A 1.1 G MAC/s Sub-Word-Parallel Digital Signal Processor for Wireless Communication Applications

Yuan-Hao Huang, *Member, IEEE*, Hsi-Pin Ma, *Member, IEEE*, Ming-Luen Liou, and Tzi-Dar Chiueh, *Senior Member, IEEE*

***Abstract*—This work proposes a communication digital signal processor (DSP) suitable for massive signal processing operations in orthogonal frequency division multiplexing (OFDM) and code-division multiple-access (CDMA) communication systems. The OFDM-based IEEE 802.11a wireless LAN transceiver and CDMA-based WCDMA uplink receiver are simulated to evaluate the computation requirements of future communication systems. The architecture of the communication digital signal processor is established according to the computational complexity of these simulations. The proposed architecture supports basic butterfly operations, single/double-precision and real- and complex-valued multiplication-and-accumulation (MAC), squared error computation, and add-compare-select (ACS) operation. This butterfly/complex MAC architecture can greatly enhance the execution efficiency of operations often found in communication applications. The processor chip is fabricated using a 0.35-μm n-well one-poly four-metal CMOS technology. The fabricated DSP chip reaches a speed of 1.1 G MAC/s when operating in the high-speed mode, and it achieves 4 M MAC/s/mW in the low-power mode.**

***Index Terms*—Digital signal processor (DSP), sub-word parallelism, wireless communication.**

I. Introduction

WIDEBAND communication applications, with their high computational complexity, have increased demand for processing power and hardware efficiency. Next-generation communication systems often employ code-division multiple-access (CDMA) and/or orthogonal frequency division multiplexing (OFDM) techniques to combat nonideal effects in the wireless communication channels. Although the two communication schemes can achieve efficient and high-quality wideband communication, their architectures are complicated and usually require components that have high computational complexity. On the other hand, the trends of highly integrated and rapidly reconfigurable systems are such that traditional hardwired devices may not be suitable for future communication systems. Accordingly, in recent years, software-defined radio (SDR) has attracted much attention from researchers worldwide [1], [2] because it features a high degree of flexibility and adaptation. Moreover, low power consumption is a major consideration for wireless communication devices. Thus, a low-power and computationally efficient digital signal processor (DSP) will be a crucial component in future communication systems.

Recently, several function-specific DSPs have been proposed for communication applications in order to improve the computational capability and to reduce the costs of development. For example, TMS320C54x [3] supports special instructions for the Viterbi decoding, which is popular in many wireless communication systems. TMS320C55x [6] contains two separate multiplication-and-accumulation (MAC) cores to accelerate multiply-and-accumulate instructions. Some DSPs [5], [6] are designed to accelerate the multiply-and-accumulate (MAC) operation by properly reconfiguring their parallel multipliers. Moreover, a primary feature of communication signal processing is the in-phase/quadrature-phase (I/Q) complex-valued signal computation, such as phase rotation in carrier recovery, I/Q signal filtering, fast Fourier transform (FFT), and others [7]. These operations cannot be executed efficiently by traditional real-valued MAC datapath. Therefore, some DSPs are designed with a view to improving the efficiency of the complex-valued MAC operation. The MDSP-II processor [8] can perform a complex-valued MAC operation in two clock cycles through two 16×16 multipliers. The CDSP processor [9] is designed to compute a single-precision 8×8 complex MAC in one clock cycle.

Two important issues concerning the development of communication-specific digital signal processing arise. One concerns the real-/complex-valued signal processing and the other concerns the sub-word parallel processing. No architecture yet exists that can support, with 100% utilization, both double-/single-precision and real-/complex-valued MAC operations. An architecture with such features can dramatically increase the computational efficiency in communication signal processing. Furthermore, most sub-word parallel processors that accommodate both high-precision and low-precision signals suffer from the problem of data misalignment [10], which may incur overhead in both hardware and software. This paper proposes a novel MAC architecture that solves these problems [11]. The proposed DSP architecture has a butterfly unit

Manuscript received February 19, 2003; revised July 23, 2003. This work was supported in part by National Science Council, Taiwan, R.O.C., under Grants NSC90-2218-E-002-039 and NSC90-2219-E-002-010, and the Ministry of Education, Taiwan, R.O.C., under Grant 89-E-FA06-2-4-7.

Y.-H. Huang was with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan 10617 R.O.C. He is now with VXIS Technology Company, Hsin-Chu, Taiwan 300, R.O.C.

