

Research Article

Accuracy-Energy Configurable Sensor Processor and IoT Device for Long-Term Activity Monitoring in Rare-Event Sensing Applications

Daejin Park and Jeonghun Cho

School of Electronics Engineering, Kyungpook National University, Daegu 702-701, Republic of Korea

Correspondence should be addressed to Jeonghun Cho; jcho@ee.knu.ac.kr

Received 10 September 2014; Accepted 25 October 2014; Published 14 December 2014

Academic Editor: Hai Jiang

Copyright © 2014 D. Park and J. Cho. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A specially designed sensor processor used as a main processor in IoT (internet-of-thing) device for the rare-event sensing applications is proposed. The IoT device including the proposed sensor processor performs the event-driven sensor data processing based on an accuracy-energy configurable event-quantization in architectural level. The received sensor signal is converted into a sequence of atomic events, which is extracted by the signal-to-atomic-event generator (AEG). Using an event signal processing unit (EPU) as an accelerator, the extracted atomic events are analyzed to build the final event. Instead of the sampled raw data transmission via internet, the proposed method delays the communication with a host system until a semantic pattern of the signal is identified as a final event. The proposed processor is implemented on a single chip, which is tightly coupled in bus connection level with a microcontroller using a 0.18 μm CMOS embedded-flash process. For experimental results, we evaluated the proposed sensor processor by using an IR- (infrared radio-) based signal reflection and sensor signal acquisition system. We successfully demonstrated that the expected power consumption is in the range of 20% to 50% compared to the result of the basement in case of allowing 10% accuracy error.

1. Introduction

In recent years, rare-event sensing systems [1] have been used in IoT-driven applications [2]. These systems feature internet connectivity, low-cost, very slow event-to-event duration, and long-lasting requirements. The IoT devices are used for activity monitoring [3], human sense interface [4], security monitoring, and medical applications [5]. The requirement for an extremely long lifetime is a critical issue in battery-operated IoT devices with a wireless connectivity.

The general-purpose microcontroller (MCU) has been widely used as a main processor of IoT-driven systems for sensor signal acquisition and data processing, which are not especially suited for rare-event sensing applications. The inefficiency of conventional MCUs in the sensing applications has been introduced in various literature [1, 6, 7] that reveal the key requirements of sensing application-specific architecture and processing methods. The latest studies of processor design for sensing applications are primarily based

on architectural approaches that consider an event-driven nature [8] in extracting informative features from raw sensory data by observing over long periods of time.

Figure 1 illustrates basic operation and power consumption in a main sensor processor of the IoT device. A timely sampling operation to measure the transition of the target environment begins with activation of the sensor. An analog-to-digital converter (ADC) and level comparators perform analog data conversion, which generates digitized sample data and an interrupt request to execute the user-defined subroutines for digital signal processing. The CPU in the MCU is woken up to execute user-programmed interrupt service routines (ISRs), which are software code to analyze the quantized sensor data. Finally, the gathered data is transferred via internet connection to host systems.

The general-purpose MCU-based IoT device consumes inefficiently operating power in an iterative manner. Discrete-time-based sampling and signal processing are executed iteratively during the entire period of sensor signal transition,

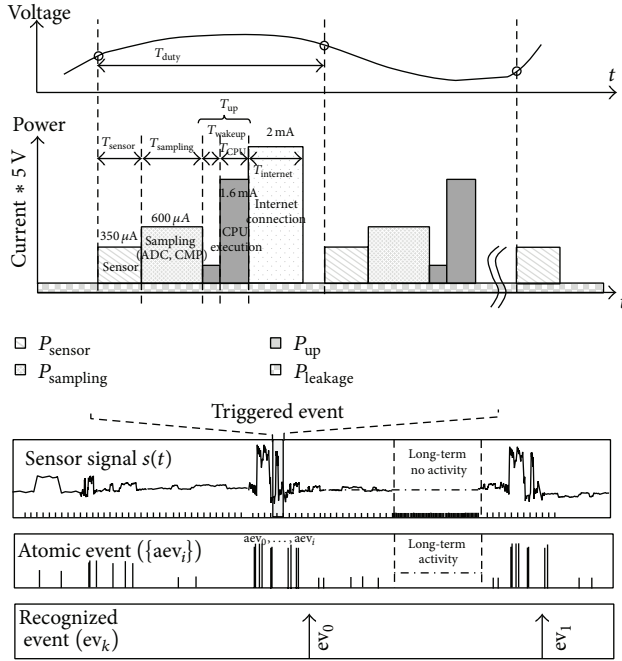


FIGURE 1: Power consumption for sensor signal sampling and processing in long-term activity monitoring.

even during the period of long-term sleep. Although this weakness in terms of the power consumption can be resolved using several programming techniques where the interrupt handlers consider the sensor signal behavior, this is not formal approach that covers the general cases of signal characteristics.

Several approaches [6, 9–11] regarding the behavior of sensing applications have been introduced in sensor-based application for low power operations, especially in activity sensor, sensor processor, and wireless sensor nodes. However, the use of the discrete-time-based operation is restricted because the most feasible devices rely on commercial off the shelf (COTS) MCUs attached with several sensors. These devices attempt to control the ratio of the sleep mode to the active mode by switching the operation mode to lower the power consumption according to being with or without the sensory data, which are continuously determined by the discrete-time-based monitoring of the signal.

2. Motivation

To address this limitation of the conventional digital system architecture by using the discrete-time-based sensor data processing method, we propose an event-driven system architecture that modifies traditional digital system design. We present a theoretical framework to implement an event-driven sensor processor for general rare-event sensing applications by analyzing the system operations.

2.1. Event-Space Signal Representation. Our main research begins with an event-space representation of the signal, instead of the digital data space domain. Figure 2(a)

illustrates fundamental signal elements using attributes of interest, and Figure 2(b) shows the event data space transition-based representation, which describes in detail the relationship between the featured points of the sensed signal.

The extracted features of the sensed signal are encoded into the elapsed time between events and informative value such as voltage level and edge phase crossing the trigger point of the signal. The fundamental event defined, which is defined as an atomic event with the most important information, provides a signal representation on an abstract level and reduces the computational complexity in performing basic data processing for extracted informative features of interest. The collected atomic events include partial information in the original signal that specifies whether the desired featured points of the signal are present.

2.2. Event-Quantization with Accuracy Error. The event-quantization concept extends the time-quantization method for signal representation that uses elapsed time to enhance the conventional data sampling and processing method. Time-quantization monitors only the specific conditions of the signal transition and captures the time-stamps. The event-quantization method also determines whether the specified characteristics of the signal exist. Figures 2(c) and 2(d) illustrate the difference between the time-sampling and event-quantization methods with an accuracy error, which monitors only the presence of the featured points of interest.

In terms of accuracy, there are two types of operations: timing accuracy and data resolution accuracy [12]. The former is dependent on the sampling frequency compared to the received signal bandwidth, and the latter is derived from the representative resolution of the sampled data. The timing resolution is dependent on the clock-duty resolution to resolve the timing window of the processing operations accurately.

Higher timing resolution for the duration measurement requires the accurate data processing in the time domain. However, in most cases, the data value resolution of the target system is tightly required, but time—related specification including the response time is relatively allowable in a certain range of error. In particular, in rare-event sensing applications, we assume that the timing measurement accuracy for the elapsed time between the arrived signal events can be performed with inaccurate operating clock relatively.

Although the events contain inaccurate time information, the final event receiver (for example, a human interface) is unable to identify the error differences compared to the ideal data within a certain error range [13].

2.3. Accuracy Configurable Signal-to-Event Conversion. The event-based approach, with a certain amount of accuracy error, is described by the proposed event-driven sensor data processing flow, as shown in Figure 2(e). The input signal is monitored with specified interest-of-signal characteristics $W(k)$ to generate the specific atomic events aev_i of the signal. The set of atomic events during the specified region of the signal are traced into the tracer memory as an event vector \vec{AEV} , which contains the sequence of the atomic events and

