# An Area Efficient 1024-Point Low Power Radix-$2^2$ FFT Processor With Feed-Forward Multiple Delay Commutators

Ngoc Le Ba, *Student Member, IEEE*, and Tony Tae-Hyoung Kim, *Senior Member, IEEE*

***Abstract*—Radix-$2^k$ delay feed-back and radix-K delay commutator are the most well-known pipeline architecture for FFT design. This paper proposes a novel radix-$2^2$ multiple delay commutator architecture utilizing the advantages of the radix-$2^2$ algorithm, such as simple butterflies and less memory requirement. Therefore, it is more hardware efficient when implementing parallelism for higher throughput using multiple delay commutators or feed-forward data paths. Here, we propose an improved input scheduling algorithm based upon memory to eliminate energy required to shift data along the delay lines. A 1024-point FFT processor with two parallel data paths is implemented in 65-nm CMOS process technology. The FFT processor occupies an area of 3.6 mm$^2$, successfully operates in the supply voltage range from 0.4–1 V and the maximum clock frequency of 600 MHz. For low voltage, high performance applications, the processor is able to operate at 400 MHz and consumes 60.3 mW or 77.2 nJ/FFT generating 800 Msamples/s at 0.6 V supply.**

***Index Terms*—Fast Fourier transform (FFT), single delay feedback (SDF), multiple delay feedback (MDF), multiple delay commutator (MDC).**

## I. Introduction

THE Fourier transform (FT) is a technique to transform a signal from the time domain into the frequency domain to analyze the signal's frequency components. In the frequency domain, the absolute value at each frequency represents its contributing amount to the original signal while the complex value explains the phase offset of the sinusoid at that frequency. By decomposing a time domain signal into a summation of multiple sinusoid signals, FT offers a great conversion tool for processing a signal that is difficult or even impossible to handle in the time domain, which makes it one of the most important algorithms in digital signal processing. Since 1960, many research works have been carried out to calculate the Fourier transform at improved speed and contributed to the Fast Fourier transform (FFT). Various FFT algorithms have been reported such as Cooley-Tukey [1], Prime-factor [2], Bruun [3], and Hexagonal [4] FFTs. Cooley-Tukey is the most commonly used FFT algorithm, which recursively breaks a discrete time series into smaller transformations to reduce the number of operations in the FT calculation at each stage. With the development of the FFT algorithms, FFT processors have achieved much higher speed with lesser hardware cost. However, for real-time applications, high throughput FFT with low power consumption is still a challenging demand. To generate higher throughput, more processing paths can be implemented in parallel at the cost of additional hardware and power. In [5], the FFT processor used a multiple input, multiple output (MIMO) scheme and dynamic voltage and frequency scaling (DVFS). It could operate at 300MHz with the throughput of up to 2.4 Gsample/s while consumes 119.7 mW at 0.85 V supply. To reduce power consumption, super-pipeline FFT cores were proposed in [6] and [7] that demonstrated sub-threshold FFT designs using optimal energy points and optimal pipeline techniques. The latter design accomplished 240 Msample/s throughput at 0.27 V and consumed 3.7mW at 33 MHz, which is suitable for various ultra-low voltage applications. However, the FFT processor in [7] uses Radix-4 multiple delay commutator (MDC) architecture, which requires a larger memory size. Since the size of the memory is proportional to the level of parallelism, higher the level of parallelism will increase the hardware area and the power linearly. Therefore, more hardware and power efficient parallel architecture needs to be developed.

In this work, we present a novel FFT architecture called Radix-$2^2$ feed-forward multiple path delay commutator (R$2^2$FFMDC) to enhance the throughput of the previous MDC topologies and reduce energy consumption by utilizing Radix-$2^2$ algorithm. The proposed FFT processor achieves the maximum throughput of 1.2 Gsample/s at 600 MHz and 1 V supply. At 0.6 V, it generates a throughput of 800 Msamples/s and consumes only 60.3 mW (77.2 nJ/FFT), which is the best energy/conversion for high performance domain FFTs. The remainder of the paper consists of as follows. Section II reviews existing radix algorithms with implementations and Section III explains the proposed Radix-$2^2$ feed-forward (MDC) FFT accelerator for low power and high throughput. The proposed input scheduling algorithm for shift register elimination is described in Section IV. Section V presents experimental results followed by conclusions in Section VI.

Manuscript received November 4, 2017; revised April 10, 2018; accepted April 24, 2018. Date of publication May 10, 2018; date of current version August 30, 2018. This paper was recommended by Associate Editor R. Azarderakhsh. *(Corresponding author: Ngoc Le Ba.)*

The authors are with Nanyang Technological University, Singapore 639798 (e-mail: leba0002@e.ntu.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSI.2018.2831007

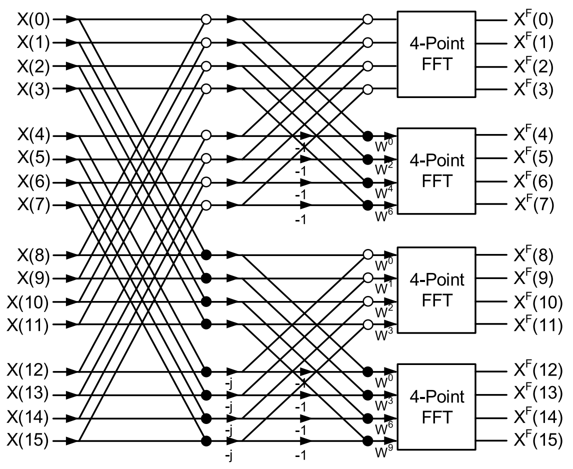


Fig. 1. Flow graph of 16-point Radix-4 FFT algorithm.

## II. Prior FFT Radix Algorithms and Implementations

### A. Radix-2, Radix-4 and Higher Radix Algorithms

In FFT, a N-point discrete time signal $x[n]$ can be represented in the frequency domain in the form of:

$$X[k]=\sum_{n=0}^{N-1}x[n].W_N^{kn},\quad k=0,1,\ldots,N-1 \tag{1}$$

where $W_N^{kn}=e^{-j\left(\frac{2\pi}{N}\right)nk}$ is called a twiddle factor.