H.-P. Ma was with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan 10617 R.O.C. He is now with the Department of Electrical Engineering, National TsingHua University, Hsin-Chu, Taiwan 300, R.O.C.

M.-L. Liou was with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan 10617 R.O.C. He is now with MediaTek Inc., Hsin-Chu, Taiwan 300, R.O.C.

T.-D. Chiueh is with the Graduate Institute of Electronics Engineering and the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan 10617 R.O.C.

Digital Object Identifier 10.1109/JSSC.2003.820861

0018-9200/04$20.00 © 2004 IEEE

Authorized licensed use limited to: National Taiwan University. Downloaded on February 26, 2009 at 23:21 from IEEE Xplore. Restrictions apply.

TABLE I
COMPUTATION COMPLEXITY IN MILLION OPERATIONS PER SECOND (MOPS) FOR THE IEEE 802.11a WIRELESS LAN RECEIVER AND UPLINK WCDMA RECEIVER

OFDM function	Main Operation	Total (MOPS)
FFT	Butterfly	400.00 (73%)
Correlation	MAC	3.98 (0.7%)
Loop Filter	MAC	113.96 (20.3%)
Subcarrier Operation	Complex Multiplication	33.33 (6%)
Viterbi Decoder	ACS	2244.99(QPSK) 4489.98(16-QAM) 6734.98(64-QAM)

CDMA function	Main Operation	Total (MOPS)
Channel Estimation	MAC	4131.93 (75%)
Carrier Synchronizer	Complex Multiplication	107.73 (2%)
Rake Combiner	MAC	368.895 (6%)
Beam Former	MAC	983.295 (17%)

capable of performing various types of butterfly operations in the FFT procedure, which is essential in OFDM systems. Also, the unit can be reconfigured to perform squared-difference and add-compare-select (ACS) operations required in the Viterbi algorithm. A sub-word parallel complex-valued MAC architecture is proposed to execute complex- and real-valued MAC operation with different precision. The 100% utilization of the complex MAC architecture makes it possible to efficiently perform convolution operations such as equalizers, matched filters, and pulse-shaping filters.

The remainder of this paper details the proposed DSP architecture and its implementation. Section II addresses the computational requirements in next-generation communication applications and present state-of-the-art DSP solutions. Section III demonstrates the architecture of the proposed DSP. Section IV introduces the detail of the complex MAC core. Section V presents the chip implementation and experimental results. Finally, a brief conclusion is given in Section VI.

II. KEY OPERATIONS IN COMMUNICATION DSP

OFDM and CDMA are the two wireless communication techniques adopted by various communication standards. In view of this fact, operations and associated complexity of the baseband signal processing in the receivers of two popular wireless communication standards were analyzed. The receivers are the OFDM-based IEEE 802.11a wireless LAN transceiver [12] and the CDMA-based WCDMA uplink receiver [13]. Table I lists computational requirements of some key processing tasks in both receivers. For the OFDM-based wireless LAN, in addition to the Viterbi decoder, the FFT operation, filtering, and subcarrier frequency-domain operations all demand a heavy computational load, mainly in the form of butterfly, MAC, ACS, and complex multiplication operations. For the CDMA-based WCDMA, channel estimator, carrier synchronizer, rake combiner, and beamformer make up an enormous amount of computational complexity, again in the form of MAC and complex multiplication operations. According to the above analysis, computation in the OFDM and CDMA receivers involves mainly butterfly operation (73% in OFDM), the MAC operation (20% in OFDM and 75% in CDMA), and the ACS operation in the Viterbi decoder. The above analysis clearly establishes that efficient execution of the ACS operation, the butterfly operation, and the complex-valued MAC operation is essential to DSPs for communication applications. In addition, issues such as signal precision and real-/complex-valued operations also strongly influence the efficiency of low-cost real-time software implementation of such baseband receivers. Several specialized hardware for accelerating the above operations in state-of-the-art DSPs are described next.