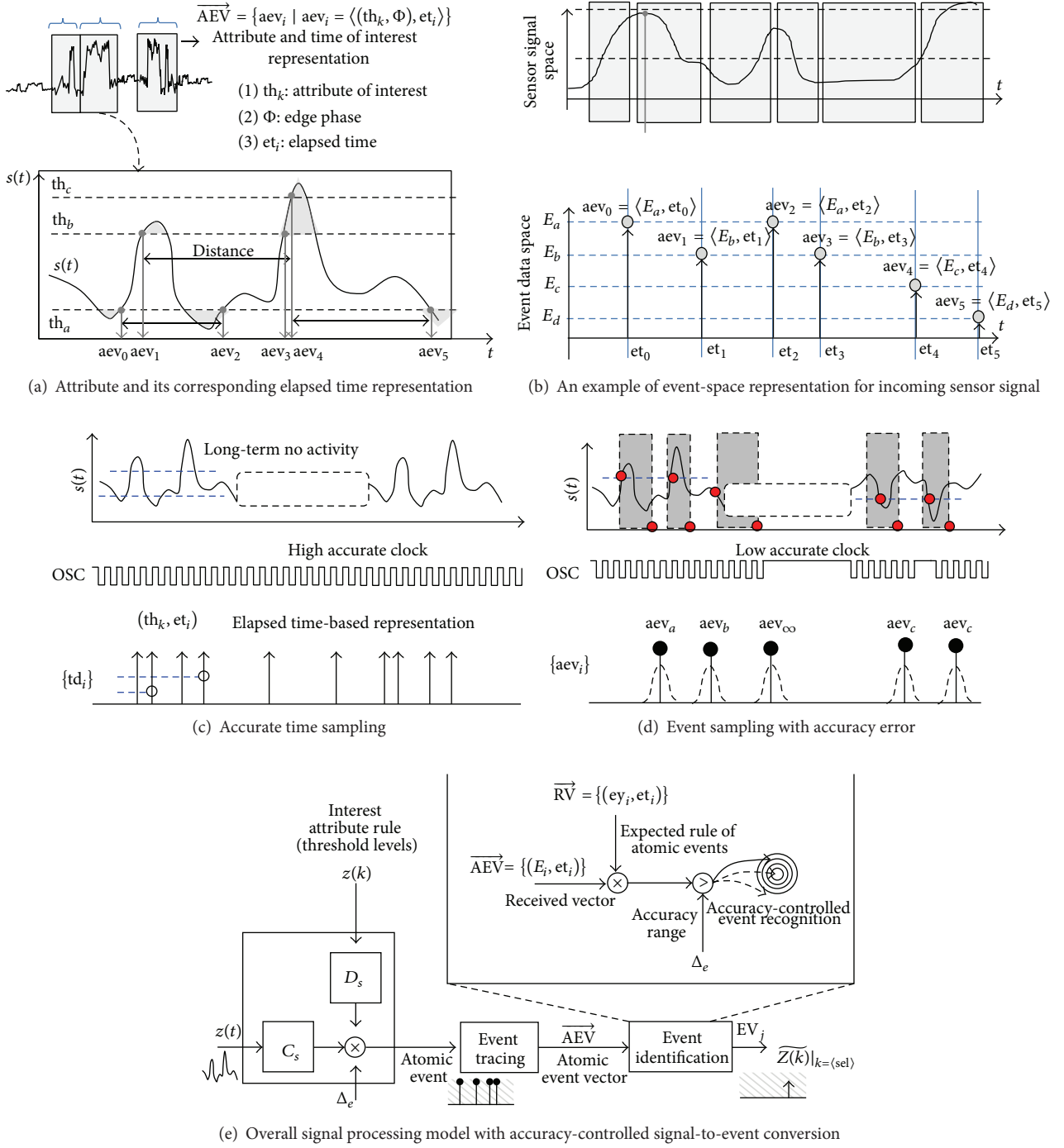


FIGURE 2: Event-space quantization and accuracy-configurable event-driven sensor processing flow.

the time distance relationship between the atomic events. The traced event vector identifies the approximate result $\widehat{Z}(k)$ as a final event by comparing it to the expected rules of the atomic events.

These approximation approaches enable us to reduce the computational complexity in order to manipulate a large amount of collected sensing data. As a result, power consumption will be reduced. For applications related to human

interaction, an approximation approach enables developers to design the computational block using smaller hardware resources, while providing sufficient performance in limited resolution of the accuracy.

In accuracy-controlling approaches defined from the specifications, our study focused on the data-representation resolution, the timing resolution of the sampling frequency, and the response time as a delay time [14]. This enables

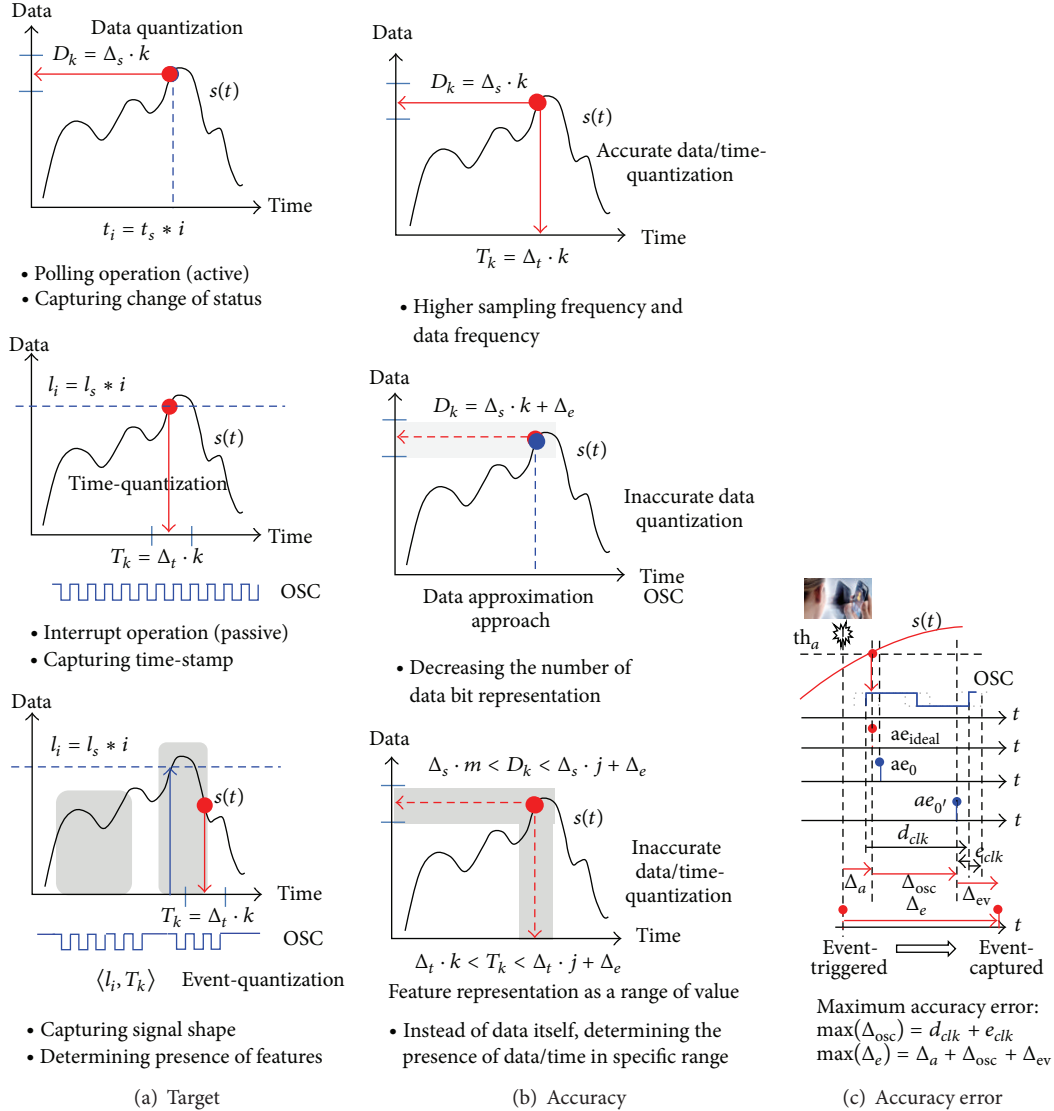


FIGURE 3: Category according to sampled target and its accuracy.

the configuration of the operation accuracy in the processor architecture level according to the abstraction level of the proposed event-quantization approach.

3. Related Work and Constraints

To overcome the weakness of the frequent CPU wake-up in the discrete-time sampling, continuous time signal processing techniques [15, 16] have been proposed. If a certain condition of the signal status, such as the voltage level at a specific time, is matched with the user-defined condition [17, 18], the time value at the triggered condition is sampled and quantized [19] by the selective method, which also helps to reduce operational power [20].

The continuous time sampling method illustrated in the second graph of Figure 3(a) requires additional hardware

resources, including a dedicated oscillator and high-accurate timer block, to measure the elapsed time value instead of data sampling in high resolution. Event-driven signal processing based on the time-quantization method requires hardware overhead and more computational time for the time-distance calculation, which gives rise to additional power consumption. The required power and hardware resource overhead, which are needed to compensate for reduced wake-up power consumption, must be considered in order to achieve benefits in total energy efficiency due to hardware-energy trade-off.

The hybrid method, which uses a level triggered system wake-up and continuous time sampling scheme, monitors important signal transitions and performs detailed analysis using the discrete-time sampling method, which involves digital signal processing for second detail signal analysis. The proposed event-driven sensor data processing method tries to capture the signal shape instead of the elapsed time-stamp

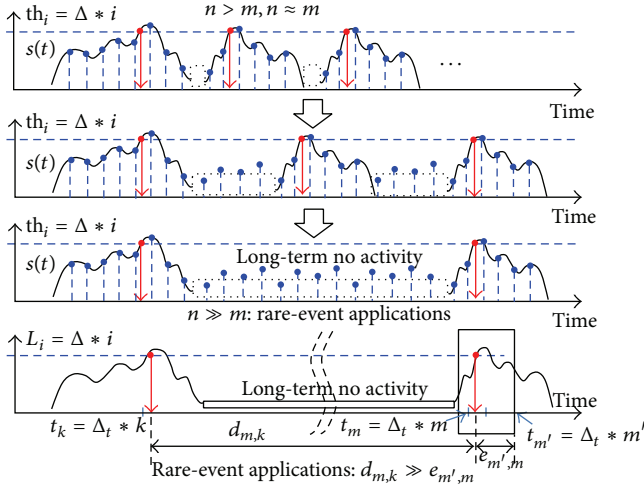


FIGURE 4: Wake-up frequency for data sampling and event-driven sampling.

at the triggered point, as illustrated in the third graph of Figure 3(a).

The trade-offs in terms of energy and accuracy have been studied widely [21, 22]. To obtain long lifetime operations under limited battery power [23], the latest research introduces inaccurate computation techniques [12, 24] with approximation-based hardware designs, as described in the second graph of Figure 3(b).

The proposed sensing processor for the rare-event sensing applications adopts the event-driven approach of the continuous time-based sampling method. Inaccurate time-data manipulation, as shown in the third graph of Figure 3(b), reduces computational complexity and sampling resolution by determining the presence of featured events in the specific range. The level that allows an accuracy error in the time-stamp measurement is depicted in Figure 3(c). The level can be adjusted by making the trade-off between the processing energy consumption and the operating specification.

