Reducing serial I/O power in error-tolerant applications by efficient lossy encoding

The MIT Faculty has made this article openly available. Please share how this access benefits you. Your story matters.

Citation

As Published
http://dx.doi.org/10.1145/2897937.2898079

Publisher
Association for Computing Machinery

Version
Author's final manuscript

Accessed
Sat Dec 15 03:14:06 EST 2018

Citable Link
http://hdl.handle.net/1721.1/113651

Terms of Use
Creative Commons Attribution-Noncommercial-Share Alike

Detailed Terms
http://creativecommons.org/licenses/by-nc-sa/4.0/
Reducing Serial I/O Power in Error-Tolerant Applications by Efficient Lossy Encoding

Phillip Stanley-Marbell
MIT

Martin Rinard
MIT

ABSTRACT
Transferring data between integrated circuits (ICs) accounts for an important fraction of the power dissipation in wearable and mobile systems. Reducing signal transitions reduces the dynamic power dissipated in the data transfer between ICs. Techniques such as Gray coding to reduce transitions between two parallel words cannot be applied when the signal transitions are between bits of a single serialized word.

This paper introduces value-deviation-bounded serial encoding (VDBS encoding). VDBS encoding significantly reduces signal transitions between bits of a single serialized word, trading power efficiency for data accuracy. This tradeoff is worthwhile when the data are from sensor sources such as sensors and destined for consumption by signal processing algorithms that are error-tolerant.

We present analytic formulas for the Pareto-optimal VDBS encoders and introduce an efficient algorithm, Rake, that reduces transitions almost as much as the optimum transition-reducing encoder. We evaluate Rake by encoding data in a pedometer system and in a text-recognition system. For the pedometer, Rake reduces transitions by 54% on average, in exchange for step count errors smaller than 5%. For the text recognizer, Rake reduces transitions by 55% on average, while maintaining OCR accuracy above 90%.

1. INTRODUCTION
Wearable computing platforms such as health-tracking and head-mounted systems present new challenges to energy-efficient design. Unlike desktop and mobile systems, they function primarily with their displays (if any) turned off. They spend a majority of their time reading data from sensors such as pressure sensors in elevation monitoring, accelerometers and gyroscopes in step counting applications, color/light sensors in pulse oximeter health applications, and cameras in head-mounted augmented-reality systems.

1.1 Sensor power dissipation is important
The processors in wearable, embedded, and mobile platforms are usually the main focus of power-reduction efforts. These processors are however often connected to many sensor integrated circuits (ICs). Because the power-efficiency for inter-IC communication is limited by printed circuit board properties, it has not scaled with semiconductor process technology and packaging advances [25]. As a result, the power dissipated in some state-of-the-art sensors is close to the power dissipation of low-power processors (Figure 1).

I/O energy costs range from 10 fJ/bit/mm to 180 fJ/bit/mm in on-chip links [12, 20], to between 2 pJ/bit and 40 pJ/bit for typical printed circuit board (PCB) traces [8, 13, 14, 17]. At data rates of 1 MB/s typical of modern embedded serial links, these energy costs per bit lead to I/O power dissipation between 2 µW and 40 µW. This is up to 13% of the processor’s power dissipation in Figure 1 and is incurred for each sensor in a system. I/O energy is thus an important problem in today’s low-power embedded systems, with no obvious solutions on the horizon.

To enable smaller device packages, smaller PCB designs, and lower costs, the inter-IC communication links in many embedded computing platforms are bit-serial and not parallel buses. Prior efforts to reduce communication power by encoding data using techniques such as Gray coding have however targeted parallel buses [3, 23] and cannot be applied to reduce data transfer power in bit-serial communication interfaces.

1.2 Value-deviation-bounded serial encoding
Because the data from sensors are often generated by a process with some inanimate noise, algorithms that consume sensor data are usually robust to small or occasional errors even if they typically require high-resolution data. Motivated by this observation, this paper introduces value-deviation-bounded serial encoders (VDBS encoders). VDBS encoders reduce signal transitions and thereby reduce dynamic power dissipation in serial communication interfaces, by permitting a selectable amount of deviation from correctness in transmitted data. Because the optimal and efficient encodings we present can be computed offline and deployed using lookup tables, their overheads in practical applications are small.