Equation (1) can be further decomposed into equations (2 - 4) to have Radix-2 as follows:

$$X[k]=\sum_{n=0}^{\frac{N}{2}-1}\left[x[n]+(-1)^k x[n+\frac{N}{2}]\right].W_N^{kn}\quad k=0,1,\ldots,N-1 \tag{2}$$

Let k = 2m and k = 2m+1 (m = 0, 1,…, N/2-1), we have:

$$X[2m]=\sum_{n=0}^{\frac{N}{2}-1}\left[x[n]+x[n+\frac{N}{2}]\right].W_{N/2}^{mn} \tag{3}$$

which is a N/2 point FFT of $x[n]+x[n+\frac{N}{2}]$ and:

$$X[2m+1]=\sum_{n=0}^{\frac{N}{2}-1}\left[x[n]-x\left[n+\frac{N}{2}\right]\right].W_N^n.W_{\frac{N}{2}}^{mn} \tag{4}$$

which is a N/2 point FFT of $\left[x[n]+x\left[n+\frac{N}{2}\right]\right].W_N^n$, respectively. N-point FFT is obtained by iterating the procedure in (3) and (4) over ($\log_2$N – 1) times. This reduces the number of operations and hardware implementation cost (multipliers and adders). This methodology can be applied to realize Radix-4 or higher radix algorithms by decomposing $X[k]$ into 4 or more sub-FFT [8]. Equation (5) shows the decomposition of the Radix-4 FFT algorithm and Fig. 1 explains the flow graph of it. The black dots represent complex subtractions and the white dots represent complex additions. With Radix-4 or higher radix FFT, the number of multipliers and adders further decreases at the cost of increased complexity in control circuits to re-arrange data into each sub-FFT.

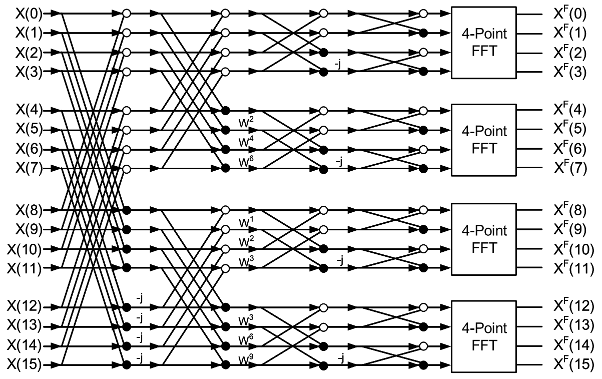


Fig. 2. Flow graph of 16-point radix-$2^2$ FFT algorithm.

$$X[k]=\sum_{n=0}^{\frac{N}{4}-1}\left[x[n]+(-j)^k x\left[n+\frac{N}{4}\right]+(-1)^k x\left[n+\frac{N}{2}\right]\right.$$
$$\left.+(-j)^k x\left[n+\frac{3N}{4}\right]\right].W_N^{kn}\quad k=0,1,\ldots,N-1 \tag{5}$$

### B. Radix-$2^2$ Algorithm

The Radix-$2^2$ algorithm was developed to inherit the simple control structure of Radix-2 but adopt operation and hardware saving technique from Radix-4. This makes Radix-$2^2$ well suited for VLSI implementation of low power FFT. In [9], by considering the first two decompositions of Radix-2 together, equation (1) can be written in another form as in equation (6)

$$X[k_1+2k_2+4k_3]=\sum_{n_3=0}^{\frac{N}{4}-1}\left[H(k_1,k_2,k_3)\right].W_{\frac{N}{4}}^{n_3k_3} \tag{6}$$

where $H(k_1,k_2,k_3)=[A+(-j)^{(k_1+2k_2)}B].W_N^{n_3(k_1+2k_2)}$

$$A=x[n_3]+(-1)^{k_1}x\left[n_3+\frac{N}{2}\right]$$
$$B=x\left[n_3+\frac{N}{4}\right]+(-1)^{k_1}x\left[n_3+\frac{3N}{4}\right]$$

Here, A and B are two conventional Radix-2 FFT and H($k_1$, $k_2$, $k_2$) is also a Radix-2 FFT of A and B with an additional complex multiplier for the twiddle factor $W_N^{n_3(k_1+2k_2)}$. Using this decomposition, the first two stages of Radix-2 can be transferred into one stage Radix-$2^2$ consisting of two cascaded Radix-2 FFT. The flow graph of this algorithm is depicted in Fig. 2.

Radix-$2^2$ has a simple architecture of Radix-2 and has the same number of iterated stages ($\log_4$N-1) as in Radix-4. In terms of calculating operations and hardware implementation, Radix-$2^2$ has less complex architecture with a smaller number of multipliers. Therefore, it is very popular for pipeline FFT implementation to reduce the power consumption.

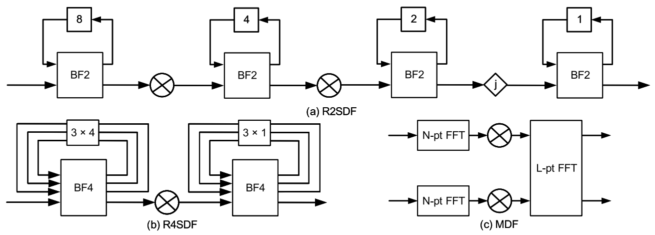


Fig. 3. Delay feedback architecture: (a) R2SDF, (b) R4SDF, (c) MDF.

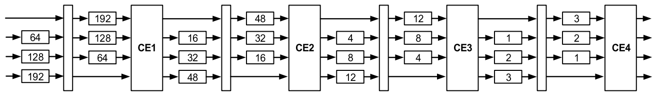


Fig. 4. Radix-4 multiple delay commutator.

TABLE I
COMPARISON OF VARIOUS FFT ARCHITECTURES

| Architecture | Multiplier | Adder | Memory | Samples/clk |
|---|---|---|---|---|
| R2MDF [12] | 2(log$_4$N-1) | 4log$_4$N | N - 1 | 2 |
| R2$^2$MDF [15] | 2(log$_4$N-1) | 8log$_4$N | N - 1 | 2 |
| R2MDC [13] | 2(log$_4$N-1) | 2log$_4$N | 3N/2 - 2 | 2 |
| R4MDC [16] | (log$_4$N-1) | 2log$_4$N | 5N/2 - 4 | 4 |

## C. High Throughput FFT Architecture

*1) Delay Feedback:* It can be seen from equations (2) - (6) that each FFT stage consists of an adding stage or butterfly (BF) and a twiddle multiplying stage. The inputs of each BF need to be separated by a correct distance (N/2 for the 1st stage, N/4 for 2nd stage …), which can be realized by a delay path. Two well-known architectures to implement this requirement are delay feedback and delay commutator. In the delay feedback scheme, outputs from BF are delayed by a predetermined distance to pair with the next coming inputs and then fed back to the BF for next operation cycles. Shift registers store the data in the feedback path as presented in the Single Delay Feedback (SDF) topology [10]. To increase the FFT's throughput, more delay paths can be used as in Multiple Delay Feedback (MDF) [11]. However, it increases the hardware complexity with additional parallel data paths. Fig. 3(a) and 3(b) describe the architecture of a Radix-2 SDF (R2SDF) and Radix-4 SDF (R4SDF) while Fig 3(c) depicts the MDF FFT architecture. When employing these architectures, 2×(log$_4$N-1) multipliers are required for R2SDF and R2MDF while only (log$_4$N-1) multipliers are necessary for R4SDF. The number of adders for R2SDF and R4SDF are 4×log$_4$N and 8×log$_4$N, respectively. Clearly, R4SDF is preferred for hardware implementation since fewer multipliers are involved [12].