Figure 4, shows the difference between the discrete-time samples and the featured events of interest, with the common shape of the rare-event sensor signal. Event sources, such as hand gesture, proximity, and object activity, generate signal pulses for which the distance between featured points of the signal is very long. The number of data samples (n) is greater than the number of events (m). In this work, we assumed an application-specific constraint of rare-event characteristics, which result in a small number of events compared to the number of data samples.

The event-quantization accuracy depending on the resolution of the elapsed time-stamp is described as $e_{m',m}$ in Figure 4. The rare-event sensing applications for which the event-to-event duration is relatively larger than the accuracy error have the following applications-specific constraints:

$$d_{m,k} \gg e_{m',m}. \quad (1)$$

With these application-specific constraints, the event identification accuracy error caused by the inaccurate time-stamp measurement clock is relatively insensitive derived

from (1). The recognized event observer, such as human eye, allows a certain amount of inaccuracy in identifying a meaning of the events, which are constructed by the proposed inaccurate event-driven sensor processor.

The proposed sensor processor is designed with these application-specific constraints by reducing the accuracy of the time-stamp measurement clock, decreasing the bit width of the timer block to capture the time-stamps, and decreasing the operational complexity of the time-to-time distance measurement blocks, which are specially implemented as a dedicated accelerator for event recognition in the implemented hardware.

4. Proposed Architecture

4.1. Data-Time Sampling with Accuracy Error. The first stage of the sensor processor is a sampler that gathers the time-variant information from the received signal. The conventional sampling method in Figure 5(a) attempts to collect all information in discrete-time from the target signal. There is no need to hold the time-stamp data. The uniformly sampled set in the time domain is described in the following definition.

Definition 1. Given continuous signal $s(t)$, let t_s be a fixed sampling time and $S_{\text{unitime}} = \{s_1, s_2, \dots, s_n\}$ a uniformly sampled set in time domain $T = \{t_1, t_2, \dots, t_n\}$. A sampled data $s_i \in S$ and its data quantization result with error Δ_d by data quantization function DQ are defined as follows:

$$d_i \pm \Delta_d = \text{DQ}(s_i = s(t_s * i)). \quad (2)$$

From this, the sampling time t_s in turn is defined as follows:

$$t_s = t_{i+1} - t_i. \quad (3)$$

The quality of the data sampling toward zero Δ_d is dependent on the accuracy of the function DQ, which is usually implemented with ADCs or a comparator, and the resolution of sampling time t_s .

The level-triggered interrupt-based sampling, as shown in Figure 5(b), tries to capture the time-stamp when it crosses a predefined condition with the parameterized value (such as the voltage level) and its transition edge phase.

Definition 2. Given continuous signal $s(t)$, let $L = \{L_1, L_2, \dots, L_n\}$ be a set of all levels of interest to monitor data from $s(t)$, TE a set of all pairs of the triggered level type and its elapsed time, $\text{TE} = \{\text{te}_i \mid \text{te}_i = \langle \text{type}, \text{et} \rangle, i = 1, 2, \dots, n\}$, and T_{clk} a fixed minimum period of the timer to measure the time-stamp at a triggered point. For the event $\text{te}_i \in \text{TE}$, sampled time t_k is elapsed time after the previous event te_{i-1} occurs, and its quantization result error Δ_t by the time-quantization function TQ is defined as follows:

$$t_k \pm \Delta_t = \text{TQ}(\text{te}_i \cdot \text{et}) = \text{te}_{i-1} \cdot \text{et} + T_{\text{clk}} * k, \quad (1 \leq k < \infty). \quad (4)$$

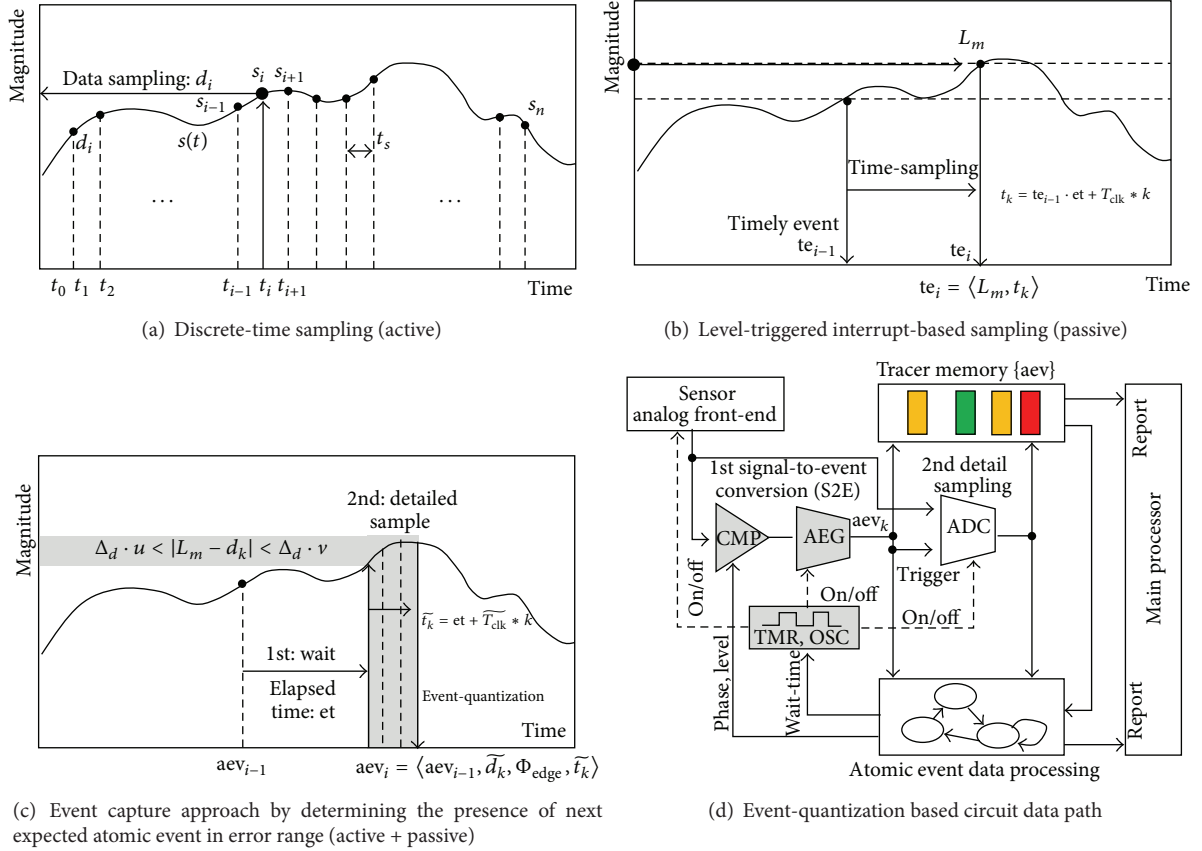


FIGURE 5: Event sampling illustrated and its circuit data path.

The elapsed time for the i th triggered event is recursively determined by searching the meet condition “ k ” of the following equation:

$$\begin{aligned} te_i \cdot et &= te_{i-1} \cdot et + t_k, \\ \forall DQ(L_m) &= DQ(s(t_k)), \quad L_m \in L. \end{aligned} \quad (5)$$

The time-stamp $te_i \cdot$ time resolution toward zero Δ_t is dependent on the minimum value of the time advance (T_{clk}) and the number-of-bits representation of (k) value to encode the time value. Higher resolution of the time value requires continuous operations of the oscillator and timer unit with higher accuracy and large size of a timer counter unit to measure the time-stamp, which leads to energy consumption overhead as a side effect.

Figure 5(c) describes our approach to capture the signal shape as an atomic event crossing a certain range of arrival time. To more formally define our approach, we begin our explanation by first presenting the following definitions.

Definition 3. Given continuous signal $s(t)$, let $AEV = \{aev_i \mid aev_i = (aev_{i-1}, \text{value}, \text{phase}, et)\}$ be a sequence of an atomic event aev_i crossing the specific level and time condition with a relationship of previous atomic event aev_{i-1} , where $aev_i \cdot$ value is a result of the approximation-based data quantization

function ADQ and $aev_i \cdot et$ is a result of the approximation-based time-quantization function ATQ, described as follows:

$$\begin{aligned} \widetilde{d}_k &= ADQ(s(t_k), L_m, \Delta_d, u, v), \\ \forall \Delta_d * u &< |L_m - d_k| < \Delta_d * v, \\ \widetilde{t}_k &= ATQ(aev_i \cdot et, \widetilde{T}_{clk}), \end{aligned} \quad (6)$$

where $\widetilde{T}_{clk} = T_{clk} + \Delta_t$.

The meet condition k , when the expected crossing is present, is described in the following equation:

$$\widetilde{t}_k = et + \widetilde{T}_{clk} * k, \quad \forall DQ(s(\widetilde{t}_k)) = \widetilde{d}_k. \quad (7)$$

The atomic event generator (AEG) builds an element with the attributes, which are encoded with the digitized signal level, elapsed time, and edge phase in the following equation:

$$AEG(s(t), L) = \{aev_i \mid aev_i = \langle aev_{i-1}, \widetilde{d}_k, \phi_{edge}, \widetilde{t}_k \rangle\}. \quad (8)$$

From (8), the extracted information, as an atomic event, is encoded with the approximation value of the signal level, the reduced time-quantization value of the elapsed time, and the relationship of the previous atomic event aev_{k-1} .

