An Area- and Energy-Efficient FIFO Design Using Error-Reduced Data Compression and Near-Threshold Operation for Image/Video Applications

Seyed Mohammad Ali Zeinolabedin, *Student Member, IEEE*, Jun Zhou, *Senior Member, IEEE*, Xin Liu, *Member, IEEE*, and Tony Tae-Hyoung Kim, *Senior Member, IEEE*

***Abstract*— Many image/video processing algorithms require FIFO for filtering. The FIFO size is proportional to the length of the filters and input data width, causing large area and power consumption. We have proposed an energy- and area-efficient FIFO design for image/video applications through FIFO with error-reduced data compression (FERDC) and near-threshold operation. On architecture level, FERDC technique is proposed to reduce the size and power consumption of the FIFO by utilizing the spatial correlation between neighboring pixels and performing error-reduced data compression together with quantization to minimize the mean square error (MSE). On circuit level, near-threshold operation is adopted to achieve further power reduction while maintaining the required performance. To demonstrate the proposed FIFO, it has been implemented using a 0.18-μm CMOS process technology. The implementation covers different FIFO length, including 128, 256, 512, and 1024. The experimental results show that the proposed FIFO operating at 0.5 V and 28.57 MHz achieves up to 99%, 65%, and 34.91% reduction in dynamic power, leakage power, and area, respectively, with a small MSE of 2.76, compared with the conventional FIFO design. The proposed FIFO can be applied to a wide range of image/video signal processing applications to achieve high area and energy efficiency.**

***Index Terms*— Data compression, FIFO, image/video processing, low power, near threshold.**

I. Introduction

HARDWARE implementation of high computational complexity image/video processing algorithms [1], such as video codec [2], computational photography [3], wavelet transform [4], [5], Laplacian pyramid [6], and biomedical applications [7], has been facilitated by the rapid advancement in VLSI design. Many image/video processing algorithms require FIFOs for filtering [8] or data synchronization [9] between different blocks. In these applications, the size of the FIFO is proportional to the lengths of the filters and the input data width, which significantly affects the total area and power of a design. According to studies in [8], 50%–80% of the power consumption is related to FIFO and data transfers.

Various approaches have been proposed to reduce the size and power consumption of FIFO in image/video applications. They can be categorized as architecture-level and circuit-level techniques. The architecture-level techniques mainly focus on sharing FIFOs [9] or changing the original algorithms so that they require either less FIFOs [10] or less frequent FIFO access [11]–[14].

The H.265/High Efficiency Video Coding decoder implemented in [9] shows an example of FIFO sharing technique. In conventional design, five FIFOs, occupying 75% of the total area, are needed in intrapredictor (IP) and deblocking filter (DF) blocks, respectively. In [9], first FIFO is shared between IP and DF by adding share above line buffer block, leading to 20% area reduction in the price of 5% overhead in computational complexity. A 16-core graphical processing unit is developed in [10] for mobile application. To drain less power, the approximated technique using wavelet is proposed. It replaces the lower scale of a flat texture adaptively by the higher one. Therefore, the average access to FIFO is reduced by 24.57% at the price of the quality degradation. The proposed discrete wavelet transform architecture [11], [12] uses tile-based image segmentation along with data interleaving to reduce the size of FIFO for low-frequency spectrum so as to reduce both the area and power consumption. In [13], an image scaling processor is designed for up/downscaling of an image. T-model and inversed T-model filters are used to replace the conventional filter structures to reduce the storage requirement from two FIFOs to one with increase of mean square error (MSE). The unified memory-centric architecture suitable for both separable and nonseparable 2-D finite-impulse response (FIR) filter is proposed in [14] to address the memory issue. It uses fully direct implementation of 2-D FIR filter together with FIFO sharing to achieve area-delay-power-efficient architecture. To do so, the block-based approach applying the equally partitioned FIFOs provides the essential data for block processing. Therefore, with the same amount of FIFO as conventional method, a block of input data will be processed at any time.

Manuscript received August 27, 2014; accepted October 26, 2014.

S. M. A. Zeinolabedin and T. T.-H. Kim are with the VIRTUS, IC Design Centre of Excellence, School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: seyedmoh001@e.ntu.edu.sg; thkim@ntu.edu.sg).

J. Zhou and X. Liu are with the Institute of Microelectronics, Agency for Science, Technology and Research, Singapore 117685 (e-mail: zhouj@ime.a-star.edu.sg; liux@ime.a-star.edu.sg).

Digital Object Identifier 10.1109/TVLSI.2014.2369052

The circuit-level methods, include latch-based FIFO [15], ultralow-power SRAMs [16], memory splitting, and clock gating [17]. Regarding the circuit-level methods, in feature extraction accelerator [15], a new latch-based FIFO is introduced to reduce the area and power as compared with the conventional flipflop-based FIFO, which requires two latches per cell. The resultant FIFO is 49% smaller with 62% lower energy for a 16 bits 1 k-entry FIFO compared with the flipflop-based FIFO. In [16], a decoupled 10-T SRAM cell with improved SNM, the write margin and bitline sensing margin is developed to achieve ultralow voltage operation to reduce the power consumption in FIFO or other on-chip storage. A 24-core processor in [17] for multimedia and communication application uses a packet-switched network-on-chips exploiting FIFOs to synchronize the on-chip data transfer. The 29.3% power reduction is achieved by splitting the memory with clock gating.

As discussed above, while the circuit-level approaches can be adopted for FIFO in general applications, most of the architecture-level approaches for reducing the size and power consumption of FIFO focus on optimization of the algorithm or architecture of specific filters, which are difficult to be generalized. In this paper, we propose a more general approach to improve the area and energy efficiency of FIFO through error-reduced data compression with near-threshold operation [18]. The proposed method can significantly reduce the size and power consumption of FIFO in image/video processing with negligible quality loss. It can be applied to a wide range of image/video applications, where FIFOs are indispensable.

The remainder of this paper is organized as follows. Section II introduces the usage of FIFO in image/video processing. Section III presents the details of the proposed area- and energy-efficient FIFO using error-reduced data compression and near-threshold operation. Section IV shows the experimental results and comparison between the proposed FIFO design and conventional design. Finally, the conclusion is drawn in Section V.