Figure 1: Active power dissipation for several state-of-the-art sensors [2, 22, 30, 31], a Bluetooth LE radio [29] in advertising mode, and an ARM Cortex M0+ [7] microcontroller running a while(1) loop from on-chip SRAM at 2 MHz and 3.0 V.
VDBS-encoded values are interpreted as though they were unencoded and VDBS therefore requires no decoder hardware. And, unlike techniques based on T0 encoding [1], VDBS encoders do not require an additional control wire on the bus. Because they require no changes to the electrical interface or to the receive side of communication links, VDBS encoders can be integrated into production systems that use interfaces such as SPI [15] and I2C [16].

### 1.3 Contributions

This work makes the following four contributions. It presents:

1. **Analytic formulas for the optimal VDBS encoders** (Section 3). These formulas can be used to generate encoding tables offline.

2. **An efficient VDBS encoder, Rake** (Section 4). Rake is linear-time in input word size. Rake reduces transitions almost as much as the optimal transition-reducing VDBS encoder and induces almost as little deviation as the optimal deviation-minimizing VDBS encoder.

3. **An evaluation of the optimal and Rake VDBS encoders** (Section 5). Rake reduces signal transitions by 67% on average when targeting a worst-case value deviation of 10% in 8-bit values. For target worst-case value deviations of 0.12% of the full-scale range for 16-bit values, Rake reduces signal transitions by 41% on average. The evaluation results show that VDBS encoders reduce transitions more than simply representing values with shorter words of equivalent effective number of bits.

4. **Two end-to-end evaluations of Rake** (Section 6). We evaluate Rake encoding data from a camera in a text-recognition application, using 392 images of text from natural scenes. Rake reduces signal transitions by an average of 55% while maintaining OCR accuracy above 90%. We also evaluate Rake encoding data from an accelerometer in a pedometer application, using over 4.6 hours of accelerometer data taken from 12 different users. Rake reduces signal transitions by 54% on average, inducing step count errors of less than 5%.

### 2. RELATED RESEARCH

Data from processors to memories must be transferred accurately. Techniques for reducing power dissipation on processor-memory buses [3] therefore employ substitution codes such as Gray codes, bus invert codes [23], or T0 codes [1]. In these prior approaches, data are recovered exactly after decoding. VDBS encoding, by contrast, requires no decoding, but is lossy.

Because of crosstalk and pin limitations among other things, both state-of-the-art high-performance systems as well as energy-sensitive wearable platforms predominantly use serial communication interfaces between ICs [10, 25]. Because serial interfaces transmit one bit of a word at a time however, they can not benefit from low-power encodings developed for parallel buses.

Low-power encodings for serial video data [4, 19] exploit tonal locality in images to reduce transitions in exchange for data representation overheads. Other approaches to reducing transitions include representing values with fewer bits [28], or using transition encoding [6]. As we show in Section 5, VDBS encoders are more effective at reducing transitions than approaches that simply reduce the number of bits transmitted.

Many signal processing [9] and recognition, mining, and synthesis applications [5] can tolerate errors in their input data. This motivates low-power encodings that trade sensor data accuracy for lower power dissipation. Prior work on the theoretical upper limits to VDBS encoder efficiency [24, 26] showed the feasibility of VDBS encoding. In this paper, we present the first optimal and efficient formulations for VDBS encoding [27] (Sections 3 and 4), we evaluate VDBS encoding numerically (Section 5), and we evaluate VDBS encoding in two end-to-end applications (Section 6).

### 3. PARETO-OPTIMAL VDBS ENCODERS

VDBS encoding generalizes the idea illustrated in Figure 2. Considering both STC reduction and induced deviation, the Pareto-optimal VDBS encoders either minimize the induced deviation, maximize the STC reduction, or both.

**Definition 1 (Family of optimal VDBS encoders).** Let $s$ and $t$ be two unsigned $l$-bit integers representing unencoded and encoded words, respectively. Let $m$ be the difference in numeric value between $s$ and $t$, and let $\#_k(k)$ be the STC for an integer $k$. We define a Boolean predicate $P_{s,t,m}$ to denote the constraint satisfied by all VDBS encoders that maintain or reduce STCs while inducing a deviation less than or equal to $m$:

$$P_{s,t,m} = (|s - t| \leq m) \land ((\#_s(s) - \#_t(t)) \geq 0).$$

Let $\Delta_{s,t} = |\#_s(s) - \#_t(t)|$ be the difference in serial transition counts between two words $s$ and $t$. Given an input word $s$ and integer $m$ indicating how much deviation in $s$ is acceptable, there are four possible encoding functions that satisfy the Boolean predicate $P_{s,t,m}$. These functions define the bounds on transition reduction and value deviation:

$$e_1(s, m) = \left( \tau \land P_{s,\tau,m} \land (|s - \tau| = \min_{0 \leq i < 2^l - 1} |s - i|) \right),$$

$$e_2(s, m) = \left( \tau \land P_{s,\tau,m} \land (|s - \tau| = \max_{0 \leq i < 2^l - 1} |s - i|) \right),$$

$$e_3(s, m) = \left( \tau \land P_{s,\tau,m} \land (\Delta_{s,\tau} = \min_{0 \leq i < 2^l - 1} \Delta_{s,i}) \right),$$

$$e_4(s, m) = \left( \tau \land P_{s,\tau,m} \land (\Delta_{s,\tau} = \max_{0 \leq i < 2^l - 1} \Delta_{s,i}) \right).$$

In what follows, we restrict our treatment to unsigned integers. The analysis easily extends to two’s-complement, fixed-, and floating-point representations.

![Figure 2: VDBS encoding reduces transitions while incurring a value deviation, $|s - t|$, of 1. In this example, all bits except the most-significant bit are modified (shown shaded in (b)), not just the lower $\lfloor \log_2(|s - t|) \rfloor$ bits.](image-url)
3.2 Properties of the optimal VDBS encoders

The four functions $e_1$ through $e_4$ bound the amount by which VDBS encoders reduce STCs and bound the deviation they induce:

- $e_1(s, m)$ causes the smallest deviations.
- $e_2(s, m)$ causes the largest deviations.
- $e_3(s, m)$ reduces STCs the least.
- $e_4(s, m)$ reduces STCs the most.

Our objective is to obtain a method for VDBS encoding whose behavior encompasses the best of the properties of all the above encoders: induced deviation close to that of $e_1$ and STC reduction close to that of $e_4$.

The subset of three encoder types $e_1$, $e_3$, and $e_4$ are Pareto-optimal when considering both serial transition reduction and deviation. Because it is strictly dominated by $e_4$, the encoder $e_2$ is not in the Pareto set. The behavior of the simplistic encoder that for a given tolerable deviation $m$ only removes transitions from the lower $\lceil \log_2(m) \rceil$ bits is similar to $e_2$ (Section 5).

4. Rake: EFFICIENT VDBS ENCODING

Given an unencoded value $s$ in which an application can tolerate a value deviation $m$, the family of encoders of Definition 1 specify the possible optimum ways in which encoding can reduce serial transitions in $s$. The encoders of Definition 1 also determine the amount of deviation that an encoding will induce, for a given selected deviation that applications can tolerate. Exact algorithms for the optimal encoders must however select an encoded value for a given tolerable deviation that constitutes a transition could be offset by the runs of lower-order bits of opposite polarity, as represented by the contents of the cumulative run contribution arrays.

To address the cost of the Pareto-optimal encoders of Definition 1, particularly for large word sizes, we present Rake, an efficient algorithm for VDBS encoding. Rake’s execution time is linear in the word size of the values it encodes. For a specified deviation $m$ in its encoded values, Rake reduces transitions more than the basic technique that simply removes all transitions from the lower-order $\lceil \log_2(m) \rceil$ bits. At the same time, Rake reduces transitions almost as much as the Pareto-optimal VDBS encoder $e_1$ that minimizes the serial transition count for a given tolerable deviation. On average, Rake incurs value deviations smaller than all the Pareto-optimum VDBS encoders except $e_1$ (which minimizes value deviation). We call the algorithm Rake because it operates in two sweeps of a word, accumulating metadata in the first sweep and leveling out transitions in the second. The Rake algorithm (Algorithm 1) operates as follows.

In the first phase (lines 1 to 6), moving across the $l$-bit input word $s$ from least-significant bit (LSB) to most-significant bit (MSB), Rake stores the number of transitions seen to-date in the transition count register, $nt$. Rake stores the indices of these transitions in the transition indices array, $tr$ (line 2). For each transition, Rake stores the length of the run of 0s or 1s leading to the transition, in the run length temporary register, $rl$ (line 3). Each such run of 0s or 1s could be bit-wise negated to either increase or decrease the value of $s$. Rake stores the change in value that such a negation would contribute, in the cumulative run contribution arrays, $cr0c$ for runs of 0s and $cr1c$ for runs of 1s (lines 4 and 5).