Figure 5(d) shows the proposed hardware data path for the atomic-event sampling in Figure 5(c), including the level

comparators, timer, oscillator, and atomic event generator (AEG) implementing operations from (2) to (8).

4.2. Atomic Event Segmentation. Atomic event generation is a method to represent a certain range of the continuous signal pattern with an abstract event. The signal representation is classified by a user-defined set of signal segments. We provide an example in Figure 6. The feature scan window in Figure 6(a), which is used to capture the atomic event of the signal, is configured with a specific voltage level, time window, and elapsed time at the feature point. The configured scan window determines if the featured points are monitored in the snapshot of the signal passing through the configured scan window and generates an element of a set of atomic events in Figure 6(b).

Definition 4. Given the configured feature scanning window to extract the atomic events from $s(t)$, let T_{start} be a start time monitoring the signal, let T_{end} be the end of monitoring the signal, let L_r be a rising signal level at which the time-stamp is T_r , let L_f be a falling signal level at which the time-stamp is T_f , let the pair of L_x and T_y be featured point, and let D_{max} be a maximum time value in which the featured points are present. The set of signal segments described by the configuration $\Omega = \{\Omega_i \mid \Omega_i = (T_{\text{start}}, T_{\text{end}}, L_r, L_f, T_r, T_f, D_{\text{max}})\}$ of the featured scanning window is defined as Ω and is used to extract the atomic events of interest for the AEG function, which is defined as follows:

$$\{\text{aev}_i\} = \text{AEG}(s(t), \Omega). \quad (9)$$

Ω_{up} defines a signal segment of the feature scanning window as an example showed in “Up-Pulse” shape of Figure 6(c). In our applications, $\{\Omega_{\text{type}} \mid \text{type} = \text{“up”}, \text{“su”}, \text{“sd”}, \text{“dp”}, \text{“isu”}, \text{“isd”}\}$ is used.

Figure 6(c) shows examples of the user-defined signal shape as an atomic event, which is determined by the configured feature scan window. The “up-pulse” pattern rises at recognized time R_r and falls at recognized time R_f for the L_{probe} level during maximum timer window D_{max} . The continuous signal shape during a configured time range of T_{start} and T_{end} is represented as abstract event R_{up} with the following equation:

$$\text{AEV}(s(t), \Omega_{\text{up}}) \simeq \text{aev}^{\text{up}} = \langle \Omega_{\text{up}}, L_{\text{probe}}, (R_r, R_f) \rangle. \quad (10)$$

The expected rule to identify aev^{up} atomic event pattern, allowing time error margin Δ , is represented by the following equation:

$$R^{\text{up}} = (\text{aev}^{\text{up}}, \Delta). \quad (11)$$

The “step-up” pattern rises at time R_r and does not fall within the timer window D_{max} . The continuous “step-up” pattern signal can be also represented by the abstract atomic event view in the following equation:

$$\text{AEV}(s(t), \Omega_{\text{su}}) \simeq \text{aev}^{\text{su}} = \langle \Omega_{\text{su}}, L_{\text{probe}}, (R_r, -) \rangle. \quad (12)$$

The ∞ -step-up pattern includes the specific range of no signal transition crossing the specified voltage level L_{probe} within maximum timer window D_{max} and a rise at any time. The continuous ∞ -step-up pattern signal can be also represented by the abstract atomic event in the following equation:

$$\begin{aligned} \text{AEV}(s(t), \Omega_{\text{isu}}) \\ \simeq \text{aev}^{\text{isu}} = \langle \Omega_{\text{isu}}, L_{\text{probe}}, (\infty, \infty, R_r) \rangle. \end{aligned} \quad (13)$$

The aev^{isu} atomic event pattern is also represented by the expected atomic event pattern rule and its time error margin using the following equation:

$$R^{\text{isu}} = (\text{aev}^{\text{isu}}, \Delta). \quad (14)$$

The ∞ -step-up pattern is a powerful method to simplify the representation of the long-term signal shape with no activity, which leads to a reduction in the capacity of the information.

Figure 6(d) shows the capability to represent various signal shape by the configuration of L_r , D_{max} , T_r , T_{start} , and T_{end} in the feature scan window. One signal shape can be divided into the several slices by user-defined signal segmentation. If the time window for signal segmentation is the same as the fixed width t_s in the discrete-time sample method, the result of the atomic event generation is equivalent to that of the discrete-timed sampling. The proposed atomic event generation approach enables a trade-off between the signal extraction accuracy and its processing power consumption.

4.3. Event-Driven Data Processing. The atomic event generator (AEG) scans the continuous signal $s(t)$ passing through the configured feature scan window to determine the presence of the signal shapes of interest, as shown in Figure 7(a). The set of atomic events is generated with a pair of attributes and time-stamps as a result of the time-quantization shown in Figure 7(b). Consider

$$\text{aev} = \{\text{aev}_i \mid \text{aev}_0, \text{aev}_1, \dots, \text{aev}_i = (“L_i”, t_{s_i})\}. \quad (15)$$

Figure 7(c) shows a signal representation by a set of atomic events with a certain amount of error. This is denoted in the following equation:

$$\widetilde{\text{aev}} = \{\widetilde{\text{aev}}_i \mid \widetilde{\text{aev}}_0, \widetilde{\text{aev}}_1, \dots, \widetilde{\text{aev}}_i = (“L_i”, \overline{t_{s_i} \pm \Delta})\}. \quad (16)$$

aev_i , which is matched with the configured scan window AE_i , is represented as an abstracted atomic event index in Figures 7(d) and 7(e), which indirectly address the detailed attributes in the constant dictionary. The continuous analog signal is converted into a set of event quantized data $\widetilde{\text{aev}}_i$, and its index value is traced only into the atomic event tracer buffer. Therefore, the traced event data processing manipulates the index value and its relationship to the representative atomic events to generate the final event EV. The proposed event-driven sensor data processing unit (EPU), which is based on event-quantization, provides the following advantages compared to conventional sensor data processing:

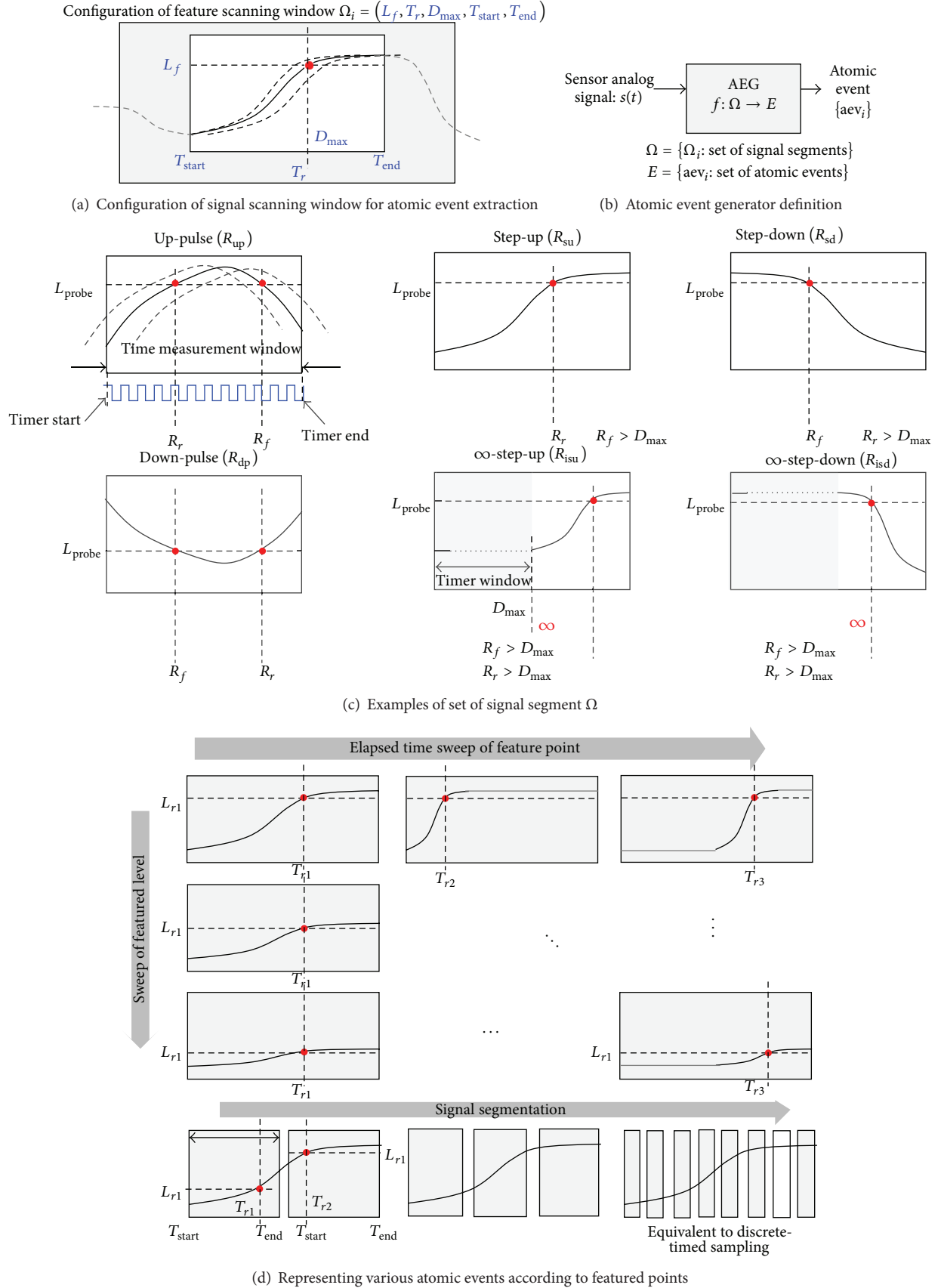


FIGURE 6: Configuration of signal feature scan window and atomic event segmentation.