*2) Delay Commutator:* Instead of feeding back BF's output, the output can be arranged immediately at the commutator using different delay paths with a correct delay distance. Fig. 4 shows a Radix-4 multiple delay commutator (MDC) FFT architecture. Because of the parallel data paths, the throughput of the MDC architecture is higher than that of the delay feedback architecture. However, the MDC architecture shows a lower multiplier utilization rate and

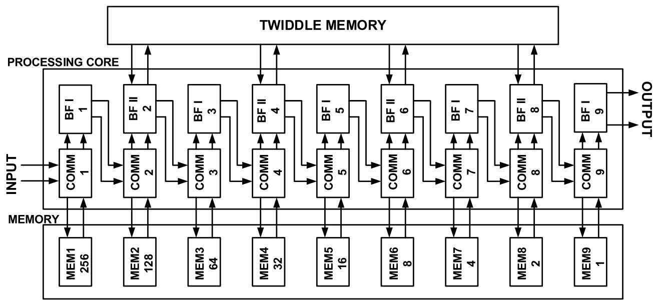


Fig. 5. Proposed 1024-point Radix-2$^2$ feedforward FFT architecture with two parallel data paths.

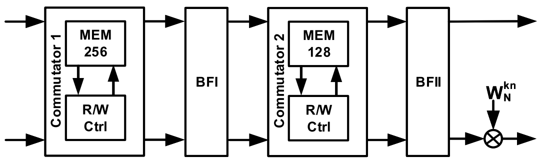
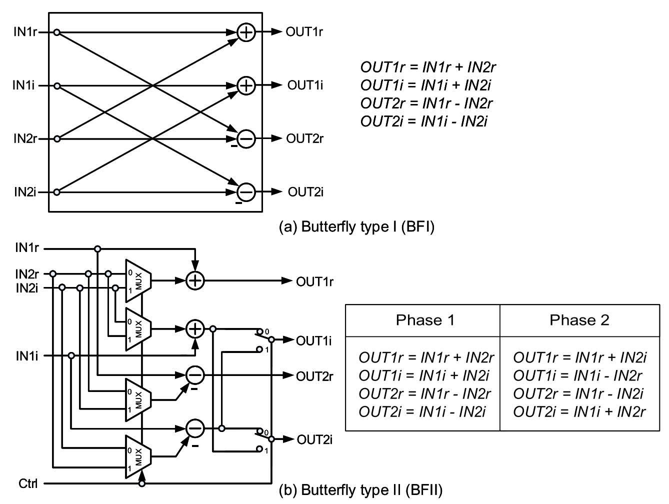


Fig. 6. Structure of a radix-2$^2$ stage which is a combination of 2 single stages, the 1st stage uses a conventional butterfly (BFI) while the 2nd stage uses a modified butterfly (BFII).

requires a larger memory size [13]. Comparisons in Table I between different topologies conclude that MDC architecture requires lesser number of adders and a larger memory size, but provides higher throughput than the MDF architecture. In terms of algorithms, the Radix-2$^2$ helps to reduce memory size however, using Radix-2$^2$ with delay feedback reduces the throughput while using with MDF provides higher throughput but at the cost of hardware overhead. Recently, a few research works have investigated Radix-2$^2$ MDC (R2$^2$MDC) for minimizing the hardware cost without throughput degradation except in [14] where an FPGA implementation for R2$^k$MDC was performed using traditional input scheduling mechanism with shift registers to prove the advantages of the R2$^k$MDC architecture over existing parallel FFT topologies. In this paper, we propose a novel input scheduling algorithm that can eliminate shift registers in the Radix-2$^2$ feed-forward (MDC) FFT architecture. Since the shift registers occupy a large circuit area and consume significant dynamic energy during data shifting along the delay lines, the proposed scheme can provide lower energy. In addition, delay commutators and address generators are optimized to work with a register-based memory and reduce memory access power.

## III. PROPOSED RADIX-2$^2$ FEED-FORWARD FFT PROCESSOR

Fig. 5 shows our proposed architecture of a 1024 point Radix-2$^2$ MDC FFT processor with two parallel data paths and nine processing stages. Here, the odd numbered processing stages execute the BF operations for calculating A and B in equation (6) while the even numbered stages compute H in equation (6). As shown in Fig. 6, the odd-numbered BF and the even-numbered BF are called Butterfly Type I (BFI) and Butterfly Type II (BFII), respectively. BFI is just a conventional Radix-2 BF while BFII is a slightly modified Radix-2



| Phase 1 | Phase 2 |
|---|---|
| OUT1r = IN1r + IN2r | OUT1r = IN1r + IN2i |
| OUT1i = IN1i + IN2i | OUT1i = IN1i - IN2r |
| OUT2r = IN1r - IN2r | OUT2r = IN1r - IN2i |
| OUT2i = IN1i - IN2i | OUT2i = IN1i + IN2r |

Fig. 7. Butterfly implementation. Note that BFI is a trivial Radix-2 butterfly while BFII requires complex multipliers and modifications in data path.

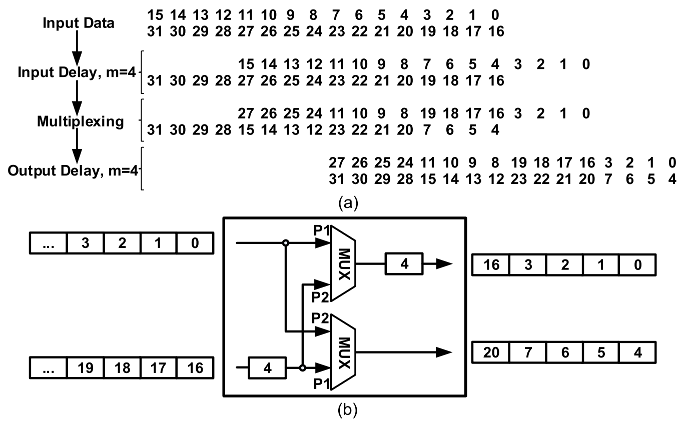


Fig. 8. A conventional input scheduling approach. Note that this approach consumes high energy for shifting data along the delay lines.