In the second phase (lines 7 to 10), Rake moves across the input in the opposite direction, from MSB to LSB, inspecting only the $nt$ bit positions that have transitions. Rake previously stored these locations in $tr$. For each of the $nt$ transition locations in $tr$, Rake checks whether the value deviation incurred by negating the bits that constitute a transition could be offset by the runs of lower-order bits of opposite polarity, as represented by the contents of $cr0c$ and $cr1c$ (lines 8 and 9). Rake removes the first transition that passes this check and completes. Rake takes 8 steps as it traverses from the LSB to the MSB, followed by at most $nt - 2$ steps in the opposite direction. The maximum value of $nt$ is $l - 1$, thus Rake takes a maximum of $2l - 3$ steps. For example, for 24-bit values, Rake requires only 45 steps, compared to having to explore a space of 16 million values for the exact optimal solution.

Rake is not only efficient, but also effective: Rake reduces transitions almost as much as the optimal VDBS encoder $e_4$ as we show in Section 5. By contrast, the naive approach of simply removing transitions from the lower-order $\lceil \log_2(m) \rceil$ bits for a tolerable value deviation of $m$ does not reduce transitions as much as Rake does.

5. NUMERICAL EVALUATION

Two objective metrics are important for VDBS encoders:

- **The average serial transition count reduction** for a given word size and tolerable deviation.
- **The average actual deviation** that is induced by encoders for a given tolerable deviation.
5.1 Transition reduction and induced deviation

We evaluate Rake and the Pareto-optimal encoders by applying them to all possible unsigned values for a given word size, \( l \). The word sizes we evaluate are \( l = 8 \) and \( l = 16 \). For each word size, we select 10 values of tolerable deviation, \( m \), uniformly spaced between 0 and 50. For each value of tolerable deviation, we apply each of the Rake and Pareto-optimal encoders to all \( l \)-bit values. From the resulting \( 2^l \) encoded values for each encoder, we compute the mean serial transition reduction at each value of tolerable deviation \( m \). From the encoded values paired with their original unencoded values, for each encoder, we compute the mean induced deviation at each tolerable deviation \( m \). Figure 3 presents the results. The figure plots the percentage reduction in serial transition count and the average value deviation resulting from encoding, as functions of the tolerable deviation specified during encoding (expressed as a percentage of the full-scale range of \( l \)-bit values).

The top row of Figure 3 shows the results for 8-bit values. For a tolerable deviation of 10 % of the full-scale range of 8-bit values, Rake reduces signal transitions by 67 %. For this tolerable deviation, the mean actual deviation is 4 % of the full-scale range (i.e., 10). The results in Figure 3 show that Rake reduces serial transitions more than all but one of the Pareto-optimal encoders: Rake reduces transitions by only 5 percentage points less than the optimal encoder that minimizes serial transitions (\( e_1 \)). The average deviation induced by Rake is also better than all but one of the Pareto-optimal encoders: At a tolerable deviation of 10 % of the full-scale range, Rake’s induced deviation is less than 4 percentage points worse than the optimal encoder that minimizes deviation (\( e_5 \)). Even at moderate tolerable deviations of 5 % of the full-scale range, Rake reduces transitions almost twice as much as existing encoding techniques for deviation-free serial buses [6].

The results for 16-bit words follow a similar trend (bottom row of Figure 3). For a tolerable deviation of 0.12 % of the full-scale range, Rake reduces signal transitions by 41 % on average, while inducing deviations of 0.05 % of the full-scale range, on average.

5.2 Effective number of bits of encoded values

The effective number of bits (ENOB) denotes the number of unique levels representable by encoded values and is computed as \( \log_2 \left( \left| \left\{ \text{unique encoder output values} \right\} \right| \right) \). Representing values with fewer bits reduces the number of signal transitions within transmitted words and in the clock signal. Figure 4 presents the serial transition count reduction as a function of the ENOB, for Rake-encoded 8-bit words as well as for progressively shorter unencoded words. For a given ENOB, Rake encoding of 8-bit words reduces transitions up to 60 % more than simply employing shorter unencoded words that have the same ENOB.

VDBS encoders such as Rake have several additional advantages over simply employing smaller word sizes. VDBS encoding reduces transitions without requiring changes to the datapath of applications (e.g., without requiring changes to algorithms to use 5-bit data instead of 8-bit data). And VDBS encoding provides 7.4-times finer-grained control of the amount of transition reduction, because it enables fractional steps in the ENOB.

6. END-TO-END EVALUATION

We evaluate Rake in two end-to-end application settings. The evaluation results indicate that Rake can significantly reduce signal transitions in exchange for small deviations in encoded values. Because these deviations are often masked by the data-flow of common sensor signal processing algorithms, the deviations lead to only small errors at the application level.