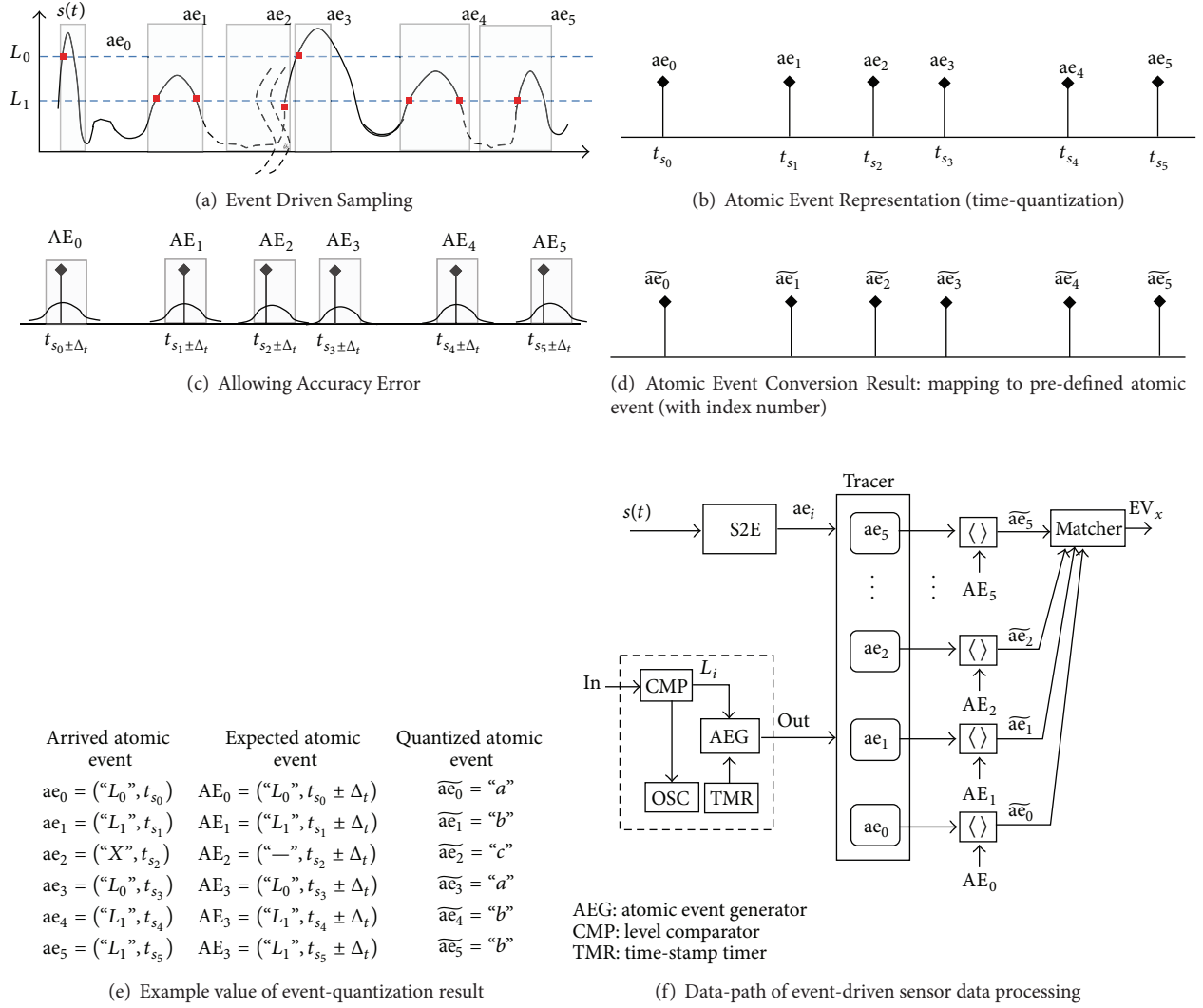


FIGURE 7: Event-driven sensor data processing concept for macro-level signal analysis.

- representing the continuous analog signal with a small number of atomic events for specially featured points,
- decreasing the number of pieces of processing data with reduced atomic events,
- allowing the accuracy error of the generated atomic events for application-specific properties of the rare-event sensing applications,
- decreasing the complexity of the expected atomic event comparison circuit, by comparing the time-stamp range, instead of the accurate value,
- mapping the recognized atomic events into the representative atomic event set with only index value,
- transforming the raw data processing into the index value.

Figure 7(f) illustrates the corresponding data path of the event-driven sensor data processing, including the atomic

event generator, tracer, feature scan window, and the pattern matcher (which is described as event-print window matcher in Figure 8(b)).

4.4. Final Event Identification. The archived atomic events in the tracer memory are evaluated as a similarity factor, which is calculated by the total sum of the distance between the collected events and the expected rule. We define this procedure as R^* -plain projection, as illustrated in Figure 8(a). The atomic event extraction procedure is described by the \overrightarrow{AEV} in the following equation. The operation \otimes describes the atomic event conversion for the continuous sensor signal $s(t)$ with the atomic event conversion rule, which is introduced in Figure 6(c). Consider

$$s(t) \otimes L_0 = \overrightarrow{AEV}. \quad (17)$$

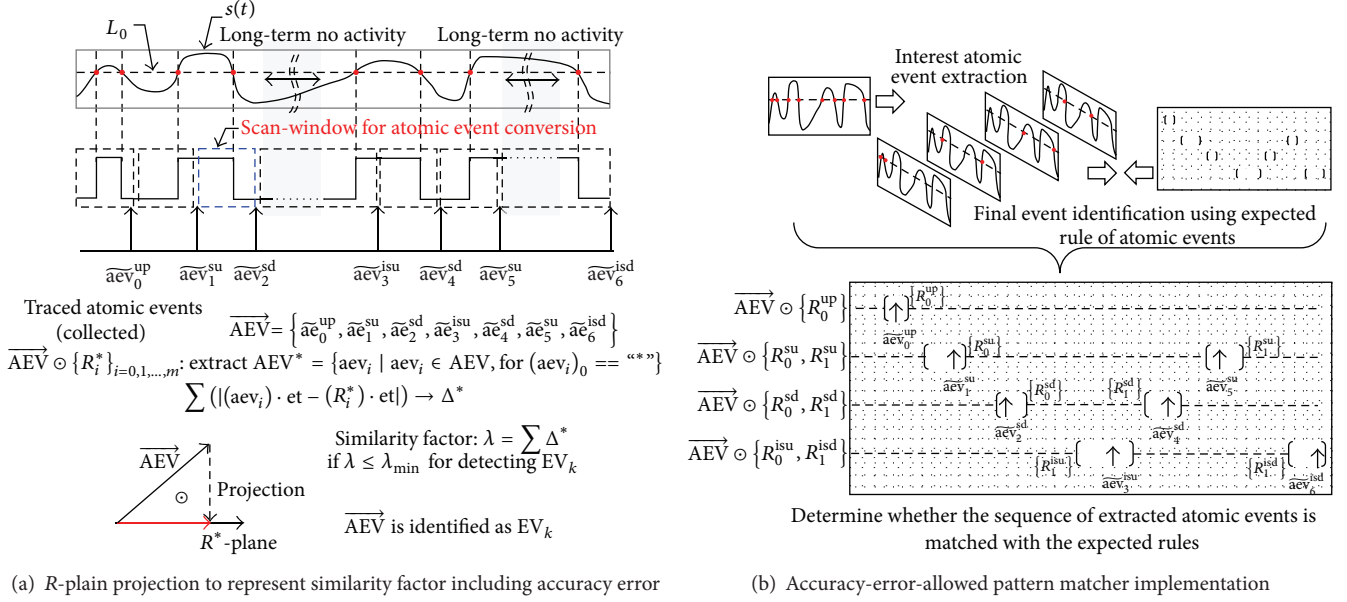


FIGURE 8: Event-print identification for atomic event of interest.

In the signal example of Figure 8(a), the result of atomic event generation is described by the following equation:

$$\vec{AEV} = \{\vec{ae}_0^{up}, \vec{ae}_1^{su}, \vec{ae}_2^{sd}, \vec{ae}_3^{isu}, \vec{ae}_4^{sd}, \vec{ae}_5^{su}, \vec{ae}_6^{isd}\}. \quad (18)$$

Figure 8(b) shows the proposed event-print window matcher implementing the R^* -plain projection to determine the final event using the result based on the similarity factor. The event-print matcher allows a certain error comparing the arrived atomic events to the expected rule, which is illustrated as blank holes in the punch card of Figure 8(b). Multiple R^* -plain projections are performed simultaneously to compare the archived atomic events with various pattern rules. We define these matching operations \odot with the following equation:

$$\vec{AEV} \odot \{R_i^*\}_{i=0,1,\dots,m}. \quad (19)$$

The extracted atomic event vector is described by the following equation:

$$\vec{AEV}^* = \{\vec{ae}_i^* \mid \vec{ae}_i \in \vec{AEV}, \text{for } (\vec{ae}_i) \cdot \text{type} == "**"\}. \quad (20)$$

The difference between the elapsed time stamp value and expected event arrival value is calculated for the extracted atomic events list using the following equation:

$$\Delta_i^* = |(\vec{ae}_i^*) \cdot \text{et} - (R_i^*) \cdot \text{et}|. \quad (21)$$

The operation \odot of the extraction rule R_i^* for atomic event vector \vec{AEV} can be considered as the R_i^* plane projection.