BF for realizing H in equation (6) with complex number multipliers. To implement BFI for two complex number inputs, only two adders and two subtractors are necessary as illustrated in Fig. 7 (a). For BFII, the H-expression in equation (6) is broken down into two phases whose outputs are shown in Fig. 7(b). BFII requires an additional control signal (Ctrl) and multiplexers for switching data between two phases. This control signal is generated from the most significant bit (MSB) of an n-bit counter where n is ($1 + \log_2 d$) and d is the distance required between two BF's inputs. For example, when the input distance in the 2nd stage is 128, an 8-bit counter is used for generating Ctrl. During FFT operations, Phase I remains for 128 clock cycles then the counter toggles its MSB to switch to Phase II for another 128 clock cycles. This counter also operates as an address generator to fetch twiddle factors from a pre-programmed register-based memory. Since twiddle factors periodically appear at the butterfly's output, they are hard-coded in ROM memories with address equals to the clock sequence so that in each cycle correct twiddle factors are fetched to the multiplier. In this design, we employ Baugh-Wooley multiplier [17] to multiply BFII's output with twiddle factors because it is popular for its regularity in array multiplication and high speed operation.

In term of hardware cost, the proposed design has $\log_2 N$ stages, a half of which use BFI and the other half use BFII (Fig. 5 and Fig. 6). BFI contains 2 adders and 2 subtractors while BFII has one more multipliers (Fig. 7). Therefore, the proposed esign requires total $4\times\log_4 N$ adders as with R2MDF, ($\log_4 N$-1) multipliers as in R4MDC. Regarding the memory size, the proposed Radix-$2^2$ FFT algorithm improves the minimum memory requirement and only needs a total memory size of N-1 bits. In comparison with other existing topologies, the proposed Radix-$2^2$MDC requires the smallest memory size and least multipliers. The number of adders is not at the smallest due to the inheriting Radix-2 BF structure. However, adders do not consume as much power and area as multipliers and memories. Assuming 1024-point FFT, the proposed design saves 50% of the multipliers from MDF structures and 30% of memory from R2MDC or 60% of memory from R4MDC. Besides, the proposed architecture has a continuous flow of data, which can provide 2 samples/cycle throughput with 2-parallel paths. It is worth to notice that by increasing the number of parallelism, higher throughput is achievable without increasing memory size [14]. Consequently, our proposed FFT architecture is more hardware efficient and scalable for higher throughput when comparing with the MDF and R4MDC structures.

## IV. Proposed Input Scheduling Algorithm

In the Radix-$2^2$ equation, the input to BFs must be separated by correct distances, which is done by input schedulers in the commutator. The most straightforward and well-known scheduling approach is using delay lines with switching networks as shown in Fig. 8. In this example, two incoming data are distanced by 16 and the output has the distance of 4. An input delay line using 4 shift registers is placed at the lower input to postpone the input data by 4 cycles. During Phase I (P1), the delayed input enters the lower MUX while the upper input enters the upper MUX for 4 cycles. After 4 cycles, MUXs switch to Phase II (P2) where the upper MUX takes data from the lower input and the lower MUX output data from the upper input. Together with the data coming out from the upper output delay line, the lower output is separated by a distance of 4. Fig. 8(a) demonstrates a data flow example using this input scheduling method. The implementation of this input scheduler requires only FIFOs to create the delay lines and MUXs with a control signal to realize the switching network. However, all data are shifted inside the delay network every clock cycle and these activities consume significant dynamic energy. This is particularly true when the system operates at high frequency for high throughput. Therefore, we propose a novel input scheduling algorithm that eliminates the delay lines and data shifting to reduce the overall switching activities and the layout area.

In our proposed input scheduler, a memory replaces the function of the conventional delay lines for low power and high density [6]. Even though static random access memory (SRAM) can be a good candidate, the requirement for high

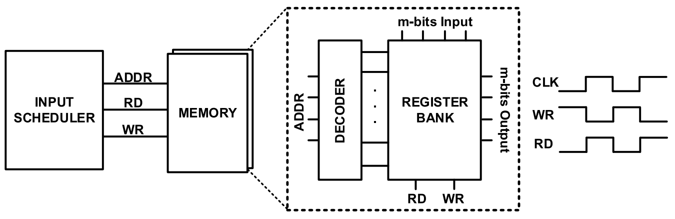


Fig. 9. Register-based memory design. Note that read and write operations are performed in the same clock cycle.

| | | CLK | In1 | In2 | Mem1 | Mem2 | Out1 | Out2 |
|---|---|---|---|---|---|---|---|---|
| Phase I | Out1 reads from Mem2 | 0 | 0 | 16 | 0 | 16 | ~ | ~ |
| | In2 writes to Mem2 | 1 | 1 | 17 | 1 | 17 | ~ | ~ |
| | Out2 reads from Mem1 | 2 | 2 | 18 | 2 | 18 | ~ | ~ |
| | In1 wrtie to Mem1 | 3 | 3 | 19 | 3 | 19 | ~ | ~ |
| Phase II | Out1 reads from Mem1 | 4 | 4 | 20 | 20 | 16 | 0 | 4 |
| | In2 write to Mem1 | 5 | 5 | 21 | 21 | 17 | 1 | 5 |
| | Out2 reads from In1 | 6 | 6 | 22 | 22 | 18 | 2 | 6 |
| | | 7 | 7 | 23 | 23 | 19 | 3 | 7 |
| Phase I | Out1 reads from Mem2 | 8 | 8 | 24 | 8 | 24 | 16 | 20 |
| | In2 writes to Mem2 | 9 | 9 | 25 | 9 | 25 | 17 | 21 |
| | Out2 reads from Mem1 | 10 | 10 | 26 | 10 | 26 | 18 | 22 |
| | In1 wrtie to Mem1 | 11 | 11 | 27 | 11 | 27 | 19 | 23 |
| Phase II | Out1 reads from Mem1 | 12 | 12 | 28 | 28 | 24 | 8 | 12 |
| | In2 write to Mem1 | 13 | 13 | 29 | 29 | 25 | 9 | 13 |
| | Out2 reads from In1 | 14 | 14 | 30 | 30 | 26 | 10 | 14 |
| | | 15 | 15 | 31 | 31 | 27 | 11 | 15 |
| | ... | . | . | . | . | . | . | . |

Fig. 10. Data flow in our proposed input scheduling algorithm. Note that there is no shifting data and data is accessed directly by its address.

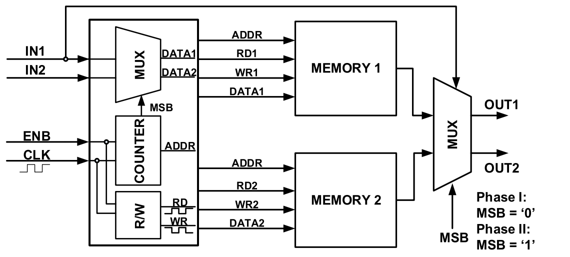


Fig. 11. Hardware implementation of the proposed input scheduler.