II. FIFO for Image/Video Processing

Filtering is an inevitable function of image/video processing algorithms, which can be generalized as

$$g(x, y) = T_{xy} * f(x, y) \tag{1}$$

where $f(x, y)$ is the input image, T_{xy} is the filter transfer function, and $g(x, y)$ is the output. The output is defined as the convolution of the pixel values located in a given filter window (T). The window is usually a square with $N \times N$ pixels, where N is an odd number to have an exact center pixel. The T can be either a 2-D filter, where T_{xy} represents the filter coefficient or two 1-D filters where t_{xy} is the coefficient of the $1 \times N$ filter. In the latter one, t is first applied to the horizontal rows of the input, followed by the vertical columns or vice versa. In either case, it requires $N-1$ FIFOs to store $(N-1)$ rows of W pixels [Fig. 1(a) and (b)], where W is the input image width. Therefore, the number of reads is N pixels during each clock cycle. Otherwise, $N \times N$ pixels have to be read every clock cycle to fill the filter window [19]. As already mentioned, FIFOs have significant impact on the circuit area and power consumption, particularly when the size of the filter, N, is large.

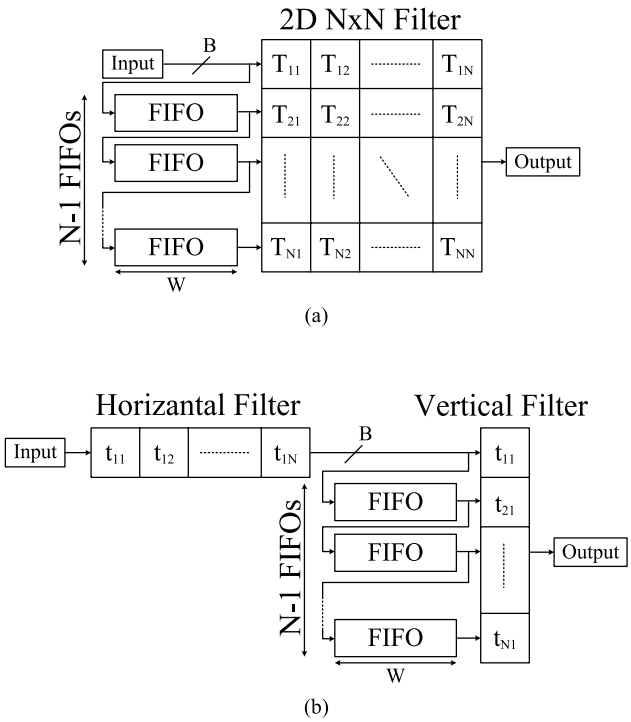


Fig. 1. Sample filter configurations. (a) 2-D filter. (b) Two 1-D filters.

III. Proposed Area- and Energy-Efficient FIFO Design

In this paper, we propose an area- and energy-efficient FIFO design. On architecture level, a technique named as FIFO with error-reduced data compression (FERDC) is proposed to reduce the FIFO size. This reduces both area and power consumption of the FIFO with negligible distortion. On circuit level, near-threshold operation is adopted to reduce the power consumption of the FIFO.

In this paper, we use MSE to evaluate the distortion introduced by the proposed technique [1]. In general cases, an acceptable MSE value is <20 [20]. Lower MSE values indicate that reconstructed images are closer to original ones.

A. FERDC Technique

Fig. 2 shows the proposed FERDC technique to realize an 8×8 filter using the architecture described in Fig. 1(b). Since there is an eight-tap vertical filter applied, seven FIFOs are required to store the data in Fig. 2. The FERDC employs a concept of pixel prediction, where every pixel can be predicted utilizing adjacent pixels and accordingly input data is horizontally decorrelated and then quantized. Finally, the encoded data read from FIFO are decoded to retrieve the original input data. Therefore, it consists of encoding and decoding parts along with the reduced-size FIFO. Therefore, in Fig. 2, all FIFOs except for the first one are fed by the output of previous FIFOs. Therefore, one encoding part is required

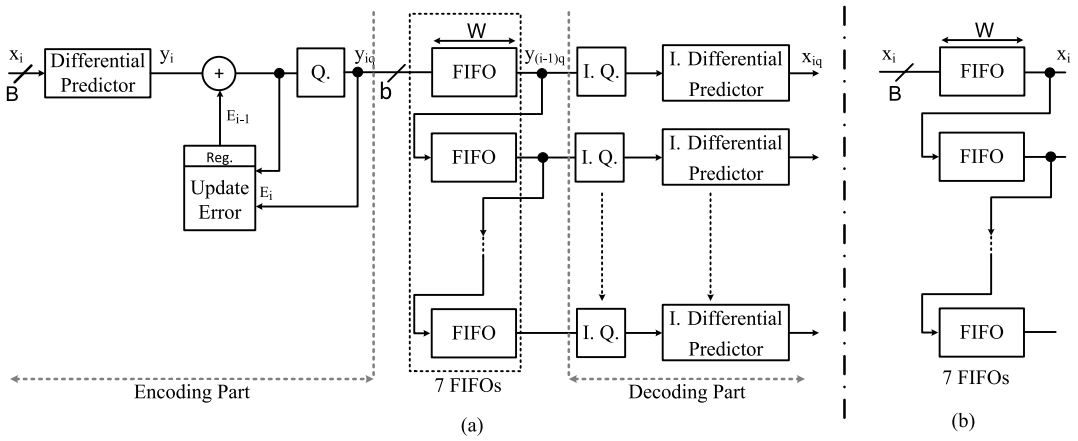


Fig. 2. FIFO architecture for an eight-tab vertical filter. (a) Using proposed FERDC technique. (b) Using conventional FIFO.

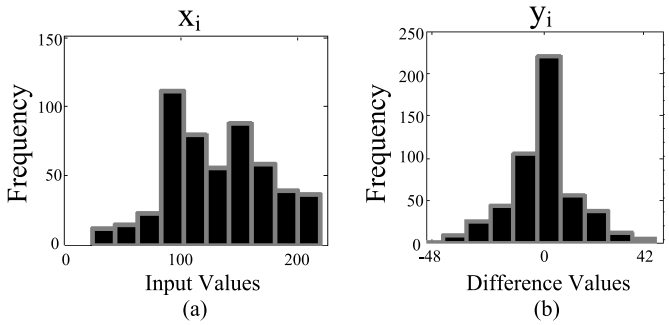


Fig. 3. Histogram of *Barbara* (first row) for (a) x_i and (b) y_i.

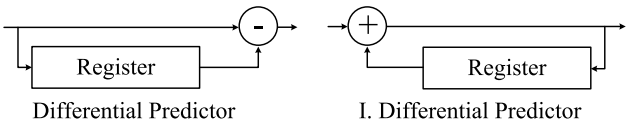


Fig. 4. Differential and inverse differential predictors' architectures.

