Energy-efficient synchronous-logic and asynchronous-logic FFT/IFFT processors

Chong, Kwen-Siong; Gwee, Bah Hwee; Chang, Joseph Sylvester

2007

Chong, K. S., Gwee, B. H. & Chang, J. S. (2007). Energy-efficient synchronous-logic and asynchronous-logic FFT/IFFT processors. IEEE journal of solid-state circuits, 42(9), 2034-1045.

<https://hdl.handle.net/10356/90560>

<https://doi.org/10.1109/JSSC.2007.903039>

IEEE Journal of Solid-State Circuits © 2007 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE. This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder.
<http://www.ieee.org/portal/site>.

Energy-Efficient Synchronous-Logic and Asynchronous-Logic FFT/IFFT Processors

Kwen-Siong Chong, *Student Member, IEEE*, Bah-Hwee Gwee, *Senior Member, IEEE*, and Joseph S. Chang

Abstract—Two 128-point 16-bit radix-2 FFT/IFFT processors based on synchronous-logic (sync) and asynchronous-logic (async) for low voltage (1.1–1.4 V) energy-critical low-speed hearing aids are described. The two processors herein are designed with the same function and similar architecture, and the emphasis is energy efficacy. The async approach, on average, features $\sim 37\%$ lower energy per FFT/IFFT computation than the sync approach but with $\sim 10\%$ larger IC area penalty and an inconsequential 1.4 times worse delay; the async design can be designed to be 0.24 times faster and with largely the same energy dissipation if the matched delay elements and the latch controllers therein are better optimized. In this low-speed application, the lower energy feature of the async design is not attributed to the absence of the clock infrastructure but instead due to the adoption of established and proposed async circuit designs, resulting in reduced redundant operations and reduced spurious/glitch switching, and to the use of latches. The prototype async FFT/IFFT processor (in a $0.35\text{-}\mu\text{m}$ CMOS process) can be operated at 1.0 V and dissipates 93 nJ.

Index Terms—Asynchronous-logic, fast Fourier transform (FFT), hearing aids, inverse FFT (IFFT), low energy, synchronous-logic.

I. INTRODUCTION

HEARING AIDS (hearing instruments) are high-efficacy assistive biomedical devices to improve the speech intelligibility of hearing impaired users. The primary challenges in these subminiature devices are often power dissipation related [1], [2] due to the need to realize a number of advanced signal processing algorithms, including a filterbank [3], noise reduction, feedback cancellation and amplification (usually Class-D amplifiers [4]). Physically, the limitations are largely due to the low-voltage (1.1–1.4 V) low-energy capacity ($\sim 100\text{ mA h}$) miniature-size battery whose life-span is expected to be ~ 100 hours or more, and due to aesthetics of hearing instruments.

There is considerable research in low power (and energy) design techniques [5]–[7] for digital circuits based on the synchronous-logic (sync) approach. The sync approach is prevalent arguably due to the maturity of its design methodology, design simplicity and the availability of many commercial EDA design

tools. The asynchronous-logic (async) approach [8]–[12], on the other hand, is an alternative with potential power advantage largely due to the absence of a global clock infrastructure (required in the sync approach), reduced redundant operations and reduced glitches/spurious switching. Nonetheless, only a small number of async designs [8]–[12] have been demonstrated to date, and the adoption of async design remains stymied, to a large part, by the nascent tools available.

In this paper, the design of an energy-efficient async 128-point 16-bit Fast Fourier Transform/Inverse Fast Fourier Transform (FFT/IFFT) processor based on the async approach for energy-critical applications, specifically for hearing instruments, is demonstrated. The async design is benchmarked against its sync counterpart in terms of energy, delay, and IC area. Both designs are realized with the same functionality and similar architecture, and fabricated using the same $0.35\text{-}\mu\text{m}$ CMOS process. In the case of the sync design, the well established clock gating approach is adopted (idle state) to reduce energy dissipation when computation is not required. It will be shown that the async design, on average, dissipates $\sim 37\%$ lower energy per FFT/IFFT computation than the sync design. However, the minor drawbacks of the async design are the required larger IC area, by 10% , and an inconsequential 1.4 times average worse delay, inconsequential because of the low-speed hearing instrument application. If the delay of matched delay elements is more aggressively tuned and faster latch controllers (Broad to Broadish) are selected, the async design can feature a delay 0.24 times less than the sync design, and with largely the same energy dissipation. The lower energy attribute is achieved largely by means of the async approach embodying established and several proposed async circuit designs, including: 1) reduced redundant (spurious) operations; 2) energy-efficient async datapath circuits (e.g., multiplier, memory, etc.); and 3) the use of simple latches (as opposed to flip-flops in sync designs). This design is yet another demonstration of how the async approach may be a good alternative to the prevalent sync approach.

