also efficient in terms of FPGA resource utilization. It uses only a single PLL and a single multiplier, as opposed to previous hardware implementations that often used multiple PLLs and/or multipliers [1], [2]. This system can function with only a single multiplier because the FPGA is capable of multiplying each sample $i[n]$ by the corresponding sample of each of the basis sinusoids and forwarding each result to the appropriate accumulator, before the next sample $i[n + 1]$ arrives. The multiplier consumes substantial logic elements on the FPGA. It utilizes 24% of all resources used by the envelope preprocessor and 13% of the resources used by the complete system. By using only one multiplier, the design is capable of fitting in a small, low-cost FPGA.

There are several ways to configure and deploy the spectral envelope preprocessor for any given application. For situations where the voltage waveform is relatively sinusoidal and “stiff,” the spectral envelopes of current can be interpreted as scaled physical quantities in steady state. As discussed in §II-C, under these assumptions, the a_1 envelope of current in steady state corresponds to a scaled estimate of real power or “P”. The b_1 envelope of current corresponds to reactive power or “Q”. In situations where the voltage is not stiff and/or not sinusoidal, the FPGA could be tasked to also compute the spectral envelopes of voltage as well as current. This more complete set of spectral envelopes could be stored or transmitted to a computation platform or metering instrument that can quickly compute estimates of real or reactive power or other quantities of interest. Alternatively, the FPGA can be reconfigured to compute quantities like real and reactive power itself. In practice, we have found the basic computation of the spectral envelopes of current, assuming a stiff voltage source, to yield information that is directly useful for load monitoring and demand-side load control and diagnostics [17].

D. CF Controller

This FPGA subsystem stores spectral envelope data on an erasable memory like a Compact Flash (CF) storage card. The subsystem is capable of storing spectral envelope data as it is produced, as well as retrieving the spectral envelope data from any point in time, on demand. To interface with the CF card, the “True IDE” interface mode is used [19]. This interface mode is universally supported by Compact Flash storage cards and it allows the system to be easily adapted to interface with other mass storage devices that use the IDE interface standard, such as an IDE hard drive. While it would be possible to impose a filesystem on the CF card (i.e. FAT32), the current design treats the CF card as a single large, raw block of storage, for simplicity. Due to the low data rate of the spectral envelope coefficients (for the prototype preprocessor with 8 spectral envelopes, each stored at 24 bits of resolution, the data rate is 2.8 KB/s), even a moderately sized CF card could store the spectral envelope data for a substantial length of time. For example, in the prototype system, a 1 GB CF card would suffice to record data continuously for approximately 4.3 days.

E. WiFi Controller

This subsystem facilitates the transmission of spectral envelope coefficients to a PC or other computation platform for further analysis or display. It makes use of an IEEE 802.11b/g WiFi transceiver and TCP/IP. The current design is capable of supporting both ad-hoc and access point (infrastructure) networks.

The transmitted data is retrieved from the CF card as needed. The WiFi subsystem can operate in two different modes. In the first mode, the system streams spectral envelope coefficients as they are generated. In the prototype, this corresponds to a data rate of 2.8 KB/s. In the event of a momentary interruption in the connection to the PC, the system will automatically buffer data from the last successfully transmitted packet and resume transmission from that point when a connection is reestablished. The system will then send data at the highest available transmit speed (54 Mb/s for 802.11g), until the system catches up to the freshly produced spectral envelope data. In the second mode, an application on the PC requests data by specifying a range of time; the system then transmits all data from the desired range of time, at the maximum possible transmission speed.

IV. FLEXIBILITY

The design presented above is just one example of an FPGA-based load monitoring interface. The modularity of the design, and the versatility of FPGAs, makes it simple to change the transmission system, for example, to wired Ethernet (IEEE 802.3) or Bluetooth (IEEE 802.15.1) or Zig-Bee, or to change the storage system to, for example, a microSD card. An FPGA permits the interconnection of a wide variety of different subsystems to form a complex utility monitoring system. Even a small, low-cost FPGA is capable of implementing both the spectral envelope preprocessor as well as the required interface logic to control the various subsystems. Thus, an FPGA can serve as the backbone of an inexpensive, complete utility or load monitoring system.