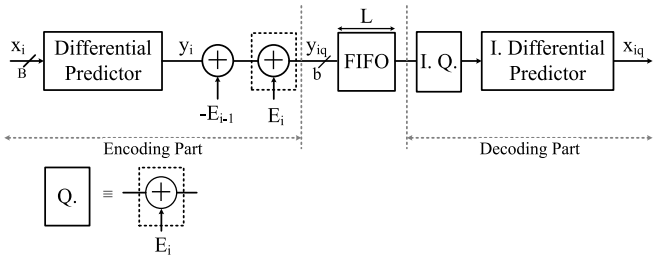


Fig. 5. Error mathematical model of FERDC.

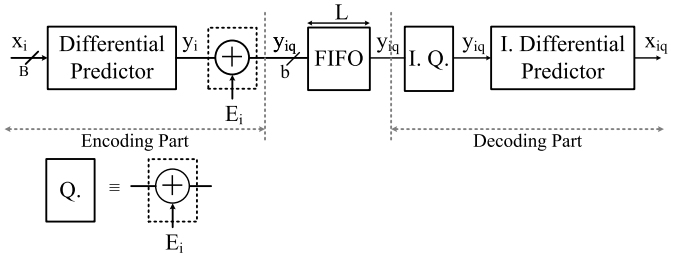


Fig. 6. Error mathematical model of FDC.

for the seven FIFOs, while one decoding part is needed at the output of each FIFO.

1) Encoder Design: For encoding, input data, x_i is given to a differential predictor, which calculates the difference between the consecutive inputs denoted as y_i. The differential predictor is to remove horizontal correlations between consecutive inputs. In that case, it ensures the energy of input correlated data, x_i [Fig. 3(a)], is compacted around zero, y_i [Fig. 3(b)] and only a few large numbers spread throughout the entire B-bit integer value range. A B-bit subtractor and adder along with a B-bit register are used for both differential predictor and inverse differential predictor, as shown in Fig. 4. Although applying the differential predictor helps to compact the energy of signal, the quantizer is required to reduce the data width as in detail explained in the following sections. By doing so, the quantization error will get accumulated at the output of the design.

Therefore, we propose an update error block to cancel the error from getting accumulated at the output. Therefore, before sending the difference values to quantizer, the energy-compacted difference is updated by the quantization error of previous difference (E_{i-1}) to prevent the quantization errors from being accumulated at the output, x_{iq}. To obtain the error mathematical model of proposed FIFO, update error and Q. (quantizer) blocks, as shown in Fig. 2, are substituted with the noise injection models as given in Fig. 5. The output can be expressed by

$$x_{iq} = y_{iq} + x_{(i-1)q}, \quad \text{for } i = 1 : x_{1q} = x_1. \tag{2}$$

After solving (2), the output, named as x_{iq_FERDC}, is

$$x_{iq\_\text{FERDC}} = x_i + E_i - E_1, \quad E_{T\_\text{FERDC}} = E_i - E_1. \tag{3}$$

To compare FERDC output error with the architecture not including update error block (FDC) (Fig. 6), similar calculation is performed on Fig. 6. The output is derived as

$$x_{iq\_\text{FDC}} = x_i + \sum_{j=2}^{i} E_j, \quad E_{T\_\text{FDC}} = \sum_{j=2}^{i} E_j. \tag{4}$$

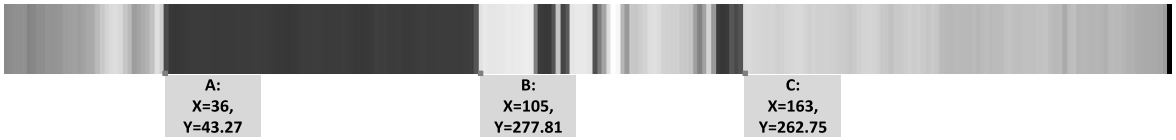


Fig. 7. Row 310 of the horizontal low-pass filtered image *Cameraman.*

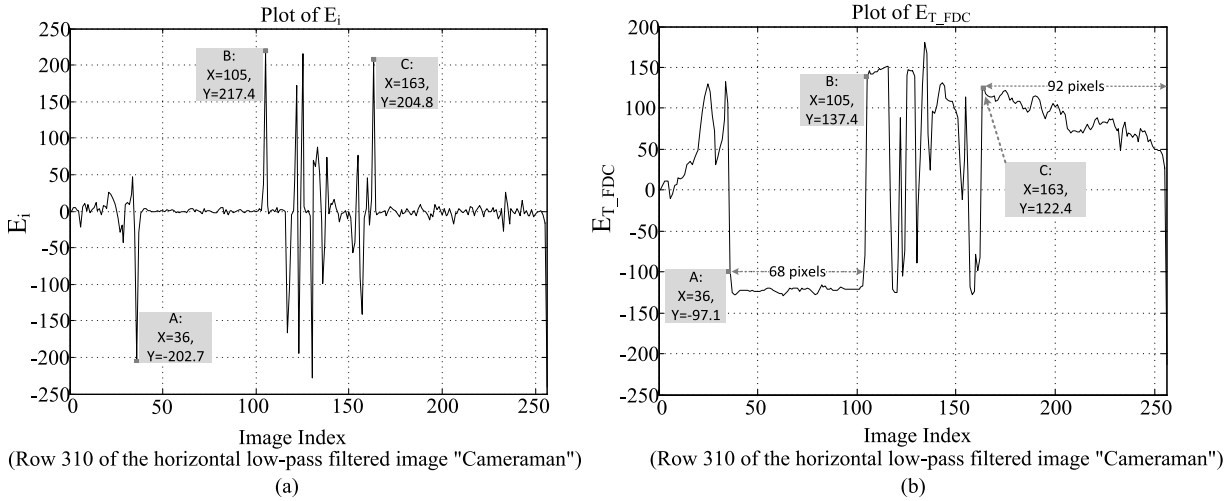


Fig. 8. Error of the quantized difference values in Fig. 7. (a) Partial error, E_i. (b) Accumulated error, E_T_FDC.