This paper is organized as follows. Section II presents an overview of the FFT/IFFT algorithm that provides a preamble to the architecture adopted for the sync and async approaches. Section III presents a comparison of the sync and async designs. Finally, Section IV concludes the paper.

II. FFT/IFFT PROCESSOR DESIGNS

In this section, an overview of the FFT/Discrete Fourier Transform (DFT) algorithm is briefly described followed by a description of the sync and async FFT/IFFT processor designs.

Manuscript received April 26, 2006; revised March 12, 2007. This work was supported in part by the ASEAN-EU University Network Programme under Grant ASE/B7-301/1997/0178-15. The work of K.-S. Chong was supported by the Singapore Millennium Foundation.

K.-S. Chong is with the Integrated Systems Research Laboratory, School of Electrical and Electronic Engineering, Nanyang Technological University, 639798 Singapore (e-mail: kschong@ntu.edu.sg).

B.-H. Gwee and J. S. Chang are with the School of Electrical and Electronic Engineering, Nanyang Technological University, 639798 Singapore (e-mail: ebhgwee@ntu.edu.sg; ejschang@ntu.edu.sg).

Digital Object Identifier 10.1109/JSSC.2007.903039

Fig. 1. Data flows for each butterfly operation: (a) to compute the real outputs, and (b) to compute the imaginary outputs.

TABLE I
SYNC (BENCHMARK) AND ASYNC FFT/IFFT PROCESSORS

	Sync Design	Async Design
Algorithm	128-point Radix-2 Decimation-in-time	
Wordlength	16-bit	
Computation Effort	7×64 butterfly operations	
Process Technology	0.35 μ m dual-poly four-metal CMOS process	
Datapath Circuits	2 shifters, 2 multipliers, 3 adders	
Memory	1 memory bank (128 $\times$ 32-bit)	2 memory banks (128 $\times$ 16-bit each)
Cell Library	Standard cells	Standard cells + async cells

Given a sequence of time samples $x(n)$ or a sequence of frequency samples $X(k)$, an N -point DFT and its inverse are, respectively,

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot W_N^{nk} \quad k = 0, 1, \dots, N-1 \quad (1)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \cdot W_N^{-nk} \quad n = 0, 1, \dots, N-1 \quad (2)$$

where the twiddle factor $W_N^{nk} = e^{-j2nk\pi/N} = \cos(2nk\pi/N) - j \sin(2nk\pi/N)$

The FFT is essentially derived from the DFT and the former reduces the computational complexity to $O(\log_r N)$ (as compared to $O(N^2)$ in the latter) where r is the radix number.

In this paper, both the sync and async FFT/IFFT processors [13] are based on a 128-point radix-2 decimation-in-time FFT algorithm. The prevalent radix-2 is adopted for its operational regularity and for its hardware simplicity. The inverse (IFFT) is obtained simply by negating the sine coefficients and multiplying the outputs by a scaling factor of $1/128$ and using the same FFT algorithm. Equations (3a)–(3d) describe the operations for one radix-2 butterfly:

$$R'_a = R_a + [(C) \cdot R_b - (-S) \cdot I_b] \quad (3a)$$

$$R'_b = R_a - [(C) \cdot R_b - (-S) \cdot I_b] \quad (3b)$$

$$I'_a = I_a + [(C) \cdot I_b + (-S) \cdot R_b] \quad (3c)$$

$$I'_b = I_a - [(C) \cdot I_b + (-S) \cdot R_b] \quad (3d)$$

where $R_a + jI_a$ and $R_b + jI_b$ are the two inputs, and $R'_a + jI'_a$ and $R'_b + jI'_b$ are the corresponding two outputs. C and S are the cosine and sine coefficients of the twiddle factor.

To compute (3a)–(3d), two multipliers and three adders perform one butterfly in two data flows as depicted in Fig. 1. Data flows in Fig. 1(a) and (b), respectively, compute the real and the imaginary outputs. Each data flow requires five pipeline stages: Memory Read, Scaling, Multiplications, Additions/Subtractions, and Write Back. In Memory Read, the signals are retrieved from the memory and in Scaling, the signals are scaled (where necessary) to accommodate the signal levels and hence avoid overflow. In Multiplications, two multiplications are performed and in Additions/Subtractions, three additions/subtractions are performed. Finally, in Write Back, the outputs are stored.

Table I summarizes the sync and async FFT/IFFT processors. The sync design is based on standard digital cells and a 128 $\times$ 32-bit memory obtained from the foundry. The async design is based on several fully and partially handcrafted async cells (including two blocks of async 128 $\times$ 16-bit memories, two async multipliers, etc.) and, where pertinent, on the same cells used in the sync design. The basic philosophy here is an attempt to make the comparison of the sync design and the async design in Section III as fair as possible.