The total similarity factor, which is compared to the expected event rules, is described as the summation of the difference in the measured time-stamp:

$$\lambda = \sum \Delta_i^*. \quad (22)$$

If the final value λ is less than the minimum value of the expected error range, the received atomic event \vec{AEV} is identified as EV_k . The trade-off between the processing accuracy and corresponding energy consumption can be selected to satisfy the design specification, which is described by the functional constraints and the required operating lifetime.

5. Implementation and Experimental Results

The proposed event-driven sensor data processing unit (EPU) is implemented as an accelerator to perform energy-efficient event recognition from the incoming sensor signal, as described in Figure 9(a). The regular case for sensing data analysis can be covered by the proposed event processor, which enables the MCU core to hibernate during sleep mode. The user-defined software configured by the MCU core allocates the configuration of the predefined atomic event conversion conditions for the feature scan window.

The set of atomic events is redirected into the tracer buffer via the dedicated DMA bus. Atomic event generation and event vector construction are performed in silent background mode, without waking up any of the main MCUs.

The newly designed event-driven sensor processor, including the general-purpose MCU core and the EPU core, is implemented using 0.18 μm CMOS embedded-flash process technology and has a die size of 1.2 mm \times 1.2 mm, as shown in Figure 9(b). The proposed method requires, approximately, an additional 7500 logic gates using a 2-input

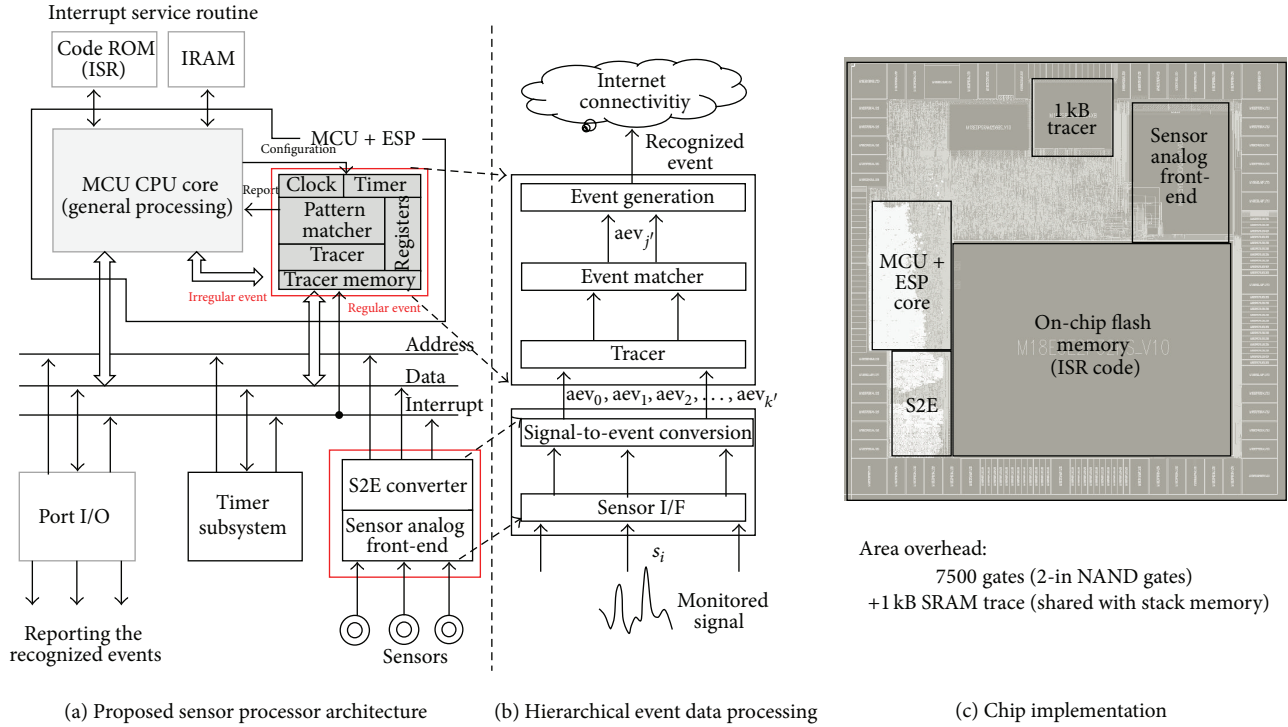


FIGURE 9: VLSI IC implementation of microcontrollers with the proposed event signal processing unit (EPU).

NAND for the timer counter, signal-to-event converter (S2E) including AEG blocks, event-print window matcher in EPU unit, and 1-KB SRAM buffer for the atomic event vector tracer.

Figure 10(a) shows the experimental design and measurement method to validate the efficiency in processing energy consumption by the proposed method and its implemented hardware. The first step of the evaluation is performed at the simulation level using the Matlab/Simulink models.

The physical sensor signal acquisition by the real activity (such as gesture swipe, proximity, and human presence) is performed off-line to save the raw dump file of time-variant sensor signal. Then the raw file of the archived sensor signal is loaded into the Matlab workspace. Figure 10(b) shows that the proposed event processing flow is evaluated by the model-based designs using discrete-event design tool sets supported by the Simulink.

The second step of the evaluation is performed by the circuit-level simulation for the synthesizable hardware design. The proposed method and its hardware architecture are implemented with the fully synthesizable Verilog RTLs, which can be physically mapped onto the FPGA or CMOS silicon chip.

The implemented chip includes the test mode interface, which requires about 1800 logic gates, to access the on-chip registers and the bus transactions in supervisor mode. If the predefined test sequences are forced into the input ports on the power-up duration, the chip is entered into the supervisor mode in which the important nodes of the system can be accessed by the external interface. The user-driven

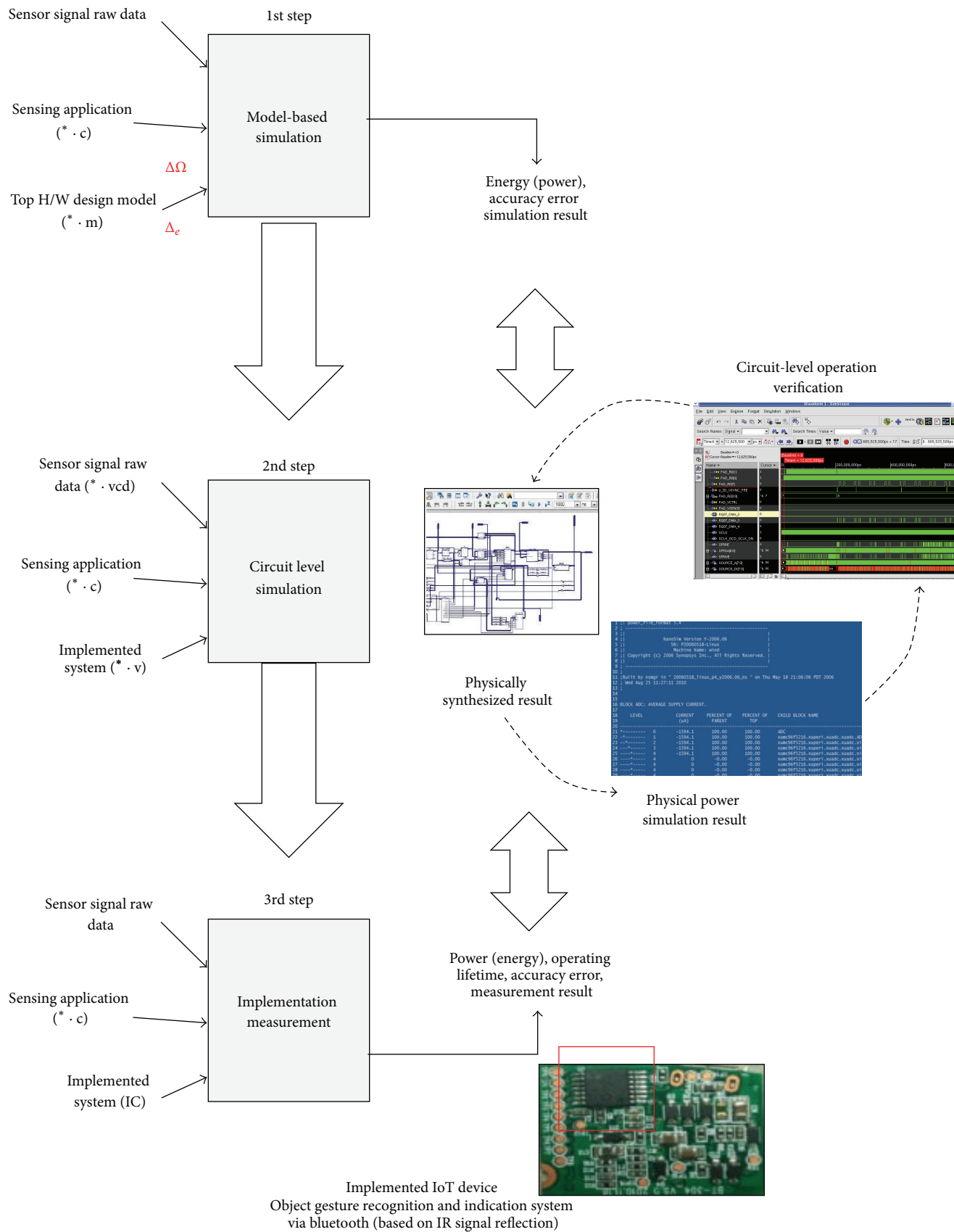
external trigger events are loaded into the on-chip via the test mode interface to emulate the dynamic operation executing user applications. The event detection timing in chip-level is compared to the expected timing and the power consumption is also measured in the real environment by the user-triggered events. The hardware overhead for the test mode interface will be excluded in a final chip for the mass production.