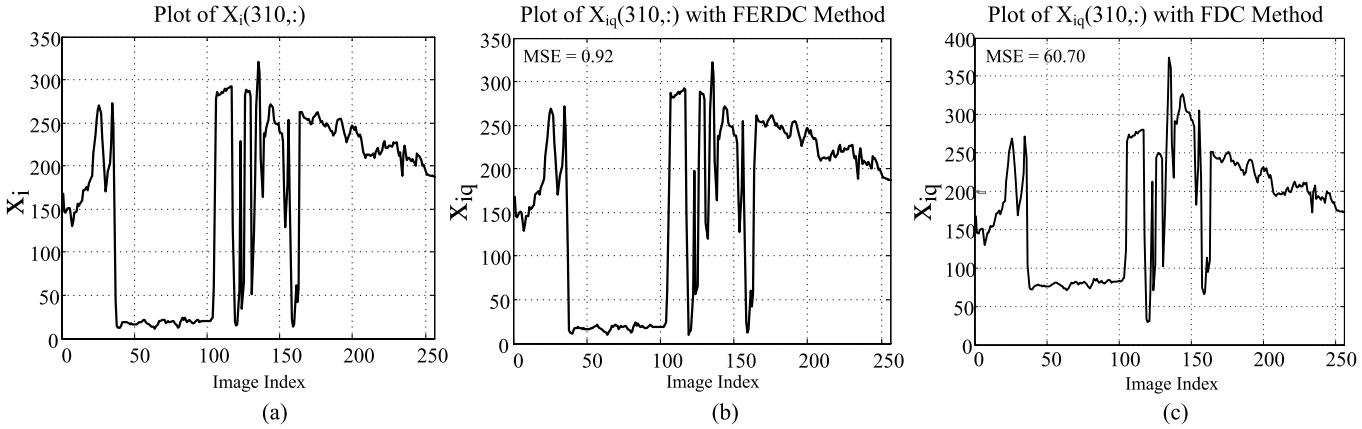


Fig. 9. Comparison of outputs for FERDC and FDC methods with reference to original input. (a) Original input. (b) x_{iq}_FERDC. (c) x_{iq}_FDC.

As clearly seen in (4), the errors in the output are added together for any given i.

It may cause the large accumulated error (E_T), if any large partial error (E_i) arises during the quantization and is not offset by next partial errors while processing the row i. Whereas the error component in (3) is considerably reduced to error difference of the ith input and the first input.

For instance, the 310th row of horizontal low-pass filtered image *Cameraman* is shown in Fig. 7. The partial error E_i and accumulated error E_{T_FDC} are drawn in Fig. 8. As observed in Fig. 8(a), there are some large errors, such as points A, B, and C due to quantization process. Note that in Fig. 8(b), the accumulated error of point A is added to all next 68 values coming right after that, the same happens to other large error points shown in Fig. 8(b). Therefore, it is observed that the large error imposed on the output, x_{iq_FDC}, makes MSE very large. This situation happens in the positions known as edge that is a pixel or pixels having the distinct brightness as compared with their neighboring pixels (Fig. 7). E_{T_FERDC} is almost similar to E_i as given in (3). As shown in Fig. 9, the output of FERDC [Fig. 9(b)] compared with the output of FDC [Fig. 9(c)] better resembles the original one [Fig. 9(a)]. The MSE of Fig. 9(b) is just 0.92, while MSE of Fig. 9(c) is 65.98 times larger ($\mathrm{MSE}_{\mathrm{Fig.\ 9(c)}} = 60.70$) because the error is getting accumulated in Fig. 9(c).

Now the updated difference values can be quantized to b-bit integer values to reduce the data width and thus the FIFO size significantly. To implement a power- and complexity-effective quantizer, a nonuniformly distributed symmetric quantizer is proposed following the same distribution of its

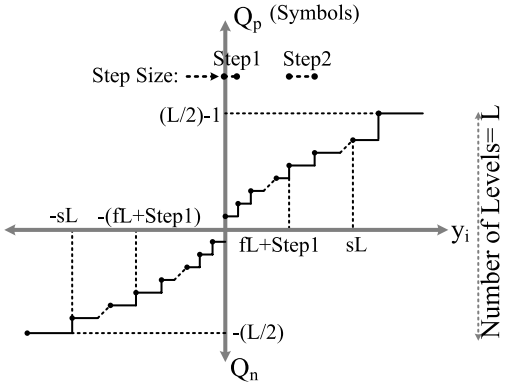


Fig. 10. Quantizer used for positive (Q_p) and negative (Q_n) numbers.

input data [Fig. 3(b)]. The quantizer comprises two parts named as Q_p and Q_n, which are applied to positive and negative numbers. Q_p and Q_n have three segments determined by fL and sL, as shown in Fig. 10. To generalize the quantizer, the first two segments are considered to have the step sizes of Step1 and Step2, respectively. The output of quantizer is just a symbol, which varies in the range of $[-L/2\ L/2)$. If the positive number, i, lies between zero and fL + Step1, it is symbolized as $\lfloor i \rfloor$. If the positive number, i, lies between fL + Step1 and sL, it is symbolized as fL + Step1 + $\lfloor (i - \text{fL} - \text{Step1})/\text{Step2} \rfloor$. Similarly, the negative numbers in the corresponding ranges are symbolized as $-\lfloor i \rfloor$ and $-(\text{fL} + \text{Step1}) + \lfloor (i + \text{fL} + \text{Step1})/\text{Step2} \rfloor$. The last segment has a constant symbol, which means that every value either larger than sL or less than −sL is represented by one symbol. Fig. 10 shows the calculation of the new FIFO cells width, b, by applying the total number of quantized levels denoted as L

$$L = 2 * \left[\text{fL} \left(\frac{1}{\text{Step1}} - \frac{1}{\text{Step2}} \right) + \text{sL} \left(\frac{1}{\text{Step2}} \right) + 3 - \frac{\text{Step1}}{\text{Step2}} \right]. \tag{5}$$

To calculate all the possible values for fL and sL with respect to Step1 and Step2, fL can be derived from (6) as the function of sL, Step1 and Step2. The maximum value of fL is achieved as (7) when sL = 0. Similarly, the maximum value of sL is also calculated when fL = 0 in (8). Among all available combinations of fL and sL, those satisfy the condition of sL > fL > 0 are selected

$$\text{fL} = \frac{\frac{L}{2} - \text{sL} \left(\frac{1}{\text{Step2}} \right) - 3 + \frac{\text{Step1}}{\text{Step2}}}{\left(\frac{1}{\text{Step1}} - \frac{1}{\text{Step2}} \right)} \tag{6}$$

$$\text{fL}_{\max} = \frac{\frac{L}{2} - 3 + \frac{\text{Step1}}{\text{Step2}}}{\left(\frac{1}{\text{Step1}} - \frac{1}{\text{Step2}} \right)} \tag{7}$$

$$\text{sL}_{\max} = \frac{\frac{L}{2} - 3 + \frac{\text{Step1}}{\text{Step2}}}{\left(\frac{1}{\text{Step2}} \right)}. \tag{8}$$