6.1 Encoding data in a text-recognition system

We apply Rake to images in transfer between a camera and processor in a text-recognition system such as that illustrated in Figure 5. Text recognition is an important component of many applications, such as augmented reality systems. We evaluate the amount
by which Rake reduces data transfer signal transitions as well as its end-to-end effect on optical character recognition (OCR) errors.

We use version 3.02 of the Tesseract OCR system [21], widely regarded to be the most accurate open-source OCR package. For input, we use the test set from the ICDAR text image dataset [32] and select as our baseline the 392 images (Figure 6) for which Tesseract returns the same recognition text as the benchmark’s ground truth. We then apply Rake to each of these 392 text images, with degrees of tolerable deviation ranging from 0 % to 20 % of the full-scale range of the 8-bit per-color-channel pixel values. We quantify the errors in text recognition using the standard edit-distance-based metric used in the text-recognition literature [18]. Figure 7 presents an example of the effect of Rake on two input text images, as well as the effect on OCR accuracy and on transitions in the serialized image data. The examples in Figure 7 illustrate how Rake applied to image data can significantly reduce transitions without affecting the output of OCR algorithms applied to the images.

Figure 8 presents Rake’s serial transition count reduction and its induced reduction in OCR accuracy as functions of the tolerable deviation in encoded values. The results in Figure 8 show that Rake reduces transitions significantly with minimal effect on OCR error. With a target tolerable deviation of 5 %, Rake reduces serial transitions by over 55 %, while maintaining an OCR accuracy of over 90 % for previously-correctly-recognized text.

6.2 Encoding data in a pedometer system

We apply Rake to accelerometer data in a pedometer system (Figure 9). Pedometer facilities are central to many health and wellness applications and these applications constitute a growing market with important positive societal impact.

Figure 6: 392 image subset from the ICDAR text recognition dataset [32] used in evaluation. This is the subset for which Tesseract [21] correctly reports OCR text identical to the benchmark-supplied ground truth.

Figure 7: Effect of Rake on OCR, for two different images (first row, second and fifth columns). Higher tolerable deviation enables more transition reduction, at the cost of OCR errors.

Figure 8: Averaged across the 392 images, for a target tolerable deviation of 5 %, Rake reduces the serial transition count (STC) by 55 % while maintaining OCR accuracy at over 90 %.

Figure 9: A VDBS encoder within a pedometer system.

We use 3-axis accelerometer data sampled at 20 Hz, a total of 334377 samples or over 4.6 hours worth of walking. The samples are taken from 12 different users in the publicly-available WISDM activity recognition dataset [11]. The WISDM dataset provides real-valued samples. In practice, however, actual accelerometer sensors provide a fixed number of bits of resolution, either directly or through the use of an ADC. We therefore convert the samples to 13-bit data, with degrees of tolerable deviation ranging from 0 % to 5 % of the full-scale range of values, before passing the encoded data to a step counting algorithm [33]. We then apply Rake to the 13-bit data, with degrees of tolerable deviation ranging from 0 % to 5 % of the full-scale range of values, before passing the encoded data to a step counting algorithm [33]. Figure 10 presents the resulting reduction in serial transition count and the induced step count errors as functions of the tolerable deviation. The results show that at target tolerable deviations of 4 %, Rake reduces transitions by up to 63 % with a mean of 54 %, inducing step counting errors of less than 5 % on average.

7. CONCLUSION

Wearable and health-tracking devices dissipate important fractions of their energy on sensor activation and data transfer. Since package and circuit board capacitances do not improve with semiconductor process advances, the fraction will continue to grow relative to components such as processors. For reasons of space and cost however, the data transfer happens over serial interfaces, not over parallel buses. This precludes encodings such as Gray codes.

Value-deviation-bounded serial encoding (VDBS encoding) reduces the dynamic power dissipation of serial data communication when applications tolerate deviations in the data values being transmitted. This paper introduces the first formulations of optimal VDBS encoders along with an efficient VDBS encoder, Rake. We evaluate the optimal and Rake VDBS encoders through numerical studies and evaluate Rake in two end-to-end applications: encoding image data within an OCR application and encoding accelerometer data within a pedometer.

Our evaluation results show that Rake performs close to optimal in reducing serial transitions. For the OCR system, Rake reduces signal transitions (and hence dynamic power dissipation of data transfer) by 55 % on average, while maintaining OCR accuracy at over 90 % for previously-correctly-recognized text. For the
pedometer system, Rake reduces signal transitions by 54% on average, while causing less than 5% error in reported step counts, on average.

References