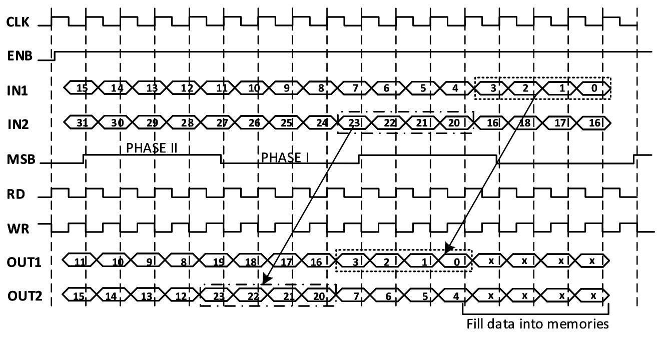


Fig. 12. Timing diagram of the input scheduler circuit. Note that its takes nfirst clocks to fill data into the memory with size n and after that a continous flow is maintained.

operating frequency (up to 500 MHz) posts a challenge in low power SRAM design. Therefore, this work employs a register-based memory and SRAM implementation shall be left for a future work. Fig. 9 shows our memory design, consisting of a register bank and a decoder to convert input address into a memory location for accessing the memory's cell. To reduce the latency and increase the speed, register memory design is chosen with read and write operations are performed within one clock cycle. The input data is transparent and held to the output when the clock is high (Phase I). When the clock is low (Phase II), write signal (WR) is raised to insert new data into the memory location. Read and write signals are two non-overlapping signals that are generated together with the address by the algorithm. In each input scheduler, there are two equal size memories which are equivalent to two delay lines from the conventional implementation. However, there is no shifting data along the register line but only direct memory access which reduces dynamic energy significantly.

The data flow in Fig. 10 explains the mechanism of our proposed input scheduling algorithm with two operational phases (Phase I and II). Phase I starts as follows. During the rst half of a clock, data are read from both memories in an interchanging manner, which means that the 2nd output (Out2) is read from the 1st memory while the 1st output (Out1) is read from the 2nd memory. In the second half of a clock, data from input stream 1 (In1) is written into the first memory while input stream 2 (In2) goes to the second memory. This procedure continues for n-clock cycles (n = 4 in this example) if a distance of n is required at the output. After n clock cycles, Phase II starts by reading data from the 1st memory for Out1. Out2 directly uses input from In1. Reading operations are achieved while the clock is high, and the writing operations are performed when the clock is low. In this phase, only data from In2 is written into the 1st memory. Phase II operations are carried out for n cycles for completion of reading data from the 1st memory then the whole process is repeated from the beginning of Phase I. By interchanging between two phases, two n-block memories are switched between the two inputs, creating a required n-distance data at the output. In comparison with the conventional input scheduler, the proposed algorithm only changes the content of one memory cell in each clock cycle since it is able to access every single cell by its address. Whereas the conventional scheduler changes the content of all FIFOs in the delay lines in every clock cycle and consumes significant switching power, especially when long delay lines are required at the beginning stages of FFT. Therefore, our proposed algorithm improves the energy used for data accessing which contributes substantially to the overall FFT energy consumption.

Fig. 11 and 12 illustrate the hardware implementation and the timing diagram of the proposed input scheduler. As described in Fig. 10, an n-block data is processed sequentially in each phase. Therefore, their addresses in the memory can also be arranged in a continuous sequence. This pattern simplifies the way of accessing the memory by using a simple counter to generate the address of memory cells. For the output distance of n, a '$1 + \log_2 n$' bit counter is required and the MSB indicates the phase order (i.e. Phase I when MSB = 0 and Phase II when MSB = 1) while the other bits are the data addresses. Two MUXs (one at the input and the other at the output) switch data as required by the proposed scheduling algorithm and their states are determined by the MSB signal. The read/write (R/W) circuit is just a buffer circuit to generate non-overlapping read (RD) and write (WR) signals from the clock (CLK) and the enable signal (ENB). As shown

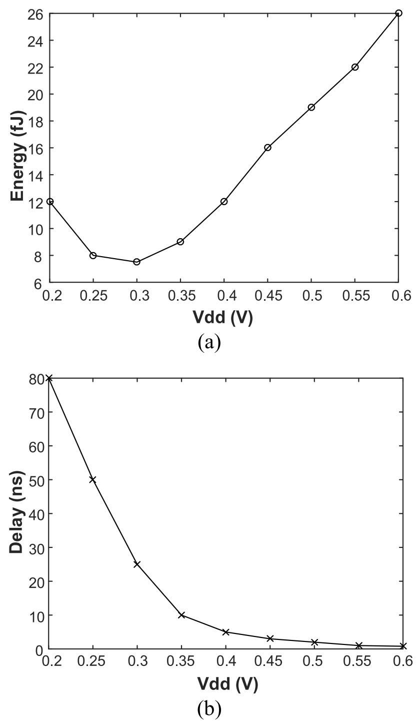


Fig. 13. Optimum operating point of the devices. Note that $V_{DD} = 0.4$ V is chosen as the operating voltage to achieve minimum energy point and $V_{DD} = 0.6$ V is chosen as the best circuit performance operating point.

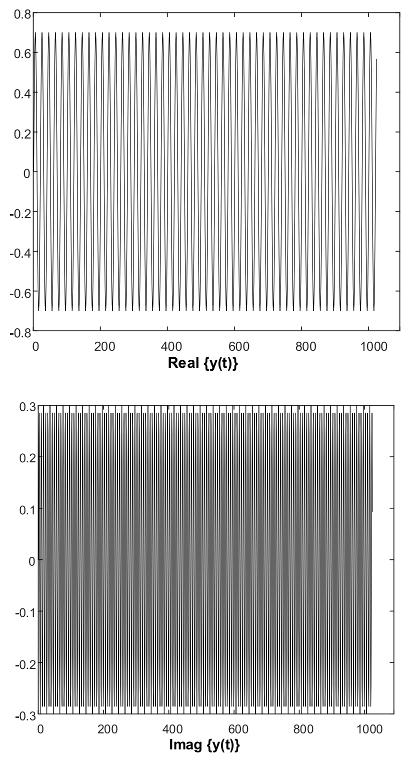


Fig. 14. A 1024 point discrete input signal used for testing: *y(t) = 0.7×sin(100π t) + j×0.3×sin(300π t).*

in Fig. 12, initially the two memories are empty and it takes n-clock cycles to fill data into the memories whose size is equal to the required distance, n. Once the memories are filled, the RD signal reads out data during the first half of the clock and the WR signal overwrites the data in the same location with new data obtained from the input. After completing the process for the first data block, the next data block is already available. Therefore, a continuous data flow inside the FFT is achieved without interruption in the output data stream. The proposed scheduler generates two data samples per clock cycle. In order to achieve higher throughput, a higher degree of parallelism can be constructed using the same FFT architecture and the input scheduling algorithm.