V. PROTOTYPE RESULTS

The FPGA-based system discussed above calculates, stores, and transmits spectral envelope data. Figure 10 shows a picture of the prototype hardware.

To make use of this data, a monitoring or control system typically includes a subsystem to receive and use the spectral envelope data. For example, a homeowner could use a personal computer to collect spectral envelope data from the FPGA preprocessor installed in or near a circuit breaker panel. The prototype includes a PC-based software application that can interface with the FPGA-based preprocessor via wired or wireless communication channels, retrieve spectral envelopes, and display spectral envelope data. Using this spectral envelope data, the PC-side application can disaggregate the operating schedule of individual loads from measurements made on an aggregate power feed serving multiple loads. The application is self-training and identifies loads in essentially real-time. Screenshots of this program are shown in Figure 11 and Figure 12.

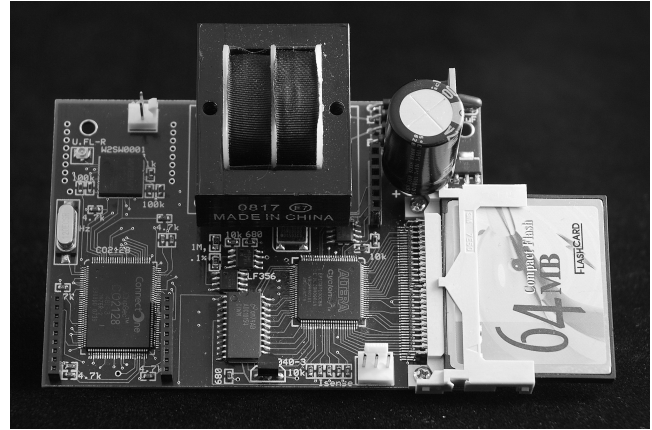


Fig. 10. Picture of the prototype FPGA-based system.

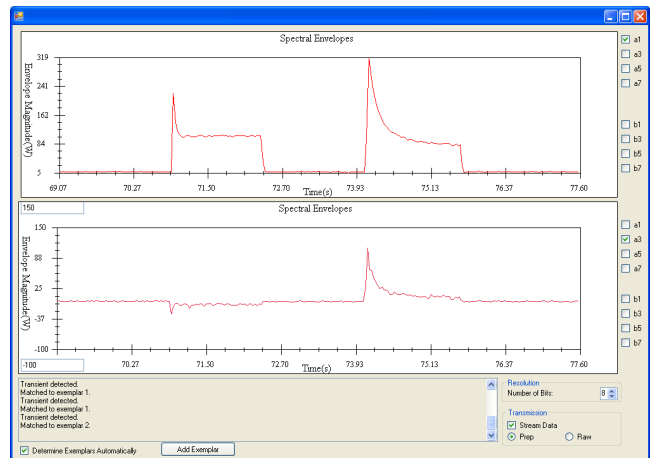


Fig. 11. Screenshot of the prototype software. The upper plot displays real power and the lower plot displays third harmonic content. This spectral envelope data corresponds to the light bulb and motor whose raw current values were shown in Figure 1. The lower left section of the screenshot contains the output of the classifier, which has correctly identified both of the loads.