The block diagram of sync FFT/IFFT processor is depicted in Fig. 2 where two clock signals, Clk1 and Clk2, are used. The frequency of Clk2 is $2 \times$ Clk1 and Clk2 is used to trigger the memory to obtain two sets of 32-bit memory data for

Fig. 2. Block diagram of the sync FFT/IFFT processor.

every Clk1 cycle (one butterfly operation). Clk1 is used as the system clock for the remaining modules. The sync FFT/IFFT processor requires 2430 Clk2 cycles and 959 Clk1 cycles for one FFT/IFFT computation (including loading new samples and with result outputs). The well-established coarse-grain and fine-grain clock gating approaches were considered and as both approaches yield somewhat similar energy savings, the former is adopted for its simplicity. Specifically, in the coarse-grain clock gating approach, Clk2 is gated after 2430 Clk2 cycles and Clk1 gated after 959 Clk1 cycles. For a typical hearing instrument where Clk1 is 1 MHz and for a time duration of 4 ms (64 samples @ 16 kHz) for one FFT/IFFT computation, the coarse-grain clock gating blocks $\sim 76\%$ (3041 of 4000) and $\sim 70\%$ (5570 of 8000) of Clk1 and Clk2 respectively; note that the datapath modules in the sync design circuits are highly pipelined, thereby resulting in useful computation in most of the active clock cycles. If fine-grain clock gating is otherwise employed, only small percentage of clock cycles (i.e., a further 63 out of 4000 Clk1 cycles and a further 126 out of 8000

Clk2) may be further gated, and collectively this translates to a small 6% improvement in energy dissipation. However, this small improvement comes at a cost including a design of higher complexity (including consideration for synchronization issues, skew problems and race conditions) and this would largely defeat the small energy advantage.

The async FFT/IFFT processor is based on the 4-phase protocol (instead of the 2-phase protocol [8] for the advantages already established in literature) and is partitioned into the control portion (Async FFT/IFFT Controller and Pulse Circuit, see Fig. 3) and the datapath portion (Memory, Shifter, Multiplier, Adder, and Write Back Datapaths, see Fig. 4). The former portion pipelines the entire operation asynchronously within the processor and the latter portion executes the butterfly operations.

In Fig. 3, the Main Sequence Controller, Sample Loading Controller, Data Loading Controller, Butterfly Unit Operation Controller, Write Back Controller, and Block Floating Point Controller are collectively grouped as the Async FFT/IFFT Controller. This Async FFT/IFFT Controller controls the

Fig. 3. Block diagram for the control portion in the async FFT/IFFT processor.

Fig. 4. Block diagram of the datapath portion in the async FFT/IFFT processor.

Fig. 5. Simplified signal transition graph for one butterfly operation.

respective datapath circuits according to the control and handshake signals. The Pulse Circuit functions as a local “clock” to synchronize the sequences of each operation in the Async FFT/IFFT Controller and the local “clock” is generated asynchronously.

Fig. 4 depicts five datapath modules and their associated handshake and control signals. The Memory Datapath reads and writes data during the FFT/IFFT operations. The Shifter Datapath is for conditional block floating point scaling to retain the dynamic range of the signals, avoiding overflow—the signals therein are scaled when overflow is detected in the Write Back Datapath. The Multiplier Datapath is used to multiply the coefficients (from ROMs) with the signals from the Shifter Datapath. The Adder Datapath performs the additions and subtractions, and finally the Write Back Datapath stores and writes the outputs back into the Memory Datapath. For sake of illustration, only the main modules are shown in each datapath. These datapaths are controlled by their respective handshake control circuits (HCCs).

Fig. 5 depicts the simplified signal transition graph for one butterfly operation in the async FFT/IFFT processor, with the initialization of *Req_Butterfly*. The solid lines indicate the direct interface relationship between the signals and the dashed lines indicate the indirect interface relationship between the signals (e.g., via some circuits that depend on other control signals). Two data flows (*xxx_Flow1* and *xxx_Flow2*, where *xxx* indicates the prefix labels for various signals) compute the multiplications and additions. After a butterfly operation is completed,

Fig. 6. Control-multiplier.

Comp_Butterfly will be triggered. In total, there are 7×64 butterfly operations.

The following subsections describe several (fully and partially handcrafted) async cells for the realization of the async FFT/IFFT processor.

Fig. 7. Memory: (a) Block diagram, (b) row decoder and word line controller, and (c) column decoder.