## V. Experimental Results

The proposed 1024-point feed-forward radix-$2^2$ FFT core with two parallel data paths is designed using 65nm low power CMOS standard cells and the FFT processor occupies an area of 3.6 $mm^2$ (2.0 × 1.8 mm). To obtain the optimum supply voltage, a chain of 64 inverters is used to simulate with HSPICE [18]–[20] for investigating the relationship between supply voltage versus energy consumption as well as versus circuit delay. Different activity factors are also considered in this simulation. The simulation results are shown in Fig. 13(a) which explains the relationship between the energy versus supply voltage and in Fig. 13(b) which explains the circuit delay as a function of supply voltage. Note that the minimum energy point is formed at 0.3 V. However, $V_{th}$for this process is about 370 mV, below which leakage energy will dominate dynamic energy due to the exponentially increased circuit delay as described in Fig. 13(b). After considering the energy and the delay curves, $V_{DD} = 0.4$ V and $V_{DD} = 0.6$ V are chosen as targeted operating points of the proposed FFT processor for optimum energy and optimum circuit performance.

The proposed FFT processor is designed in Verilog with ModelSim to verify its functionality and the required circuit blocks are synthesized for fabrication. The layout netlist is simulated again in ModelSim using 1 V standard cell library showing reliable FFT operations with the maximum clock frequency of up to 800 MHz. To verify the functionality and measure the FFT performance, a complex signal in Fig. 14, $y(t) = 0.7 \times \sin(100\pi t) + j \times 0.3 \times \sin(300\pi t)$ is digitized into 16 bit input data sequence. The input data sequence is then programmed into a logic analyzer for pattern generation. The output of the logic analyzer is converted to 0.4 V, 0.6 V and 1 V for testing the FFT at different supply voltage levels. Finally, the truncated outputs of the FFT processor are collected and compared with the output from Matlab FFT of y(t) for accuracy assessment. Bit length truncation and finite bit representation in FFT causes quantization error and impacts accuracy, which is quantified by Signal-to-Quantization Noise Ratio (SQNR) [21].

Fig. 15 shows the FFT spectrum of y(t) signal obtained from Matlab FFT function (Fig. 15(a)) and actual measurements of the proposed FFT (Fig. 15(c)). Since the output bit-width increases in each stage and it is not possible to accommodate all the lengths due to the substantial hardware cost, a truncation step is required. From simulations, using uniform bit-width from the first stage degrades the accuracy from the theoretical value (Fig. 15(a)) by 30 dB as in Fig. 15(b).

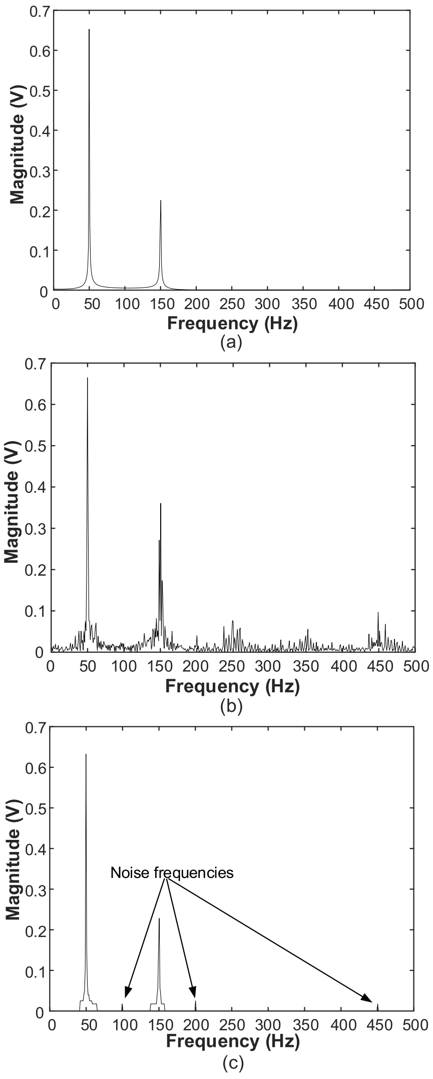


Fig. 15. FFT spectrum of the signal $y(t) = 0.7 \times sin(100\pi t) + j' \times 0.3' \times sin(300\pi t)$. (a) Matlab FFT spectrum. (b) FFT spectrum using uniform bit width (c) Actual FFT spectrum from measurement results.

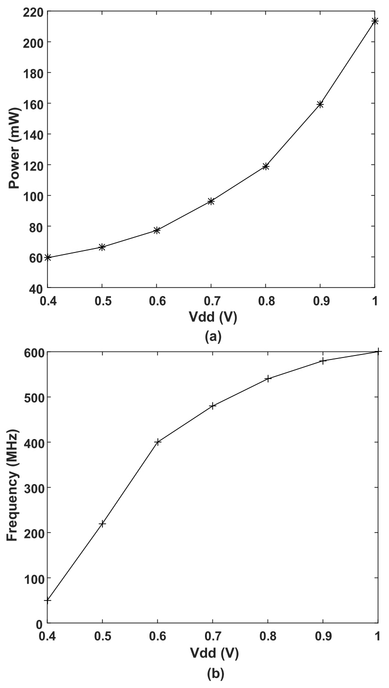


Fig. 16. Measurement results for (a) energy consumption (b) clock speed performance.

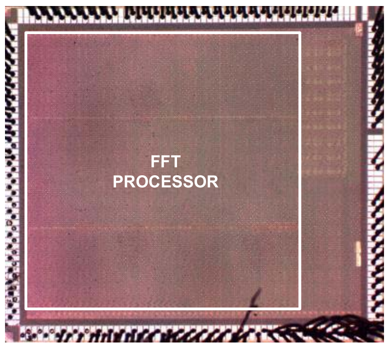


Fig. 17. Micrograph of the proposed FFT processor (area = 3.6 mm$^2$).

In [22], it shows that the output noise is added mainly by the truncation of fractional parts from the initial stages and the final stage. Therefore, to have a better trade-off between hardware cost and SNDR performance, the first few stages, and the final stage should not truncate fractional part of the data. In our design, the data word-width is allowed to grow up to 40 bits from the 1st stage to the 4th stage. Truncations are only applied from the 5th stage and at the final stage there is no multiplication hence no truncation for fractional data is required. Since this is a non-uniform word-length design, multipliers in each stage have different size. For example, stage 2 uses the smallest 16 bit multipliers and stage 8 requires the largest 32-bit multiplier (truncations are applied from stage 5-8). From layout simulations, multipliers contribute to 8.1% of the total chip area and 5.6% of the total power. By employing this strategy, the proposed FFT output spectrum achieves $-1$ dB difference from the ideal value. Fig. 15 (c) shows measured FFT output spectrum of the proposed FFT processor. Two signal frequencies are formed at 50 Hz and 150 Hz together with some noisy frequency components that are noticeable at 100 Hz, 200 Hz and 450 Hz. These components contribute to the quantization noise and deteriorate the SQNR by 1 dB. Consequently, our proposed FFT processor achieves much higher accuracy compared with the uniform data width designs.