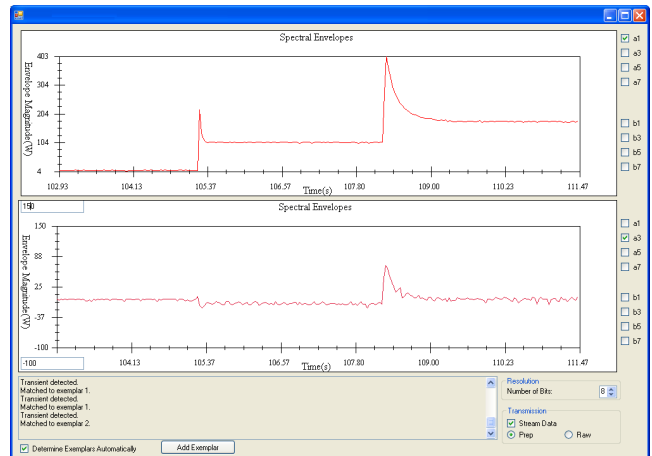


Fig. 12. Screenshot of the prototype software. The data shows the same light bulb and motor as Figure 11, with the difference that the motor starts while the light bulb is still turned on. As shown in the lower left corner of the screen, the system still classifies both devices correctly.

This software communicates via TCP/IP with the FPGA-based preprocessor. The software can retrieve any subset of recorded data, as well as issue commands, such as changing the sampling resolution of the ADC. Once spectral envelope data is retrieved, this software makes use of the Expectation-Maximization (EM) algorithm [6] to classify and recognize the operation of individual loads.

This software is only one example of the many possible ways to use spectral envelope data. For example, a system that uses this data to control a set of generators in a micro-grid is currently in development. Data could be retrieved by a web application that displays a live stream of data on a webpage. Other embedded systems could communicate with the FPGA-based system and use the retrieved spectral envelope data to control electrical loads.

VI. APPLICATIONS

The FPGA preprocessor can provide a turn-key component for creating utility monitoring, energy score-keeping, and diagnostic applications for a variety of systems. The preprocessor is relatively simple compared to microprocessor or DSP-based data acquisition systems. The concepts and hardware illustrated in this paper could be incorporated into individual loads, circuit breakers, or circuit breaker panels to provide energy consumption information for both monitoring and control.

The simplification in data storage and transmission bandwidth requirements afforded by the FPGA can also be extended to other domains. For example, it is possible to extend the non-intrusive monitoring concept beyond the realm of electrical distribution. We have used a single acoustic sensor to monitor the flow of water to a collection of hydraulic loads. This acoustic data is not “line locked” to any particular “utility frequency.” However, fingerprint acoustic signatures can still be developed by examining spectral content at frequencies dictated by resonant properties of the pipe. These fingerprints permit recognition of hydraulic loads or events in the water distribution system, as shown in Figure 13.

We are exploring extending the nonintrusive monitoring concept to gas mains, water mains, and to various occupancy detection schemes using air flow measurements and acoustic signals. We are also exploring nonintrusive electrical load detection by looking at both quasi-static and radiated signals associated with the operation of different loads.

Acknowledgements

This research was funded by the Grainger Foundation, the BP-MIT Alliance, the Office of Naval Research under the ESRDC program, the MIT Sea Grant College Program, and by the MIT Center for Materials Science and Engineering. The authors also gratefully acknowledge the advice and support of Dr. Manny Landsman.

REFERENCES

[1] S. B. Leeb, “A conjoint pattern recognition approach to nonintrusive load monitoring,” Ph. D. dissertation, Dept. Elect. Eng. Comput. Sci., Mass. Inst. Technol., Cambridge, MA, Feb. 1993.