The third step compares the simulation results with the electrical results, which are measured for the implemented IoT sensor device, including the proposed sensor processor. Because the gate-level synthesized design files are nearly equivalent to the physical hardware, the power simulation results show that the proposed sensor processor architecture may reduce energy consumption.

The implemented IoT sensor device is a type of IR- (infrared radio-) based standalone system that senses a change in the movement of an object. The IR transmitter, IR receiver, the proposed sensor processor, Bluetooth for the wireless connectivity, and on-board battery are integrated on a single tiny PCB board, as shown in the board screenshot of Figure 10(a).

The IR transmitter generates a specific pattern of signal pulses. The IR receiver acquires the light signal reflected by the target objects, and the implemented sensor processor performs signal processing to analyze the meaning of the signal.

Figure 11 shows the experimental results by the implemented IoT device, comparing the operating lifetime according to the number of sample differences by the processing method in Figure 11(a). The S2D describes the result by the



(a) Experimental design to compare with the implementation result

FIGURE 10: Continued.

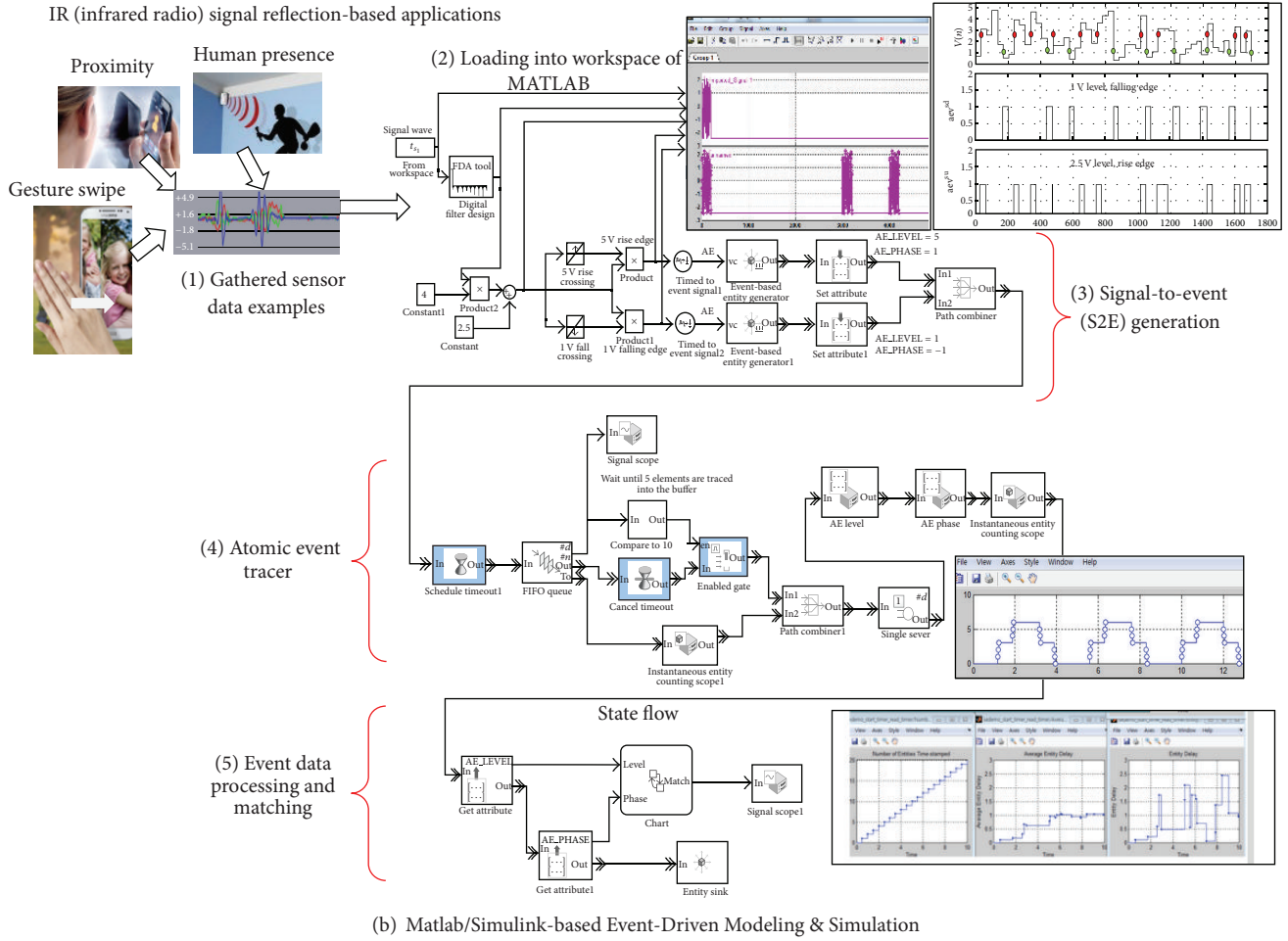


FIGURE 10: Experimental design and measurement of the implementation result.

conventional polling-based sensor data processing method. The S2T shows a result from the level-triggered interrupt-based signal processing method. The S2E shows the results from our proposed method. The improved results from allowing an accuracy error are also evaluated. These results, which allow for a 25% error variation in the specific constraints in our applications, are shown in Figures 11(a) and 11(b).

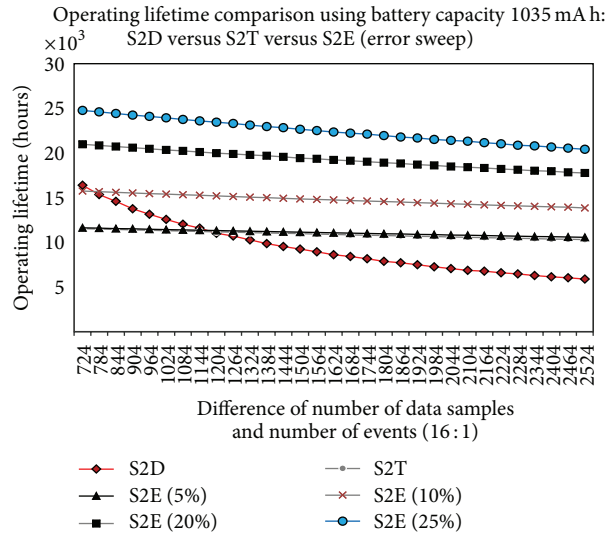
All building blocks in the implemented IoT sensor device hibernate during sleep mode except that of the EPU (including the signal-to-event converter), which is only active in order to trace the transition of the incoming signal by comparing the signal features of interest. When the final event is identified by the event-print window matcher, the main MCU wakes up to activate the subsequent building blocks to transfer the recognized events to a host system using the wireless connectivity.

As sensing applications of the implemented IoT sensor device, the low-power recognition performance based on the proposed method is evaluated in terms of its energy consumption. Figure 11(c) shows the results for the specific IR sensor signal recognition using a defined set of signal

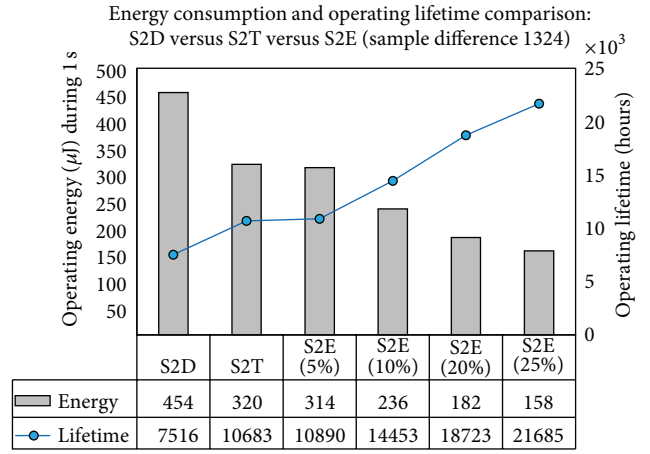
segments Ω . The figure shows an 80% reduction when a 10% accuracy error is allowed, compared to the original work.

Figure 11(d) shows the energy-efficient recognition of the hand-swipe gestures, which are described by the two types of signal segments Ω and show a 50% reduction when a 10% accuracy error is allowed, compared to the original work. The reduction in energy consumption is achieved by configuring the implemented architectural framework with application-specific constraints that allow the required recognition accuracy error.

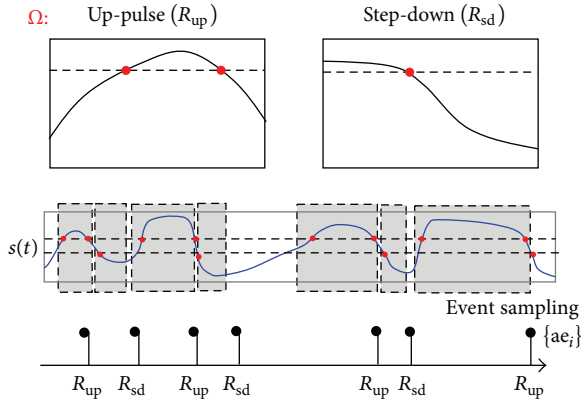
The margin of the acceptable accuracy error is dependent on the application-driven requirement. The trade-off between the event detection accuracy and low-power consumption has to be considered in implementing the IoT device. The proposed chip architecture provides the configuration register to allow the user-defined accuracy error for more long-term operation of the IoT device. In the environmental monitoring applications, such as proximity, hand gesture, temperature, and light intensity, the response error in less than several seconds for event detection is small enough to allow the 50% accuracy error.