A. Configurable Complex- and Real-Valued MAC

The MAC operation is one of the most important features of a DSP, not only for communication applications but also for multimedia applications. The main difference between these two types of applications is that baseband processing in communication applications often involves complex-valued signals (I/Q components). Traditional DSPs with a real-valued MAC datapath cannot efficiently execute arithmetic operations on complex-valued signals. Accordingly, recent DSPs are designed with a view to improving the efficiency of the complex-valued MAC operation [5], [6], [8], [9]. The complex-valued MAC operation consists of four real-valued multiplications and two accumulations, which can be formulated as

$$\begin{aligned}&(\mathrm{ACCR}+j\mathrm{ACCI})+X\times Y\\&\quad=(\mathrm{ACCR}+j\mathrm{ACCI})+(X_R+jX_I)\times(Y_R+jY_I)\\&\quad=(\mathrm{ACCR}+j\mathrm{ACCI})+(X_RY_R-X_IY_I)\\&\quad+j(X_RY_I+X_IY_R).\end{aligned}\tag{1}$$

Dual MAC architectures, such as DSP16000 [6] and MDSP-II [8], use two parallel 16×16 multipliers to compute the products and sum the real and imaginary parts, respectively, in two cycles, thereby completing one complex-valued MAC operation. The swap multiplexer in DSP16000 and MDSP-II is used to select a pair of inputs from four operands and generate four partial products in the complex-valued multiplication operation. The other dual 16×16 MAC architecture in LODE [5] operates by a similar strategy except that it employs a delay register to store the input operands for the next MAC operation. This approach also executes one complex-valued MAC in two cycles. The latest complex-valued MAC architecture in CDSP [9] can process one complex-valued MAC in a single instruction cycle, but only low-precision 8×8 complex MAC operation is supported. Four instruction cycles are required for the 16×16 complex-valued MAC operation. Straightforward extension of the previous architectures to a 16×16 complex-valued MAC architecture using four parallel real MAC units is possible but unpopular, primarily because of its lack of hardware efficiency. Although the four-parallel-MAC architecture can perform one

Authorized licensed use limited to: National Taiwan University. Downloaded on February 26, 2009 at 23:21 from IEEE Xplore. Restrictions apply.

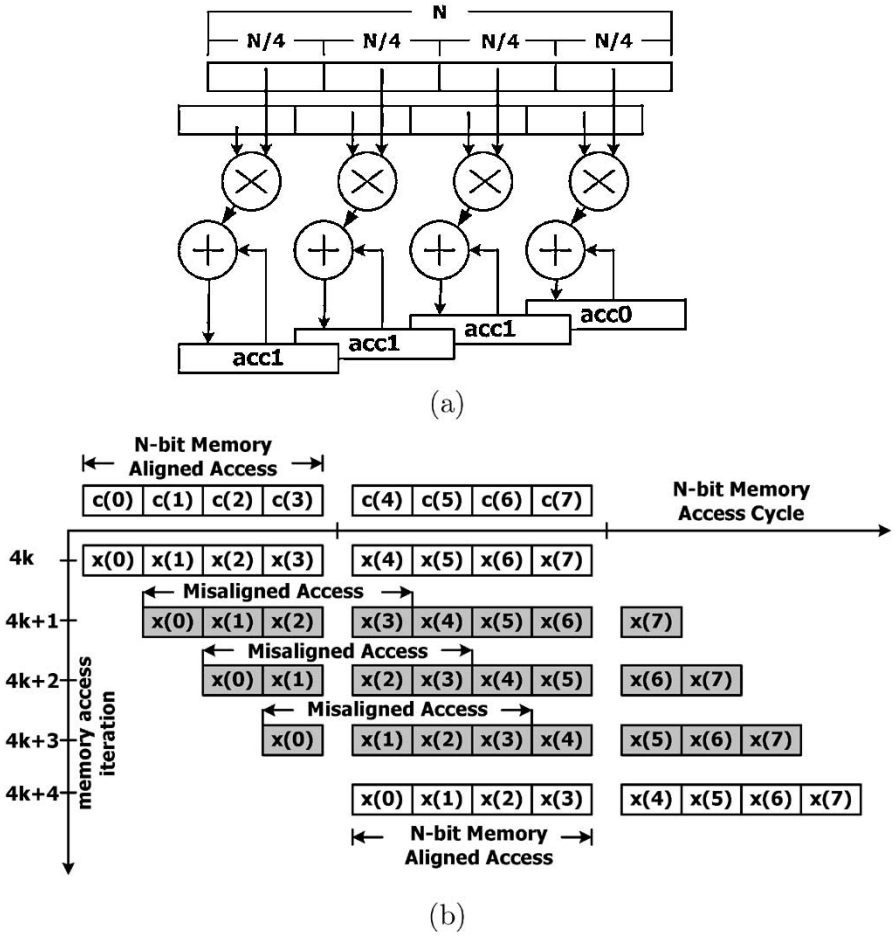


Fig. 1. (a) Sub-word parallel MAC architecture. (b) Data misalignment problem in sub-word parallel DSPs.

complex-valued MAC operation in one cycle, three MAC units sit idle in a real-valued MAC operation. An efficient four-parallel MAC architecture that can fully utilize all four MAC units is proposed in the DSP chip.