To measure energy consumption, the FFT is supplied at different voltages from 0.4 V to 1 V while sweeping the clock frequency from 50 MHz to 600 MHz. Fig. 16 summaries the measured FFT energy consumption and operating frequency versus voltage supply. At the target operating voltage of 0.4 V and 50 MHz, the FFT consumes 5.8 mW and takes 1024/($10^8$) seconds to complete one FFT calculation with 100 Msamples/s, therefore, energy per FFT is 59.3 nJ/FFT. At 0.6 V with maximum clock frequency of 400 MHz, the proposed FFT consumes 60.3 mW or energy per FFT of 77.2 nJ/FFT for 800 Msample/s throughput. Using normalized energy conversion formula [23] in which normalized

TABLE II

HARDWARE COMPARISON TABLE BETWEEN HIGH THROUGHPUT FFT TOPOLOGIES

| Spec | This work | $R2^2$SDF | $R2^2$MDF | R4MDC |
|---|---|---|---|---|
| # of Adders | $4\times\log_4N$ | $4\times\log_4N$ | $4\times\log_4N$ | $2\times\log_4N$ |
| # of Multipliers | $\log_4N$-1 | $\log_4N$-1 | $\log_4N$-1 | $\log_4N$-1 |
| Memory size | N-1 | N-1 | N-1 | 5N/2-4 |
| Control | Simple | Simple | Simple | Complex |

TABLE III

COMPARISON TABLE BETWEEN THIS WORK AND OTHER STATE-OF-THE-ART FFT DESIGNS

| Specifications | This work | [24] | [5] | [7] |
|---|---|---|---|---|
| FFT length (N) | 1024 | 1024 | 256 | 1024 |
| Technology | 65 nm | 65 nm | 90 nm | 65 nm |
| Word-length | 32 bit | 16 bit | 10 bit | 16 bit |
| Topology | $R2^2$ FF | Mix Radix MDF | $R2^4$ MDF | Modified R4 MDC |
| Supply Voltage | 0.4~1.0 V | 0.5~1.0 V | 0.625~1.0 V | 0.27~1.0 V |
| Die Area | 2.0×1.8 $mm^2$ | 2.39×1.40 $mm^2$ | 2.26×2.26 $mm^2$ | 2.71×3.15 $mm^2$ |
| # of Adders | $4\times\log_4N$ | $4\times\log_4N$ | $4\times\log_4N$ | $8\times\log_4N$ |
| # of Multipliers | $\log_4N$ - 1 | $\log_4N$ - 1 | $\log_4N$ - 1 | $3\times(\log_4N-1)$ |
| Memory size | 3N/2 – 2 | N – 1 | 5N/2 - 8 | 7N/2 - 4 |
| Power Consumption | 60.3 mW @ 0.6 V | 13.83 mW @ 0.45 V | 16.4 mW @ 0.625 V | 82 mW @ 0.6 V |
| Max Clock Frequency | 600 MHz | 20 MHz | 500 MHz | 290 MHz |
| Normalize Energy for high throughput | 77.2 nJ @ 0.6 V, 800 Msample/s | 89.5 nJ @ 0.43 V, 80 Msample/s | 167.9 nJ @ 0.625 V, 2.4 Gsample/s | 84.0 nJ @ 0.6 V, 2.4 Gsample/s |

energy is directly proportional to the execution time and power consumption but inversely proportional to the data path width and technology node, different FFT designs are comparable across FFT length, word width and technology node. Compared to the state-of-the-art FFT design [7] which requires 84.0 nJ normalized energy for high performance of 0.6 V supply and 290 MHz clock frequency, our proposed processor has better energy efficiency for mid-high range performance while it has slightly higher energy consumption in ultra-low voltage domain (below 0.4 V). he maximum throughput of 1.2 Gsamples/s is achieved at 1 V and 600 MHz clock frequency. In this mode, the proposed processor consumes the energy of 213.3 nJ/FFT. The proposed FFT processor only occupies 3.6 $mm^2$ in comparison with 8.5 $mm^2$ in [7] and 5.1 $mm^2$ in [5]. This improvement in the layout area comes from the proposed Radix-$2^2$ algorithm requiring the smallest memory size and MDC for minimizing the number of multipliers. The micrograph of the FFT core is shown in Fig. 17. Table II is the hardware cost comparison of different FFT topologies and Table III shows the performance comparison of our design with other benchmarks in ultra-low voltage and high performance FFT processors using the normalizing formulas introduced in [23].

## VI. CONCLUSIONS

This paper present the first SOC Radix-$2^2$ MDC FFT architecture for low power, high throughput applications. Combining benefits of both Radix-$2^2$algorithm and feed-forward architecture, the proposed processor achieves the lowest hardware requirements for multipliers and memory size in compared with delay feedback architecture and Radix-K MDC. A 1024-point Radix-$2^2$ MDC FFT design is fabricated in STM 65 nm process, occupying and area of 3.6 $mm^2$ (2.0 × 1.8 mm) which is the smallest area among recently reported FFTs with the same technology node and FFT length. Using register-based memory structure and SRAM oriented input scheduling algorithm, delay lines with shift registers are eliminated to avoid generating redundant power for shifting data along the delay line. As a result, the power consumption of our design is comparable with the state-of-art FFT processors in ultra-low voltage domain while achieves better energy saving for high performance domain. At 0.6 V supply, our processor is able to operate at 400 MHz clock frequency, generates 60.3 mW power for 800 Msample/s output. The maximum throughput of 1.2

Gsamples/s is achieved at 1 V nominal supply using 213.3 nJ/FFT. With our proposed input scheduling algorithm, SRAM design can be used to replace the current register memory and more parallel levels can improve the throughput at no additional memory cost. These improvements which should be discussed in our future work would bring larger energy saving together with higher throughput.

## REFERENCES

[1] M. Heideman, D. Johnson, and C. Burrus, "Gauss and the history of the fast fourier transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 1, no. 4, pp. 14–21, 1984.

[2] L. X. Jiang, C. Y. Liu, and P. Zhang, "A novel overall in-place in-order prime factor FFT algorithm," in *Proc. 5th Int. Congr. Image Signal Process. (CISP)*, Chongqing, China, 2012, pp. 1500–1503.