For instance, if L is 256, b becomes 8 bits. Now for a certain value of b, the quantizer parameters fL and sL can be selected by monitoring the best values of MSE (or peak signal-to-noise ratio). Besides, since many image processing algorithms are implemented with fixed-point arithmetic, B can be measured at the output of the horizontal filter [Fig. 1(b)] performed by the direct-form implementation, as shown in Fig. 11. In Fig. 11, if the very input data is assumed to be represented by K bits and the filter coefficients (tap weight), t_{ij}, are quantized into M integer bits, the worst case value of B is calculated as (10), where +1 is for the sign bit. To compensate the M-bit integer representation of the filter coefficients, an M-bit right shifter is added in Fig. 11

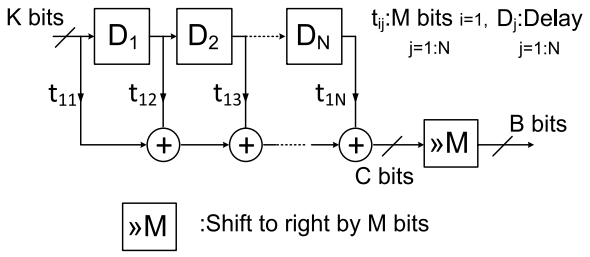


Fig. 11. Direct-form implementation of horizontal filters.

$$C = \left\lceil \log 2 \left(\sum_{j=1}^{N} |t_{1j}| \right) \right\rceil + \left\lceil \log 2 (2^k - 1) \right\rceil \tag{9}$$

$$B = C - M + 1. \tag{10}$$

2) Decoder Design: In decoding part, the read data from the FIFOs are sent to the inverse quantizer (IQ) to retrieve the b-bit data corresponding to the symbol introduced by quantizer. The positive symbol, i, between zero and fL + Step1 is corresponding to value $\lfloor i \rfloor$+0.5, the positive symbol i between fL + Step1 and sL is corresponding to value fL + Step1 + $\lfloor (i - \text{fL} - \text{Step1})/\text{Step2} \rfloor$Step2 + Step2/2 and finally, symbol$(L/2) - 1$ is retrieved as a constant value, sL + Step2/2 + 15. Similarly, the negative symbol, i, between zero and −(fL + Step1) is corresponding to value $-\lfloor i \rfloor + 0.5$, the negative symbol i between −(fL+Step1) and sL is corresponding to value $-(\text{fL} + \text{Step1}) + \lfloor (i + \text{fL} + \text{Step1})/\text{Step2} \rfloor$Step2 + Step2/2 and finally, symbol $-(L/2)$ is retrieved as a constant value, $-\text{sL} - \text{Step2}/2 - 15$. Then, the inverse differential predictor generates the output (x_{iq}) using the retrieved b-bit values (as shown in Fig. 4).

B. Near-Threshold Operation Technique

Voltage scaling has been proved to be an effective way to reduce power consumption in digital circuits and systems [21]. In conventional voltage scaling, the supply voltage is scaled only to a voltage in superthreshold region, resulting in limited power reduction. Recently, it has been shown that significant power reduction can be achieved by scaling the supply voltage to subthreshold/near-threshold region [22]–[24]. This requires new circuit design methodology called near-threshold/subthreshold design. Compared with subthreshold design, near-threshold design can achieve much better performance with only slight power increase. In the design of the proposed FIFO, near-threshold operation is adopted to achieve further power reduction.

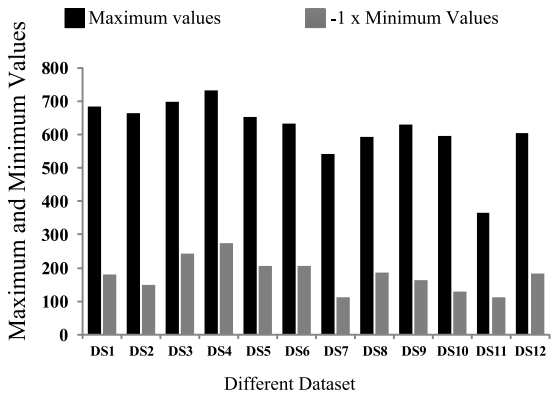


Fig. 12. Maximum and absolute minimum values of data sets.

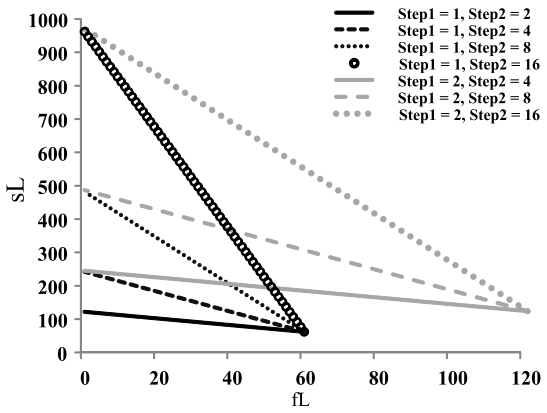


Fig. 13. fL versus sL for the various combinations of Step1 and Step2.

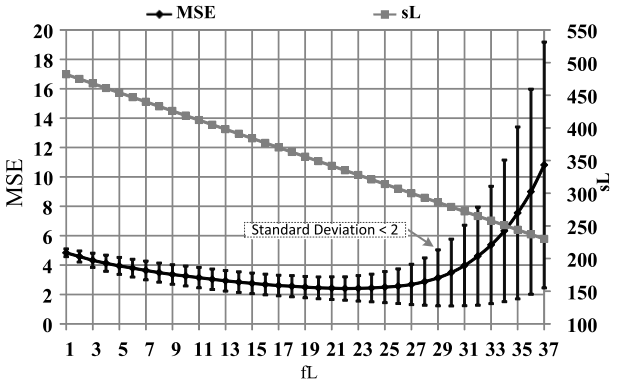


Fig. 14. MSE and sL versus fL for Step1 = 1 and Step2 = 8.

According to the studies in the past, most of digital standard cells are able to operate in the near-threshold region except the cells with high fan-in, which show functional failure or large delay variation. In this design, based on a standard cell library for nominal voltage operation, extensive simulations are performed to evaluate the performance of cells with high fan-in. The library is reconstructed by excluding the cells with fan-in >4. Following that, the reconstructed library is characterized at 0.5 V, which is a near-threshold voltage in the selected process technology to obtain timing information for synthesis and back-end design [25]. A near-threshold level shifter [26] is adopted in the design to interface between the core and I/O. For near-threshold operation, SRAM need to be redesigned as they have problems working at very low voltage. This usually causes large design effort [16]. In this design, however, as the FIFO size is significantly reduced, flipflops can be used to implement FIFO. Therefore, the additional design effort is saved. In addition, the flipflop is usually more robust than SRAM in ultralow voltage operation.