A. Multiplier

In the computation of the 128-point FFT/IFFT algorithm, approximately 42% of the multiplications are trivial (i.e., $\times 1$, $\times -1$, and $\times 0$). The computation of trivial multiplications can be simplified either by passing the inputs directly to the outputs (for $\times 1$), by setting the outputs to be zeros (for $\times 0$), or by negating the inputs (for $\times -1$). In these instances, the

effective energy dissipation due to the multiplier can be reduced by having the appropriate control circuits output these trivial multiplication products accordingly. Fig. 6 depicts the proposed async 16×16 -bit multiplier with the added control circuits (comprising a control logic, a converting circuit, and multiplexers) and this multiplier is named the Control-Multiplier in Fig. 6. This multiplier design is, in part, based on our reported design [14] where the spurious switching/glitch

Fig. 8. Latch controllers: (a) Broadish, (b) Broad I, and (c) proposed Broad II.

therein is largely eliminated by means of Latch Adders (LAs) timed by delay lines, D_1 to D_8 .

The inputs are the multiplicand X (input signals) and the multiplier Y (coefficients from the ROMs). In Fig. 6, the shaded inverting latches block these inputs to the multiplier core in cases of trivial multiplications. For nontrivial multiplications, the shaded inverting latches will be updated with new data to the multiplier core for multiplications. The control circuits determine if the multiplications are trivial or otherwise, initiate multiplications when REQ is asserted, and generate $COMP$ upon the completion of a multiplication.

Based on simulations with 40% (estimated percent of the actual multiplications) of the multiplications being trivial, the Control-Multiplier dissipates 22% lower energy dissipation per multiplication compared to a multiplier [14] without control circuits. The tradeoff is a manageable 7% larger IC area for the Control-Multiplier.

For completeness, note that for fair comparison in Section III later, the multipliers in the sync FFT/IFFT processor also feature a somewhat similar function—the multipliers therein for trivial multiplications are disabled by operand isolation (with control signals to the input registers of the multipliers).

B. Memory

Fig. 7(a) depicts the block diagram of the async 128×16 -bit single-port memory macrocell. For low-energy dissipation, the

well-established partitioning approach [6] is adopted to reduce the capacitance. In this approach, the memory is subdivided into four 32×16 -bit sub-blocks (Blocks A to D) and these sub-blocks are controlled by their respective Word Line Controllers. The Row Decoder, Column Decoder, and Control Circuitry control the memory for the write and read accesses. For the read access, the stored data are retrieved (via the Transfer Gate from one of the four sub-memory blocks) to the output buffers, according to the address bus, $A[6:0]$. For the write access, the new input is written (from the input buffers) into one of the four sub-memory blocks, according to $A[6:0]$. The REQ and $COMP$ signals respectively indicate the request and completion signals of the memory.

The standard 6-transistor single-port SRAM is employed as the basic memory storage. For low-energy reasons [6], the two-stage decoding technique in a NOR-NAND structure is adopted to construct the Row Decoder [see left-hand side in Fig. 7(b)] according to the address bus $A[6:2]$. The Word Line Controllers [see right-hand side in Fig. 7(b)] activate the selected sub-memory block and prevent multiple assertions of unselected sub-memory blocks, resulting in reduced energy dissipation. The Word Line Controller is controlled by the Column Decoder [see Fig. 7(c)] which selects the specific column of the memory according to the address bus $A[1:0]$. The Memory Access (MA) signal is a buffered signal from REQ and the memory will only be selected when REQ (and MA) is asserted.

Fig. 9 (a) Inverting latch, (b) non-inverting latch, (c) pulse generator, and (d) delay line.

Pearl scripts are developed to floorplan the four sub-blocks. The Word Line Controllers are placed adjacent to their respective sub-memory blocks to keep the interconnections short. The entire async 128×16-bit memory cell occupies an IC area of 410 $\mu\text{m} \times 346.4 \mu\text{m}$.

C. Async Handshake Circuits

Three types of 4-phase async latch controllers are employed and are depicted in Fig. 8 where the top figures are the schematic diagrams and the lower figures are their corresponding signal transition graphs. The first controller [15] in Fig. 8(a) is in Broadish data valid [8] when the En signal is ready to be asserted after R_{out} is low ("−"). Similarly, the second latch controller [15] in Fig. 8(b) is in Broad data valid [8] when the En signal is asserted after A_{out} is low ("−"). A potential problem with these two latch controllers is that R_{out} can be asserted even though En has not been reset to low. This results in the possibility of an incomplete data transfer and the

subsequent macrocells may undesirably perform redundant operations when their inputs are not ready. To circumvent this potential problem, a different Broad controller is proposed and this is depicted in Fig. 8(c). In this case, R_{out} will only be asserted when En is low.