[3] S. Mittal, Z. A. Khan, and M. B. Srinivas, "Area efficient high speed architecture of Bruun's FFT for software defined radio," in *Proc. IEEE Global Telecommun. Conf.*, Nov. 2007, pp. 3118–3122.

[4] J. B. Birdsong and N. I. Rummelt, "The hexagonal fast fourier transform," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 1809–1812.

[5] Y. Chen, Y. W. Lin, Y. C. Tsao, and C. Y. Lee, "A 2.4-Gsample/s DVFS FFT processor for MIMO OFDM communication systems," *IEEE J. Solid-State Circuits*, vol. 43, no. 5, pp. 1260–1273, May 2008.

[6] 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.

[7] D. Jeon, M. Seok, C. Chakrabarti, D. Blaauw, and D. Sylvester, "A super-pipelined energy efficient subthreshold 240 MS/s FFT core in 65 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 47, no. 1, pp. 23–34, Jan. 2012.

[8] K. R. Rao, D. N. Kim, and J. J. Hwang, *Fast Fourier Transform: Algorithms and Applications*. Amsterdam, The Netherlands: Springer, 2010.

[9] S. He and M. Torkelson, "A new approach to pipeline FFT processor," in *IEEE Proc. IPPS*, Apr. 1996, pp. 766–770.

[10] G. Bi and G. Li, "Pipelined structure based on Radix-$2^2$ FFT algorithm," in *Proc. IEEE Conf. Ind. Electron. Appl.*, Jun. 2011, pp. 2530–2533.

[11] J.-F. Tang, X.-J. Li, G. Zhang, and Z.-S. Lai, "Design of high-throughput mixed-Radix MDF FFT processor for IEEE 802.11.3c," in *Proc. Int. Conf. Solid-State Integr. Circuit Technol.*, 2013, pp. 1–3.

[12] V. Stojanovic. *Fast Fourier Transform: VLSI Architectures, Course Materials for Comminication System Design*. MIT OpenCourseWave. Accessed: 2006. [Online]. Available: http://ocw.mit.edu

[13] G. Bi and E. V. Jones, "A pipelined FFT processor for word-sequential data," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 12, pp. 1982–1985, Dec. 1989.

[14] M. Garrido, J. Grajal, M. A. Sánchez, and O. Gustafsson, "Pipelined Radix-$2^k$ feedforward FFT architectures," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 1, pp. 23–32, 2013.

[15] N. Li and N. P. van der Meijs, "A Radix $2^2$ based parallel pipeline FFT processor for MB-OFDM UWB system," in *Proc. IEEE Int. SOC Conf.*, Sep. 2009, pp. 383–386.

[16] E. E. Swartzlander, W. K. W. Young, and S. J. Joseph, "A Radix 4 delay commutator for fast Fourier transform processor implementation," *IEEE J. Solid-State Circuits*, vol. SSC-19, no. 5, pp. 702–709, Oct. 1984.

[17] C. R. Baugh and B. A. Wooley, "A Two's complement parallel array multiplication algorithm," *IEEE Trans. Comput.*, vol. C-22, no. 12, pp. 1045–1047, Dec. 1973.

[18] R. Gonzalez, B. M. Gordon, and M. A. Horowitz, "Supply and threshold voltage scaling for low power CMOS," *IEEE J. Solid-State Circuits*, vol. 32, no. 8, pp. 1210–1216, Aug. 1997.

[19] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "The limit of dynamic voltage scaling and insomniac dynamic voltage scaling," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 11, pp. 1239–1252, Nov. 2005.

[20] S. Hanson *et al.*, "Ultralow-voltage, minimum-energy CMOS," *IBM J. Res. Develop.*, vol. 50, nos. 4–5, pp. 469–490, Sep. 2006.

[21] D. James, "Quantization errors in the fast Fourier transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-23, no. 3, pp. 277–283, Jun. 1975.

[22] P. Gupta, "Accurate performance analysis of a fixed point FFT," in *Proc. 22nd Nat. Conf. Commun. (NCC)*, 2016, pp. 1–6.

[23] B. M. Baas, "A low-power, high-performance, 1024-point FFT processor," *IEEE J. Solid-State Circuits*, vol. 34, no. 3, pp. 380–387, Mar. 1999.

[24] C.-H. Yang, T.-H. Yu, and D. Marković, "A 5.8 mW 3GPP-LTE compliant 8×8 MIMO sphere decoder chip with soft-outputs," in *Proc. IEEE Symp. VLSI Circuits (VLSIC)*, Jun. 2010, pp. 209–210.

**Ngoc Le Ba** (S'08) received the B.S. degree in electrical and electronics engineering from Nanyang Technological University, Singapore, in 2008, where he is currently pursuing the Ph.D. degree. He joined Hewlett Packard Pte Ltd., Singapore, in 2008, where he was a Test Development Engineer. Since 2012, he was a Project Officer with Nanyang Technological University. His research interests include ultra-low-power circuits, digital processors, and low-power and high-performance VLSI systems.

**Tony Tae-Hyoung Kim** (M'06–SM'14) received the B.S. and M.S. degrees in electrical engineering from Korea University, Seoul, South Korea, in 1999 and 2001, respectively. He received the Ph.D. degree in electrical and computer engineering from the University of Minnesota, Minneapolis, MN, USA, in 2009. From 2001 to 2005, he was with Samsung Electronics where he performed research on the design of high-speed SRAM memories, clock generators, and IO interface circuits. In 2007 and 2009, he was with the IBM T. J. Watson Research Center and Broadcom Corporation, where he performed research on circuit reliability, low power SRAM, and battery backed memory design, respectively. In 2009, he joined Nanyang Technological University, where he is currently an Associate Professor.

He received the Best Demo Award at APCCAS2016, the Low Power Design Contest Award at ISLPED2016, the best paper awards at 2014 and 2011 ISOCC, the AMD/CICC Student Scholarship Award at the IEEE CICC2008, the Departmental Research Fellowship from University of Minnesota in 2008, the DAC/ISSCC Student Design Contest Award in 2008, the Samsung Humantec Thesis Award in 2008, 2001, and 1999, and the ETRI Journal Paper of the Year Award in 2005.

He is an author/co-author of over 130 journal and conference papers and has 17 U.S. and Korean patents registered. His current research interests include low power and high performance digital, mixed-mode, and memory circuit design, ultra-low voltage circuits and systems design for IoT and Biomedical applications, variation and aging tolerant circuits and systems, and circuit techniques for 3-D ICs. He has served numerous conferences as a committee member. He is the Chair of the IEEE Solid-State Circuits Society Singapore Chapter. He serves as an Associate Editor for the IEEE TRANSACTIONS ON VLSI SYSTEMS.