B. Computation With Signals of Various Precisions

The variety of modulation methods and receiving techniques in communication applications dictate that single- and double-precision arithmetic operations be executed flexibly and efficiently by a DSP programmed as a software receiver. For example, most processing in high-order quadrature amplitude modulation (QAM) and OFDM communication systems requires high-precision computation. However, low-precision computation is sufficient for some major processing tasks in communication receivers, such as matched filtering and pseudorandom (PN) code spreading/despreading in direct-sequence spread spectrum (DSSS) CDMA systems. Yet others, such as channel estimation and equalization, have different precision requirements under different conditions. In response to this fact, data-level parallelism, also called sub-word parallelism (SWP), has been adopted in several recent DSP designs. Fig. 1(a) shows the sub-word parallel MAC used in the TigerSHARC DSP [10]. When the N-bit MAC module is switched to do $N/4$-bit MAC operation, it is divided into four $N/4$-bit sub-word MAC modules and four parallel MAC operations are issued per cycle on sub-words of $N/4$ bits. Each of the four products is accumulated in one of the four dedicated accumulation registers.

Despite its hardware utilization efficiency, the SWP processing suffers from the data misalignment problem, as shown in Fig. 1(b). For the finite-impulse response (FIR) operation using the low-precision format, the filter outputs at time $t \neq 4k$ will encounter memory misalignment, that is, the data needed are not aligned with the N-bit memory words. The TigerSHARC DSP employs a dedicated hardware, called the data-alignment buffer, and extra instructions to address the appropriate low-precision sub-word in due time. In another general-purpose SWP-type MMX processor [14], data rearrangement and data reformatting instructions are provided to re-align the misaligned low-precision data. Both designs incur overhead in hardware as well as in software.

Although many recent communication signal processors recognize the importance of a complex-valued MAC unit with sub-word parallel structure, most of them provide either a complex-valued MAC without SWP or a sub-word parallel MAC that can process only real-valued data. In the proposed DSP, an MAC architecture that combines both features is designed, and simultaneously the problem of data misalignment is addressed.

Authorized licensed use limited to: National Taiwan University. Downloaded on February 26, 2009 at 23:21 from IEEE Xplore. Restrictions apply.

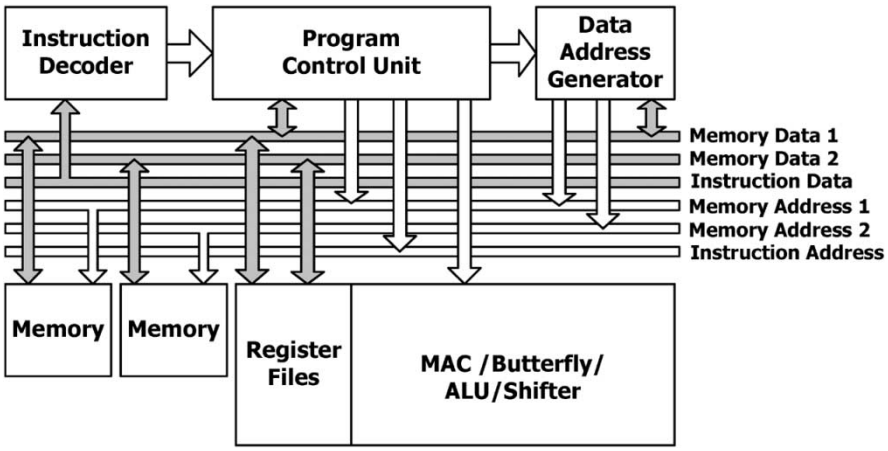


Fig. 2. Architecture of the proposed communication DSP.