IV. Experimental Results

To demonstrate the proposed FIFO design, it has been implemented using a 0.18-μm CMOS process technology. The implementation covers different FIFO length, including 128, 256, 512, and 1024. For comparison, the FIFO using the conventional architecture, as shown in Fig. 2, is also implemented. Postsynthesis simulation results show that the proposed design is able to operate at 28.57 MHz at 0.5 V and 200 MHz at 1.8 V. To provide realistic input data for testing, the commonly used CDF 9/7 biorthogonal filter [27] is used to implement 2-D wavelet transform in MATLAB with the standard test images as its input. Then, the output generated by horizontal wavelet is used as input to FIFOs.

A. Parameters Selection and MSE Results

To determine the parameters fL, sL, Step1, and Step2, first L is assumed to be 128 in (5). In other words, the data input bit-width (B), which is 11 bits for the generated tested data set shown in Fig. 12, is compressed to 7-bit data before the input of FIFOs (b) in Fig. 2(a). Second, if the histogram of y_i [the output of differential predictor in Fig. 2(a)] is monitored, it is observed that $>68\%$ of difference values lie in the range of $(-74.6, 78.4)$ for given data sets. Fig. 13 shows the sL versus fL for various combinations of Step1 and Step2. Step1 $= 1$ better fits to the variation of $(-74.6, 78.4)$ because the maximum value of fL is equal to 61, while it is 122 for Step1 $= 2$ which is quite far from the range $(-74.6, 78.4)$.

In addition, because the average maximum value is $\sim$615 (refer to Fig. 12) Step2 is selected to be 8. Hence, maximum value of sL becomes 482 which is closer to 615 than that of Step2 when equals 16. The selected values for Step1 and Step2 in Fig. 13 are the factors of two to achieve the multiplierless design. Then, the average and standard deviation of MSE of data set along with sL versus fL are shown in Fig. 14. As shown in Fig. 14, standard deviation of MSE is less than two for all fL values equal to or <29, which means that sensitivity of MSE to various data sets is negligible. Thus, finally, fL and sL leading to standard deviation of MSE and MSE <4 are chosen as general accepted values [20]. We selected 25 and 314 for fL and sL, respectively, which in average leads to MSE of 2.76 (Figs. 14 and 15).

B. Area Reduction

Fig. 16 shows the area for conventional FIFO and the proposed FIFO. The percentage of reduction increases from 25.14% to 34.91% when the width is increased from 128 to 1024. The area reduction is mainly from the reduced

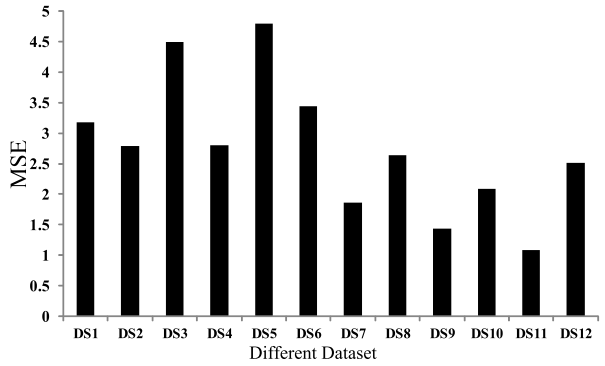


Fig. 15. FERDC-based FIFO with $L = 128$ (fL = 25, sL = 314. Step1 = 1 and Step2 = 8).

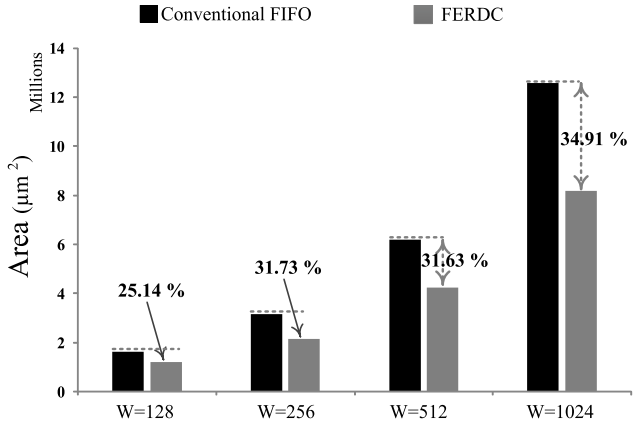


Fig. 16. Area for the conventional FIFO and proposed FIFO.

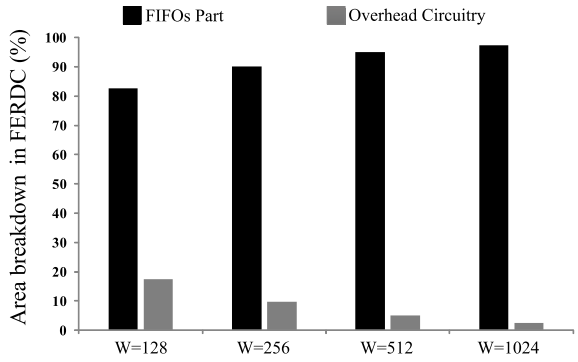


Fig. 17. Area breakdown for the proposed FIFO.

FIFO size due to proposed FERDC technique. Fig. 17 shows that the overhead circuitry (including differential predictor, update error, quantizer, IQ, and inverse differential predictor) varies between 17.37% ($W = 128$) and 2.6% ($W = 1024$) in the proposed FIFO, which implies that area overhead is very small when W is >256.

C. Power Consumption Reduction

To estimate the power consumption, the generated tested data are applied to the conventional and the proposed FIFO. The comparison of power consumption is shown in Fig. 18. It shows that the total power reduction is >99% between the

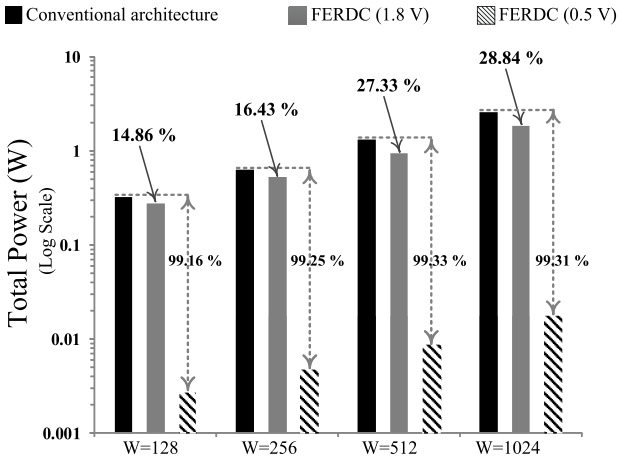


Fig. 18. Comparison between the total power of the conventional FIFO and proposed FIFO.