One assumption made in the operation of these three latch controllers is that the predetermined delay pulse (from $En-$ to $En+$ to $En-$) has to be sufficiently long to allow the complete data transfer to the latches.

Fig. 9 depicts four other handcrafted circuits used widely in the async FFT/IFFT processor. Fig. 9(a) and (b) respectively depict an inverting latch [16] and a non-inverting latch [17], and these latches are more energy-efficient than the usual flip-flops in sync designs. Fig. 9(c) depicts a pulse generator circuit where the generated pulsewidth is determined by the delay lines [Fig. 9(d)]. The self-reset property (low to high to low) of the pulse generator circuit makes it very easy to control the async controllers. Fig. 9(d) depicts the delay line where the transistors

Fig. 10. A modelling timing diagram (from Verilog simulator) for the Async FFT/IFFT Controller—pertinent signals only are shown and the delays herein do not reflect the actual circuit delays.

connected to V_{DD} and ground are sized to obtain the desired amount of delay with little penalty in energy dissipation [14].

D. Other Modules

The Shifter Datapath employs conventional static CMOS multiplexers. In a radix-2 FFT algorithm, the maximum overflow that can occur per butterfly operation is 2 bits. Hence, the shifters only need to be implemented using simple 3-to-1 multiplexers. The Adder Datapath comprises three carry-completion sensing adders [18] where only the carry blocks are implemented using dual-rail logic for completion detection. The Write Back Datapath primarily comprises latches and the Block Floating Point circuit for temporary storage of the intermediate outputs and for the overflow detection respectively.

The Async FFT/IFFT Controller is modeled using Verilog and its interfaces are modified for async operations. The simple linear pipeline for the Controller is adopted as it does not require high-speed operation in view of the intended hearing instrument application. In other words, although the processing within one butterfly is operated at different pipeline stages and at different rates, the butterflies are in fact realized sequentially. Fig. 10 depicts a simplified modelling timing diagram captured from the Verilog simulator for the Async FFT/IFFT Controller.

III. COMPARISON OF THE SYNC AND ASYNC DESIGNS

Fig. 11(a) and (b), respectively, depict the microphotographs of the prototype sync and async FFT/IFFT processor ICs based on a $0.35\text{-}\mu\text{m}$ CMOS process with $|V_{Tp}| = 0.69\text{ V}$ and $V_{Tn} = 0.5\text{ V}$. The async FFT/IFFT processor occupies 1.6 mm^2 IC area, 10% larger than its sync counterpart. Table II tabulates the operating conditions for both the sync and async processors.

Fig. 12 depicts the energy dissipation from measurements on prototype ICs for the async and sync designs. The async design features lower energy dissipation where on average from 1.1 V to 1.4 V (typical voltage range for hearing instruments), the async design dissipates $\sim 37\%$ lower energy per FFT/IFFT

Fig. 11. Microphotographs of the FFT/IFFT processors: (a) Sync design and (b) async design.

computation than the sync design and this is attributed to three reasons.

First, in the sync design, despite the coarse-grain clock gating described earlier, a small number of datapath circuits (e.g., during the initialization phase where signals are not ready for the particular pipeline stage) remain asserted by the clock signal(s), resulting in $\sim 6\%$ redundant operations in

TABLE II
OPERATING CONDITIONS FOR THE SYNC AND ASYNC FFT/IFFT PROCESSORS

	Condition
Voltage	1.1V to 1.4V
Computation Time	4ms (64 samples @ 16KHz sampling frequency)
Operation	128 new samples loaded + one FFT/IFFT transform + 128 outputs

Fig. 12. Energy dissipation for the async and sync FFT/IFFT processors.

Fig. 13. Delays for the async and sync FFT/IFFT processors.

some pipeline stages. To design the sync circuits to eliminate the 6% redundant operations by means of fine-grain clock gating would require a more complex sync controller with multi-phasic clocks and complex logic circuits, hence largely defeating the 6% saving and this would likely incur further costs due to the associated overheads. This redundant switching does not occur in the async design. Further, the 96 flip-flops that serve as delay registers to pipeline the data sequence to the datapath circuits in the sync design are unnecessary in the async design. Overall, this first reason accounts for an overall reduced energy of $\sim 9\%$.

Second, the spurious switchings in the async datapath circuits are reduced by synchronizing the arrival time of the inputs to the custom-designed datapath circuits (e.g., memories, multipliers, adders, etc.) with better control by means of the latches and delay lines, by gating the redundant hazards and transitions with async handshake circuits, and by disabling the unused blocks (particularly for latches whose data are not updated for computations). For example, our async multiplier [14] is $\sim 30\%$ lower energy than the sync multiplier (with operand isolation). In the sync design, the inputs of the signals can also be synchronized at added cost and increased design complexity, and this is applicable only to the circuits where input flip-flops are present. Overall, this second reason resulting in reduced switchings accounts for $\sim 18\%$ energy saving.