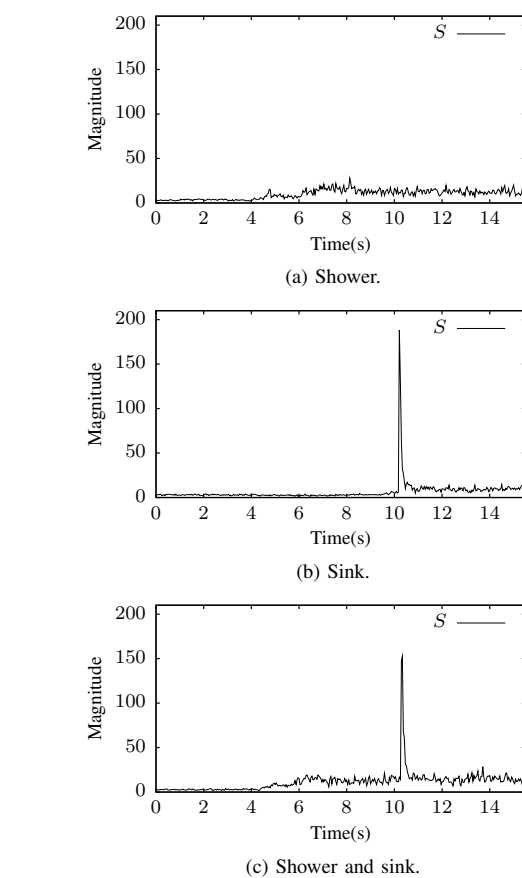


Fig. 13. Sample data showing the application of the FPGA-based preprocessor to acoustic signals from hydraulic loads. The plots show spectral envelope content S at 192 Hz for a shower, sink, and a combination of both.

[2] S. B. Leeb, S. R. Shaw, and J. L. Kirtley, “Transient event detection in spectral envelope estimates for nonintrusive load monitoring,” *IEEE Trans. Power Del.*, vol. 10, no. 3, pp. 1200-1210, Jul. 1995.

[3] S. R. Shaw, C. B. Abler, R. F. Lepard, D. Luo, S. B. Leeb, and L. K. Norford, “Instrumentation for high performance nonintrusive electrical load monitoring,” *Trans. ASME J. Sol. Energy Eng.*, vol. 120, no. 3, pp. 224-229, Aug. 1998.

[4] S. R. Shaw, “System identification techniques and modeling for non-intrusive load diagnostics,” Ph. D. dissertation, Mass. Inst., Technol., Cambridge, MA, Feb. 2000.

[5] J. Paris, et al. “Scalability of Non-Intrusive Load Monitoring for Shipboard Applications,” ASNE Day 2009, National Harbor, Maryland, April, 2009.

[6] A. P. Dempster, N. M. Laird, D. B. Rubin, “Maximum Likelihood from Incomplete Data via the EM Algorithm,” *Journal of the Royal Statistical Society, Series B* 39:1-38, 1977.

[7] “The Smart Grid: An Introduction,” U. S. Department of Energy, <http://www.oe.energy.gov/1165.htm>.

[8] U. S. Patent 5,212,441, “Harmonic-Adjusted Power Factor Meter,” May 18, 1993.

[9] U. S. Patent 5,229,713, “Method for Determining Electrical Energy Consumption,” July 20, 1993.

[10] U. S. Patent 5,548,527, “Programmable Electrical Energy Meter Utilizing a Non-Volatile Memory,” August 20, 1996.

[11] U. S. Patent 6,615,147, “Revenue Meter with Power Quality Features,” September 2, 2003.

[12] U. S. Patent 7,525,423, “Automated Meter Reading Communication System and Method,” April 28, 2009.

[13] <http://www.p3international.com/products/special/P4400/P4400-CE.html>.

[14] Shaw, S. R., S. B. Leeb, L. K. Norford, R. W. Cox, “Nonintrusive Load Monitoring and Diagnostics in Power Systems,” *IEEE Transactions on Instrumentation and Measurement*, Volume 57, No. 7, July 2008, pp. 1445-1454.

[15] Khan, U. A., S. B. Leeb, M. C. Lee, “A Multiprocessor for Transient Event Detection,” *IEEE Transactions on Power Delivery*, Volume 12, Number 1, pp. 51-60, January 1997.

[16] W. M. Siebert *Circuits, Signals and Systems*, The MIT Press, 1986.

[17] Laughman, C. R., “Fault Detection Methods for Vapor-Compression Air Conditioners Using Electrical Measurements,” MIT BT Ph. D. thesis, September 2008.

[18] <http://www.lem.com/docs/products/la%2055-p%20e.pdf>

[19] “CF+ & CF Specification Rev. 4.1”, <http://compactflash.org>