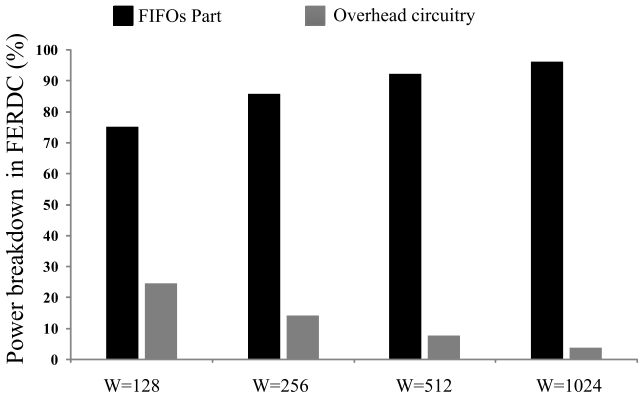


Fig. 19. Power breakdown of the proposed FIFO for different W.

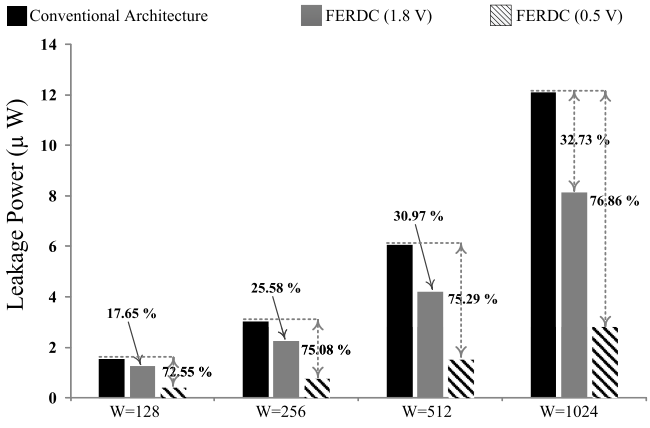


Fig. 20. Leakage power comparison of the conventional FIFO and proposed FIFO.

conventional FIFO and the proposed FIFO with both FERDC and near-threshold operation. Among them 14.86% ∼ 28.84% (for various W) is from the FERDC and 70.47% ∼ 84.3% is from near-threshold operation. Fig. 19 shows that the percentage of power consumption of the overhead circuitry varies from 24.65% ($W = 128$) to 3.79% ($W = 1024$) in the proposed FIFO. It is negligible for $W > 256$. The leakage

power reduction of the proposed FIFO with FERDC and near-threshold operation is >75% compared with conventional FIFO, as shown in Fig. 20. The FERDC contributes 17.65% to 32.73% of the leakage power reduction. The near-threshold operation exponentially reduces the leakage current, which contributes 44.13% $\sim$ 54.9% of the leakage power reduction.

V. Conclusion

In this paper, we have proposed an area and energy efficient FIFO design using error-reduced data compression and near-threshold operation for image/video applications. The proposed design is applied to an image filtering application, and achieves up to 99%, 65%, and 34.91% reduction in dynamic power, leakage power, and area, respectively, with a small MSE of 2.76, compared with conventional FIFO design. It can be used in hardware realization of image/video signal processing applications to reduce the area and power.

References

[1] W. K. Pratt, *Digital Image Processing: PIKS Scientific Inside*. Hoboken, NJ, USA: Wiley, 2007.

[2] C.-T. Huang, M. Tikekar, C. Juvekar, V. Sze, and A. Chandrakasan, "A 249Mpixel/s HEVC video-decoder chip for quad full HD applications," in *IEEE ISSCC Digest Technical Papers*, Feb. 2013, pp. 162–163.

[3] R. Rithe, P. Raina, N. Ickes, S. V. Tenneti, and A. P. Chandrakasan, "Reconfigurable processor for energy-scalable computational photography," in *IEEE ISSCC Digest Technical Papers*, Feb. 2013, pp. 164–165.

[4] C. Zhang, C. Wang, and M. O. Ahmad, "A pipeline VLSI architecture for fast computation of the 2-D discrete wavelet transform," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 8, pp. 1775–1785, Aug. 2012.

[5] Y. Hu and C. C. Jong, "A memory-efficient scalable architecture for lifting-based discrete wavelet transform," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 60, no. 8, pp. 502–506, Aug. 2013.

[6] S. M. A. Zeinolabedin, N. Karimi, and S. Samavi, "Low computational complexity hardware implementation of Laplacian pyramid," in *Proc. 18th Iranian Conf. Electr. Eng.*, May 2010, pp. 465–470.

[7] W. A. Atabany, B. McGovern, K. Mehran, R. Berlinguer-Palmini, and P. Degenaar, "A processing platform for optoelectronic/optogenetic retinal prosthesis," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 3, pp. 781–791, Mar. 2013.

[8] K. Danckaert, K. Masselos, F. Cathoor, H. J. de Man, and C. Goutis, "Strategy for power-efficient design of parallel systems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 7, no. 2, pp. 258–265, Jun. 1999.

[9] C.-H. Tsai, H. T. Wang, C.-L. Liu, Y. Li, and C. Y. Lee, "A 446.6 K-gates 0.55–1.2 V H.265/HEVC decoder for next generation video applications," in *Proc. IEEE Asian Solid-State Conf. (A-SSCC)*, Nov. 2013, pp. 305–308.

[10] Y. J. Chen *e al.*, "A 130.3 mW 16-core mobile GPU with power-aware approximation techniques," in *Proc. IEEE Asian Solid-State Conf. (A-SSCC)*, Nov. 2013, pp. 297–300.

[11] J.-W. Kim, J. Song, S. Lee, and I.-C. Park, "Tiled interleaving for multi-level 2-D discrete wavelet transform," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, New Orleans, LA, USA, May 2007, pp. 3984–3987.

[12] J. Song and I.-C. Park, "Novel pipelined DWT architecture for dual-line scan," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Taipei, Taiwan, May 2009, pp. 373–376.

[13] S.-L. Chen, "VLSI implementation of a low-cost high-quality image scaling processor," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 60, no. 1, pp. 31–35, Jan. 2013.

[14] B. K. Mohanty, P. K. Meher, S. Al-Maadeed, and A. Amira, "Memory footprint reduction for power-efficient realization of 2-D finite impulse response filters," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 1, pp. 120–133, Jan. 2014.

[15] D. Jeon, Y. Kim, I. Le, Z. Zhang, D. Blaauw, and D. Sylvester, "A 470 mV 2.7 mW feature extraction-accelerator for micro-autonomous vehicle navigation in 28 nm CMOS," in *IEEE ISSCC Digest Technical Papers*, Feb. 2013, pp. 166–167.