Third, a latch is typically dissipated 50% lower energy than a flip-flop and by comparison, there are a total of 783 flip-flops in the sync design against the 122 flip-flops and 459 latches in the async design. The sync design can also be designed based on a less prevalent latch-based approach, a somewhat unconventional approach that is more complex including the need to consider the signal synchronization more carefully. Overall, this third reason accounts for $\sim 10\%$ energy saving.

It is worthwhile to note that the clocking energy in the sync design is in fact small, $\sim 5\%$ of the total energy. This is largely because of the low clock rate, the FFT/IFFT architecture and the

simplicity of a simple buffer network for the clock infrastructure. In other words, in this specific low clock frequency example, the absence of a global clock infrastructure in the async realization does not contribute significantly to the energy savings. The energy dissipation due to the clock infrastructure is likely to be more significant if the clock rate is high and/or if the IC is large.

Fig. 13 depicts the measured delays for the async and sync designs, where the delay is defined as the minimum time for one complete FFT/IFFT computation. By means of increasing the clock frequency until the sync design fails, the delay of the async design is deemed to be ~ 1.4 times worse. The modality of increasing the clock frequency in the sync design is somewhat equivalent to reducing the delay of the delay elements in the matched delay async design—the former can be adjusted externally but the latter is an on-chip adjustment. On the basis of simulations, if the delay of the matched delay elements is more aggressively tuned (instead of simply designing to meet the 1.215 ms per computation requirement), the delay of the async design can be reduced to ~ 0.6 times worse than the sync design. Further, if speed was the primary objective, the async design can be designed to be faster by means of adopting high-speed latch controllers [8], and with largely the same energy dissipation—specifically, the slow Broad controllers adopted in the async design can be replaced by the higher speed Broadish controllers (see Fig. 8). Overall, by means of the aggressively decreasing the delay of the matched delay elements and adopting the Broadish controllers, the delay of the async design can in fact be made faster than the sync design by 0.24 times and largely without compromising energy. For completeness, the Broad controllers used were chosen and designed largely for convenience as the speed specification was relatively relaxed and not the primary concern.

An added observation from Figs. 12 and 13 is the lower functional operating voltage for the async design albeit the longer

TABLE III
CHARACTERISTICS OF RECENT FFT PROCESSORS AND THE PROPOSED ASYNC AND SYNC FFT/IFFT PROCESSORS

	V_{DD} (V)	L_{eff} (μm)	N	W_{bit} (bit)	α	Algorithm	Energy @ computation	Normalized FFTs (million) per Energy
Bass, 1999 [21]	1.10	0.60	1024	20	0.27	Radix-2	$3.14\mu J$	7.6
Wang, 2005 [22]	0.35	0.18	1024	16	0	Radix-2	$0.16\mu J$	3.7
Ding, 1999 [20]	3.30	0.35	64	24	0	Radix-4	$1.95\mu J$	3.5
Yeh, 2003 [19]	3.30	0.35	64	12	0	Split-radix	$0.33\mu J$	8.0
Sync, Fig. 11 (a)	1.10	0.35	128	16	0	Radix-2	$0.19\mu J$	5.3
Async, Fig. 11 (b)	1.10	0.35	128	16	0	Radix-2	$0.12\mu J$	8.3

delay. It fails at $V_{DD} < 1$ V largely due to the delay mismatch in some matched delay components. A dual-rail delay-insensitive approach [8] can resolve the delay mismatch problem, thereby theoretically allowing the async design to operate as long as $V_{DD} > V_T$, but at the expense of higher energy dissipation and area overhead. In the case of the sync FFT/IFFT design, the prototype IC fails at $V_{DD} < 1.1$ V. We attribute this largely to clock skew because clock skew is aggravated when gating is applied. For example from simulations, at $V_{DD} = 1$ V, the clock skew with clock gating ≈ 8.2 ns while the clock skew without clock gating ≈ 6.0 ns.

Table III tabulates a comparison of several low-energy sync FFT designs [19]–[22], including the benchmarked sync FFT/IFFT and the proposed async FFT/IFFT design herein. The entries in the last column, normalized FFTs per energy, have been scaled to a $0.35\text{-}\mu m$ CMOS, 1.1 V, 16-bit wordlength, and 128-point FFT algorithm based on (4) [21], shown at the bottom of the page, where L_{eff} is the effective transistor length of the CMOS process, W_{bit} is the wordlength, α is the truncated portion for multipliers therein, and N is the number of points of the FFT algorithm.