III. Processor Architecture

Fig. 2 shows the architecture of the proposed 16-bit DSP. The processor has one bank of program memory and two independent banks of data memory that can be separately addressed. The address generation unit (AGU) can generate, in one cycle, two data addresses that are used to access two operands via dual data and dual address buses. This architecture can provide efficient memory access in complex-valued operations, which occur relatively often in wireless communication baseband processing. The AGU has 13 index address registers, including eight normal registers and five specific registers for special addressing modes, such as modular addressing, circular addressing, and bit-reversed addressing (needed in FFT). It also supports post-increment/post-decrement by one/two/constant. The program control unit supports zero-overhead looping and processes call subroutines, interrupt subroutines, and conditional branches. The program control unit consists of a program counter (PC), an instruction register, stack registers, some flag registers, and an interrupt controller that can accommodate two external interrupt signals. In addition, two $1\mathrm{K} \times 16$ $\cos/\sin$ ROM modules store sinusoidal wave samples that are needed in signal processing tasks such as FFT and numerically controlled oscillator. The main datapath units include an arithmetic logic unit (ALU), a barrel shifter, and a butterfly/complex-valued MAC (BCMAC) module. The function of the BCMAC module will be described in the following.

A. Butterfly Processor

The butterfly processor, shown in Fig. 3, is designed with a view to supporting one-cycle execution of decimation-in-frequency FFT operations. The proposed processor has five special instructions for execution of six basic butterfly operations in FFT algorithms, such as Cooley-Tukey radix-2/4 [15], recursive-radix [16], radix-2^n [17], split-radix [18], and others. All basic FFT operations except the decimation-in-time butterfly operation can be executed in one clock cycle. The twiddle factors are supplied from the built-in cos/sin ROM, which stores 1024 sinusoidal wave samples (1/8 of a period) and can support FFT computation with up to 8192 points.

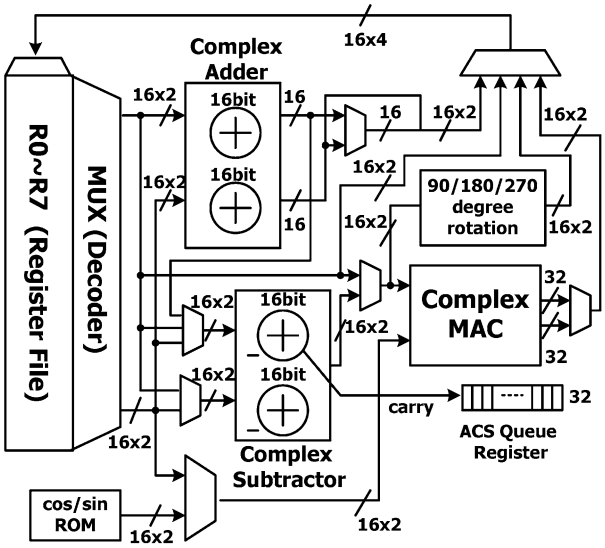


Fig. 3. Block diagram of the butterfly/complex-valued MAC (BCMAC) module.

1) Add-Compare-Select Operation for the Viterbi Algorithm: Convolutional coding is the most popular error correcting coding scheme in wireless communications. The Viterbi algorithm on the code trellis diagram is widely adopted to decode convolutional-coded data. In this algorithm, the most computation-intensive operation is the ACS operation. The ACS operation can be executed in one cycle using the complex adder/subtractor in the BCMAC module. Moreover, a 32-bit queue is included to store the comparison results of the ACS operation, which are needed during the backward path tracing in the decoding process [7].

2) Sum of Squared Difference Operation: The Euclidean distance between two complex values and the sum of squared difference operation are significant in computing the minimum mean square error (MMSE) or maximum likelihood (ML), often found in channel estimation and channel equalization algorithms for communication systems. When computing the Euclidean distance between two complex-valued numbers, the

Authorized licensed use limited to: National Taiwan University. Downloaded on February 26, 2009 at 23:21 from IEEE Xplore. Restrictions apply.