[16] T. T. Kim, J. Liu, J. Keane, and C. H. Kim, "Circuit techniques for ultra-low power subthreshold SRAMs," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Seattle, WA, USA, May 2008, pp. 2574–2577.

[17] P. Ou *et al.*, "A 65 nm 39GOPS/W 24-core processor with 11 Tb/s/W packet-controlled circuit-switched double-layer network-on-chip and heterogeneous execution array," in *IEEE ISSCC Digest Technical Papers*, Feb. 2013, pp. 56–57.

[18] S. M. A. Zeinolabedin, J. Zhou, X. Liu, and T. T. Kim, "An area- and power-efficient FIFO with error-reduced data compression for image/video processing," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Melbourne, Australia, Jun. 2014, pp. 2277–2280.

[19] D. G. Bailey, *Design for Embedded Image Processing on FPGAs*. Singapore: Wiley, 2011.

[20] D. Salomon, *Data Compression: The Complete Reference*, 4th ed. London, U.K.: Springer-Verlag, 2007.

[21] L. Chang *et al.*, "Practical strategies for power-efficient computing technologies," *Proc. IEEE*, vol. 98, no. 2, pp. 215–236, Feb. 2010.

[22] D. Jeon, M. Seok, Z. Zhang, D. Blaauw, and D. Sylvester, "Design methodology for voltage-overscaled ultra-low-power systems," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 59, no. 12, pp. 952–956, Dec. 2012.

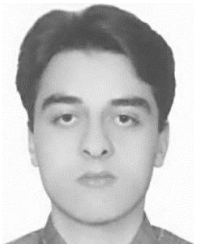
[23] A. Wang and A. Chandrakasan, "A 180-mV subthreshold FFT processor using a minimum energy design methodology," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 310–319, Jan. 2005.

[24] L. Chang and W. Haensch, "Near-threshold operation for power-efficient computing? It depends...," in *Proc. ACM/EDAC/IEEE Design Automation Conference*, Jun. 2012, pp. 1155–1159.

[25] X. Liu *et al.*, "Ultra-low-energy near-threshold biomedical signal processor for versatile wireless health monitoring," in *Proc. IEEE Asian Solid-State Conf. (A-SSCC)*, Nov. 2012, pp. 381–384.

[26] J. Zhou, C. Wang, X. Liu, X. Zhang, and M. Je, "A fast and energy-efficient level shifter with wide shifting range from sub-threshold up to I/O voltage," in *Proc. IEEE Asian Solid-State Conf. (A-SSCC)*, Nov. 2013, pp. 137–140.

[27] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Process.*, vol. 1, no. 2, pp. 205–220, Apr. 1992.



Seyed Mohammad Ali Zeinolabedin (S'12) was born in Isfahan, Iran. He received the B.S. degree in electrical engineering from Azad University, Tehran, Iran, in 2006 and the M.S. degree in electrical engineering from the Isfahan University of Technology, Isfahan, in 2010. He is currently pursuing the Ph.D. degree in electrical engineering with Nanyang Technological University, Singapore.

His current research interests include hardware implementation of image, video and biomedical algorithms, and ultralow-power circuits and systems design with high-energy efficiency.

Jun Zhou (M'07–SM'14) received the dual B.S. degree in communication engineering and microelectronics from the University of Electronic Science and Technology of China, Chengdu, China, in 2004 and the Ph.D. degree in microelectronics system design from Newcastle University, Newcastle upon Tyne, U.K., in 2008.

He joined as a Research Scientist with the Ultra-Low Power Digital Signal Processor Group, imec, Eindhoven, The Netherlands, in 2008, where he was involved in ultralow-power biomedical signal processor design for wearable healthcare applications in collaboration with Philips and NXP. In 2011, he joined the Institute of Microelectronics at the Agency for Science, Technology and Research, Singapore, where he leads projects and supervises Ph.D. students on energy-efficient digital signal processor design for wireless intelligent sensing applications, including healthcare, tire monitoring, and smart camera. He has authored over 30 papers in international prestigious conferences and journals, including the IEEE International Solid-State Circuits Conference, the IEEE JOURNAL OF SOLID-STATE CIRCUITS, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-PART I: REGULAR PAPERS, the Design Automation Conference, and the IEEE Asian Solid-State Circuits Conference, and holds five patents. His current research interests include energy-efficient digital signal processor design, ultralow-voltage and variation-resilient VLSI design, and 3-D integrated circuits design.

Xin Liu (M'12) received the B.Eng. degree in electrical engineering from Tianjin University, Tianjin, China, in 2000 and the Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore, in 2007.

He was a Research Fellow with the National University of Singapore, Singapore, from 2006 to 2007. Since 2007, he has been with the Institute of Microelectronics, Agency for Science, Technology and Research, Singapore. He is currently the Principal Investigator of the Digital-IC Group with the Integrated Circuits and Systems Laboratory. He leads the DIC Group's research activities in energy-efficient digital signal processor architecture, reliable ultralow-voltage digital circuit design, hardware-oriented algorithm development for on-chip signal processing, cognitive wireless radio baseband, advanced 2.5D/3D TSV/TSI design, and digital-assisted mixed-signal circuit design.

Dr. Liu has served on the Technical Program Committee of the IEEE International Wireless Symposium and the IEEE International Symposium on Independent Computing.

Tony Tae-Hyoung Kim (M'06–SM'14) received the B.S. and M.S. degrees in electrical engineering from Korea University, Seoul, Korea, in 1999 and 2001, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Minnesota, Minneapolis, MN, USA, in 2009.

He joined the Device Solution Network Division at Samsung Electronics, Yongin, Korea, in 2001. From 2001 to 2005, he performed research on the design of high-speed SRAM memories. In 2007 and 2008, he was with IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA, where he was involved in NBTI/PBTI-induced circuit reliability measurement circuits. In 2009, he was an Intern with Broadcom Corporation, Irvine, CA, USA, where he performed research on ultralow-power SRAM design. He joined the School of Electrical and Electronic Engineering at Nanyang Technological University, Singapore, in 2009, where he is currently an Assistant Professor. He has authored or co-authored over 70 journal and conference papers. His current research interests include ultralow-power and high-performance integrated circuits, including low-voltage circuits, silicon and non-silicon memories, and energy-efficient circuits and systems.

Prof. Kim was a recipient of the Best Paper Award at the IEEE International SoC Design Conference in 2011, the AMD/CICC Student Scholarship Award in 2008, the DAC/ISSCC Student Design Contest Award in 2008, the Departmental Research Fellowship from the University of Minnesota in 2008, the Samsung Humantec Thesis Award in 2008, 2001, and 1999, and the *ETRI Journal* Paper of the Year Award in 2005.