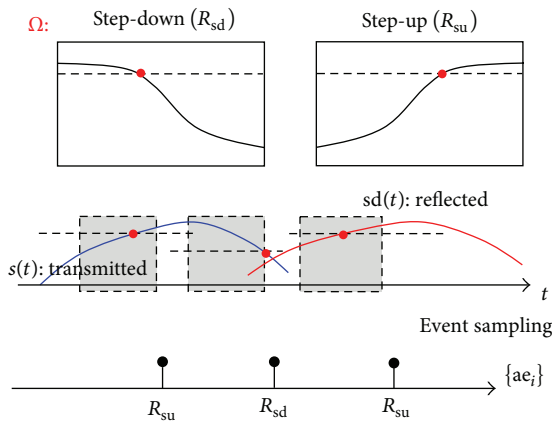
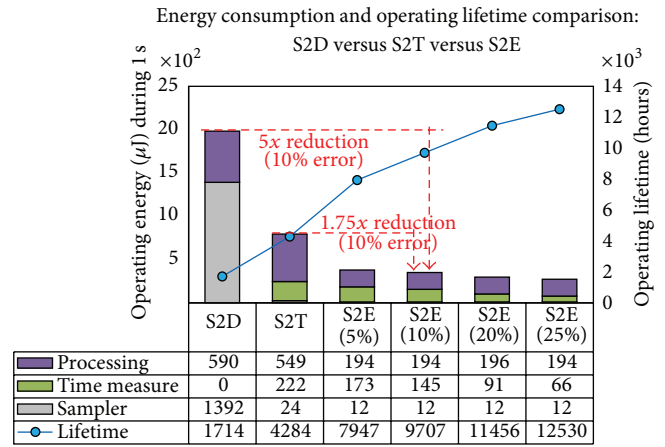
(a) Operating lifetime according to the difference of the number of samples



(b) Operating energy & lifetime comparison according to the accuracy



(c) Energy consumption of proposed method for specific IR reflection signal pattern recognition



(d) Energy consumption of proposed method for IR signal reflection arrival time distance measurement (Gesture Swipe Applications)

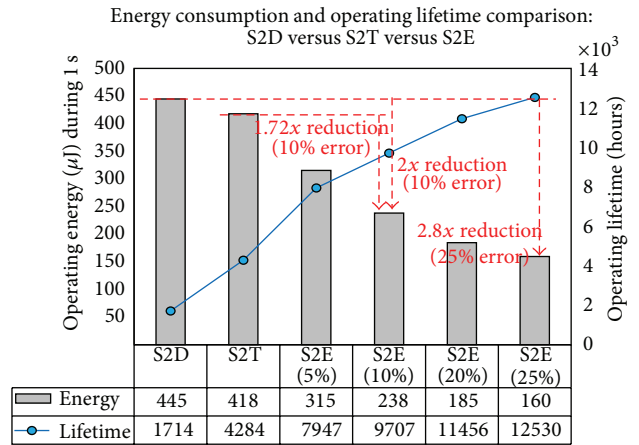


FIGURE 11: Experimental results.

6. Conclusion

The proposed event-driven sensor processor architecture for sensing applications with rare-event constraints is proposed and implemented as an accelerator. This enables the sensor signal processing in an energy-efficient mode by allowing for the accuracy error, which is caused by the abstraction of the original signal as atomic events. All of the building blocks, including atomic event generators and the EPU core, are implemented as a single system-on-a-chip, which is integrated into the IoT sensor device.

The proposed method uses the characteristics of rare-event sensing applications, in which timing accuracy error is relatively insensitive in sensor signal recognition and introduces the concept of atomic event generation as a method of event-quantization. The event-space representation of the sensor signal by the extracted set of atomic events is constructed with a user-defined event signal segmentation by the configured feature scan window.

The event-quantization result is archived as a set of unique indexes into the tracer buffer. The event-print matcher determines the presence of the featured signal points for the collected atomic event vector, in order to identify the event from the original sensor signal. The event-matching process is based on the similarity factor calculation, named R^* -plain projection that describes the expected rules of the signal patterns.

The implementation results, which are evaluated for an IR-based signal recognition system for object activity monitoring applications, show a reduction in total energy consumption by delaying the activation of the main processor and the Bluetooth interface for the wireless connectivity. The proposed sensor processor provides an architectural framework by providing an application-specific configuration of the event-quantization level for the energy-accuracy trade-off. This results in additional benefit of the energy consumption, which maximizes the operating lifetime of the IoT sensor device.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2014R1A6A3A04059410), the BK21 Plus project funded by the Ministry of Education, School of Electronics Engineering, Kyungpook National University, Korea (21A20131600011), and the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the C-ITRC (Convergence Information Technology Research Center) support program (NIPA-2014-H0401-14-1004) supervised by the NIPA (National IT Industry Promotion Agency).

References

- [1] H. T. Chaminda, V. Klyuev, and K. Naruse, "A smart reminder system for complex human activities," in *Proceedings of the 14th International Conference on Advanced Communication Technology (ICACT '12)*, pp. 235–240, 2012.
- [2] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensor-based activity recognition," *IEEE Transactions on Systems, Man and Cybernetics C: Applications and Reviews*, vol. 42, no. 6, pp. 790–808, 2012.
- [3] K. Choi, R. Soma, and M. Pedram, "Fine-grained dynamic voltage and frequency scaling for precise energy and performance tradeoff based on the ratio of off-chip access to on-chip computation times," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 1, pp. 18–28, 2005.
- [4] A. B. da Cunha and D. C. da Silva Jr., "An approach for the reduction of power consumption in sensor nodes of wireless sensor networks: case analysis of mica2," in *Proceedings of the 6th International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS '06)*, vol. 4017 of *Lecture Notes in Computer Science*, pp. 132–141, Springer, Samos, Greece, July 2006.
- [5] B. French, D. P. Siewiorek, A. Smailagic, and M. Deisher, "Selective sampling strategies to conserve power in context aware devices," in *Proceedings of the 11th IEEE International Symposium on Wearable Computers (ISWC '07)*, pp. 77–80, Boston, Mass, USA, October 2007.
- [6] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 1, pp. 124–137, 2013.
- [7] M. Hempstead, N. Tripathi, P. Mauro, G.-Y. Wei, and D. Brooks, "An ultra low power system architecture for sensor network applications," *ACM SIGARCH Computer Architecture News*, vol. 33, no. 2, pp. 208–219, 2005.
- [8] M. Hempstead, G.-Y. Wei, and D. Brooks, "Architecture and circuit techniques for low-throughput, energy-constrained systems across technology generations," in *Proceedings of the International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES '06)*, pp. 368–378, New York, NY, USA, October 2006.
- [9] A. B. Kahng and K. Seokhyeong, "Accuracy-configurable adder for approximate arithmetic designs," in *Proceedings of the ACM/EDAC/IEEE 49th Annual Design Automation Conference (DAC '12)*, pp. 820–825, ACM, San Francisco, Calif, USA, June 2012.
- [10] K. van Laerhoven, H.-W. Gellersen, and Y. G. Malliaris, "Long-term activity monitoring with a wearable sensor node," in *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks (BSN '06)*, pp. 171–174, April 2006.
- [11] Ó. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013.
- [12] M. T. Lazarescu, "Design of a WSN platform for long-term environmental monitoring for IoT applications," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 1, pp. 45–54, 2013.
- [13] S.-W. Lee and K. Mase, "Activity and location recognition using wearable sensors," *IEEE Pervasive Computing*, vol. 1, no. 3, pp. 24–32, 2002.

- [14] Y. Lee, J. Kim, and C.-M. Kyung, "Energy-aware video encoding for image quality improvement in battery-operated surveillance camera," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 2, pp. 310–318, 2012.
- [15] K. Leuenberger and R. Gassert, "Low-power sensor module for long-term activity monitoring," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC '11)*, pp. 2237–2241, 2011.
- [16] W. Li, M. Bandai, and T. Watanabe, "Tradeoffs among delay, energy and accuracy of partial data aggregation in wireless sensor networks," in *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA '10)*, pp. 917–924, April 2010.
- [17] M. Miskowicz, "The event-triggered integral criterion for sensor sampling," in *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE '05)*, vol. 3, pp. 1061–1066, 2005.
- [18] P. Panek, "Error analysis and bounds in time delay estimation," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3547–3549, 2007.
- [19] D. Park, C. M. Kim, S. Kwak, and T. G. Kim, "A low-power fractional-order synchronizer for syncless time-sequential synchronization of 3-D TV active shutter glasses," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 2, pp. 364–369, 2013.
- [20] B. Schell and Y. Tsividis, "Analysis of continuous-time digital signal processors," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '07)*, pp. 2232–2235, 2007.
- [21] J. Sorber, A. Balasubramanian, M. D. Corner, J. R. Ennen, and C. Qualls, "Tula: balancing energy for sensing and communication in a perpetual mobile system," *IEEE Transactions on Mobile Computing*, vol. 12, no. 4, pp. 804–816, 2013.
- [22] Y. Tsividis, "Event-driven data acquisition and continuous-time digital signal processing," in *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC '10)*, pp. 1–8, 2010.
- [23] Y. Tsividis, "Event-driven data acquisition and digital signal processing—a tutorial," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 8, pp. 577–581, 2010.
- [24] Y. Yilmaz, G. Moustakides, and X. Wang, "Spectrum sensing via event-triggered sampling," in *Proceedings of the 45th Asilomar Conference on Signals, Systems and Computers (ASILOMAR '11)*, pp. 1420–1424, November 2011.