Although a comparison between the various designs is contentious due to large variations of the designs and parameters therein, it is nonetheless worthwhile to note that the sync and async designs described in this paper are indeed energy-efficient and the latter being the more efficient.

IV. CONCLUSION

Two low-energy 128-point 16-bit FFT/IFFT processors have been designed using the sync and async approaches, for low-voltage energy-critical low-speed hearing instrument applications. The async approach has been shown to be higher energy efficient by means of established and proposed circuit designs, including: 1) reduced redundant operations; 2) energy-efficient datapath circuits; and 3) the use of simple latches. The drawbacks of the async design have been shown to be a 10% IC area penalty and an inconsequential (for the intended application) 1.4 times worse delay. When the async design is benchmarked

against reported designs, it has been shown to be competitive in terms of energy efficiency.

ACKNOWLEDGMENT

The authors would like to acknowledge T.-P. Loy for his help with the sync FFT/IFFT processor.

REFERENCES

- [1] D. G. Gata *et al.*, "A 1.1-V 270- μA mixed-signal hearing aid chip," *IEEE J. Solid-State Circuits*, vol. 37, no. 12, pp. 1670–1678, Dec. 2002.
- [2] P. Mosch *et al.*, "A 660- μW 50-MOPS 1-V DSP for a hearing aid chip set," *IEEE J. Solid-State Circuits*, vol. 35, no. 11, pp. 1705–1712, Nov. 2000.
- [3] K.-S. Chong, B.-H. Gwee, and J. S. Chang, "A 16-channel low power non-uniform spaced filter bank core for digital hearing aids," *IEEE Trans. Circuits Syst. II*, vol. 53, no. 9, pp. 853–857, Sep. 2006.
- [4] B.-H. Gwee, J. S. Chang, and V. Adrian, "A micropower low-distortion digital class-D amplifier based on an algorithmic pulsewidth modulator," *IEEE Trans. Circuits Syst. I*, vol. 52, no. 10, pp. 2007–2022, Oct. 2005.
- [5] T. Genmeke, M. Gansen, H. J. Stockmanns, and T. G. Noll, "Design optimization of low-power high-performance DSP building blocks," *IEEE J. Solid-State Circuits*, vol. 39, no. 7, pp. 1131–1139, Jul. 2004.
- [6] J. B. Rabeay, A. Chandrakasan, and B. Nikolić, *Digital Integrated Circuits: A Design Perspective*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2002.
- [7] R. W. Brodersen, M. A. Horowitz, D. Markovic, B. Nikolić, and V. Stojanovic, "Methods for true power minimization," in *Proc. IEEE ICCAD*, 2002, pp. 35–42.
- [8] J. Sparsø and S. Furber, *Principles of Asynchronous Circuit Design: A Systems Perspective*. Norwell, MA: Kluwer Academic, 2001.
- [9] K. S. Stevens *et al.*, "An asynchronous instruction length decoder," *IEEE J. Solid-State Circuits*, vol. 36, no. 2, pp. 217–228, Feb. 2001.
- [10] L. S. Nielsen and J. Sparsø, "Designing asynchronous circuits for low power: An IFIR filter bank for a digital hearing aid," *Proc. IEEE*, vol. 87, no. 2, pp. 268–281, Feb. 1999.
- [11] Y. W. Li, G. Patounakis, K. L. Shepard, and S. M. Nowick, "High-throughput asynchronous datapath with software-controlled voltage scaling," *IEEE J. Solid-State Circuits*, vol. 39, no. 4, pp. 704–708, Apr. 2004.
- [12] A. J. Martin, M. Nyström, and C. G. Wong, "Three generations of asynchronous microprocessors," *IEEE Des. Test Comput.*, vol. 20, no. 6, pp. 9–17, Nov.–Dec. 2003.
- [13] K.-S. Chong, B.-H. Gwee, and J. S. Chang, "A low-energy asynchronous FFT/IFFT processor for hearing aid applications," in *Proc. IEEE Int. Conf. Electron Devices and Solid-State Circuits*, 2005, pp. 751–754.

$$\text{Normalized FFTs per Energy} = \frac{\left(\frac{L_{eff}}{0.35}\right) \times \left(\frac{V_{DD}}{1.1}\right)^2 \times \left[\frac{2}{3} \times \frac{W_{bit}}{16} + (1 - \alpha) \times \frac{1}{3} \times \left(\frac{W_{bit}}{16}\right)^2\right] \times \left(\frac{(N/2) \log_2 N}{448}\right)}{\text{Energy Dissipation}} \quad (4)$$

- [14] K.-S. Chong, B.-H. Gwee, and J. S. Chang, "A micropower low-voltage multiplier with reduced spurious switching," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 2, pp. 255–265, Feb. 2005.
- [15] M. Lewis, J. Garside, and L. Brackenbury, "Reconfigurable latch controllers for low power asynchronous circuits," in *Proc. IEEE Int. Symp. Advanced Research in Asynchronous Circuits and Systems*, 1999, pp. 27–35.
- [16] J. Yuan and C. Svensson, "High-speed CMOS circuit technique," *IEEE J. Solid-State Circuits*, vol. 24, no. 1, pp. 62–70, Feb. 1989.
- [17] A. Peeters and K. van Berkel, "Single-rail handshake circuits," in *Proc. IEEE Conf. Asynchronous Design Methodologies*, 1995, pp. 53–62.
- [18] K.-S. Chong, B.-H. Gwee, and J. S. Chang, "A robust low voltage low energy asynchronous carry-completion sensing adder for biomedical applications," in *Proc. IEEE BioCAS*, 2004, pp. S1/2-18–S1/2-21.
- [19] W.-C. Yeh and C.-W. Jen, "High-speed and low-power split-radix FFT," *IEEE Trans. Signal Process.*, vol. 51, no. 3, pp. 864–874, Mar. 2003.
- [20] T. J. Ding, J. V. McCanny, and Y. Hu, "Rapid design of application specific FFT cores," *IEEE Trans. Signal Process.*, vol. 47, no. 5, pp. 1371–1381, May 1999.
- [21] B. M. Bass, "A low-power, high-performance, 1024-point FFT processor," *IEEE J. Solid-State Circuits*, vol. 34, no. 3, pp. 380–387, Mar. 1999.
- [22] 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.

Kwen-Siong Chong (S'03) received the B.Eng. and M.Phil. degrees in electrical and electronic engineering from Nanyang Technological University (NTU), Singapore, in 2001 and 2002, respectively. He has submitted his Ph.D. thesis to NTU, Singapore, in 2006.

He is presently a Research Associate with NTU. His research interests include asynchronous VLSI designs, low-voltage low-power VLSI circuits, audio signal processing, and digital filterbank designs.

Bah-Hwee Gwee (S'93–M'97–SM'03) received the B.Eng. degree in electrical and electronic engineering from the University of Aberdeen, Aberdeen, U.K., in 1990, and the M.Eng. and Ph.D. degrees from Nanyang Technological University (NTU), Singapore, in 1992 and 1998, respectively.

He worked as a Research Engineer in a National Science and Technology Board (NSTB) funded project in NTU in collaboration with SEIKO—Human Interface Engineering Lab from 1990 to 1993. From 1995 to 1998, he was a Lecturer at Temasek Polytechnic, Singapore. He was an Assistant Professor in the School of Electrical and Electronics Engineering, NTU, from 1999 to 2005, and has been an Associate Professor at NTU since 2005. He is the principal investigator of several project grants including the ASEAN-EU University Network Programme (AUNP), NTU-Panasonic Collaboration, NTU-Lingköping University Collaboration and Defence Science and Technology Agency (DSTA) projects. He has filed several patents in circuit design and has one U.S. patent granted. His research interests include low-power asynchronous microprocessor and digital signal processor design, Class-D amplifiers and soft-computing.

Dr. Gwee was the Chair of IEEE Singapore—Circuits and Systems Chapter in 2005 and 2006. He has been a member of IEEE CAS DSP and VLSI Technical Committee since 2004. He has served as the secretary of IEEE BioCAS-2004, publication chair of IEEE APCCAS-2006, and Technical Program Chair of ISIC-2007.

Joseph S. Chang received the B.Eng. degree in electrical and computer systems engineering from Monash University, Melbourne, Australia, in 1983 and the Ph.D. degree from the Department of Otolaryngology, University of Melbourne, Melbourne, Australia, in 1990.

He previously worked for CSIRO, Melbourne, and Texas Instruments, Singapore, as an Engineer. From 1989 to 1991, he was a Senior Research Scientist/Engineer at the Human Communication Research Centre, University of Melbourne. He is presently the Associate Dean (Research and Graduate Studies) at the College of Engineering, Nanyang Technological University, Singapore. His research interests include analog and digital signal processing, very large-scale integration design, speech perception, biomedical engineering, and hearing instrument (hearing aid) research. He holds several patents and has several pending patents in circuit design.

Dr. Chang received the Commendation for the Best Presentation of a Paper Award in 1989 for a paper presented at the Microelectronics Conference. He served as the Chairperson of the International Symposium on Integrated Circuits, Devices and Systems (ISIC-2004). He is a co-founder of a high-tech spin-of company specializing in electro-acoustics. He currently serves as an associate editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, and as a co-editor of the "Open Column" in the *IEEE Circuits and Systems Magazine*.