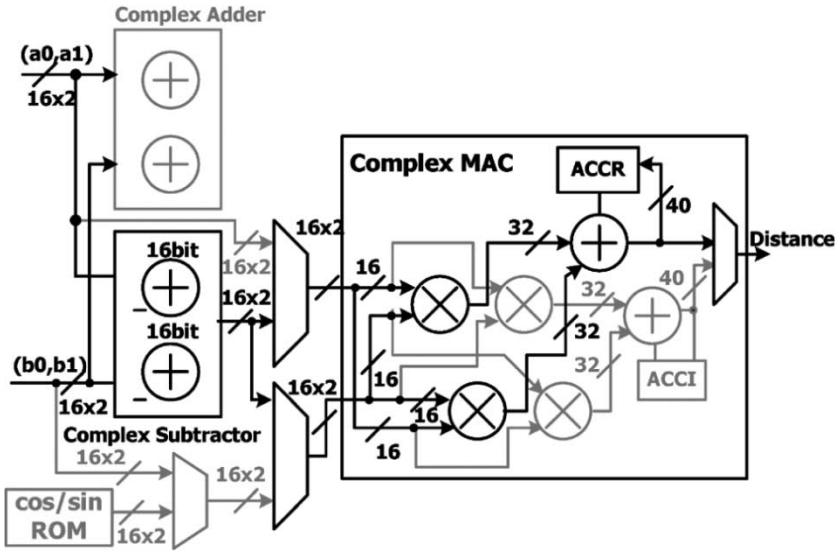


Fig. 4. BCMAC configuration for computing sum of squared difference. Note (a_0, a_1) and (b_0, b_1) are two complex values whose difference will be computed, squared, and possibly accumulated.

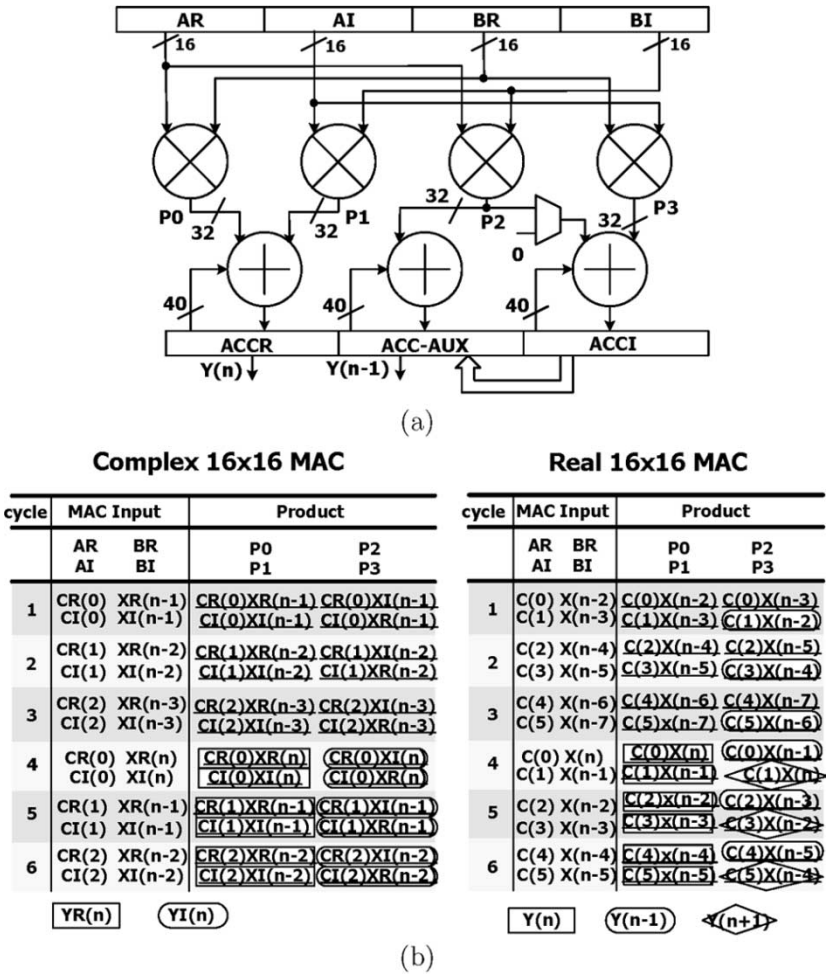


Fig. 5. (a) MAC architecture for 16 × 16 complex-valued MAC and 16 × 16 real-valued MAC operations. (b) Processing schedules of a complex 3-tap and a real 6-tap double-precision FIR convolution using the architecture in (a).

two squared differences are summed. When computing the squared difference between two complex-valued vectors, the two squared differences should be accumulated. In this regard, the BCMAC module can be configured, as shown in Fig. 4, to perform both subtract-square-add and subtract-square-accumulate operations.

3) Other Real/Complex-Valued Operations: The proposed BCMAC module can also execute other basic arithmetic

Authorized licensed use limited to: National Taiwan University. Downloaded on February 26, 2009 at 23:21 from IEEE Xplore. Restrictions apply.

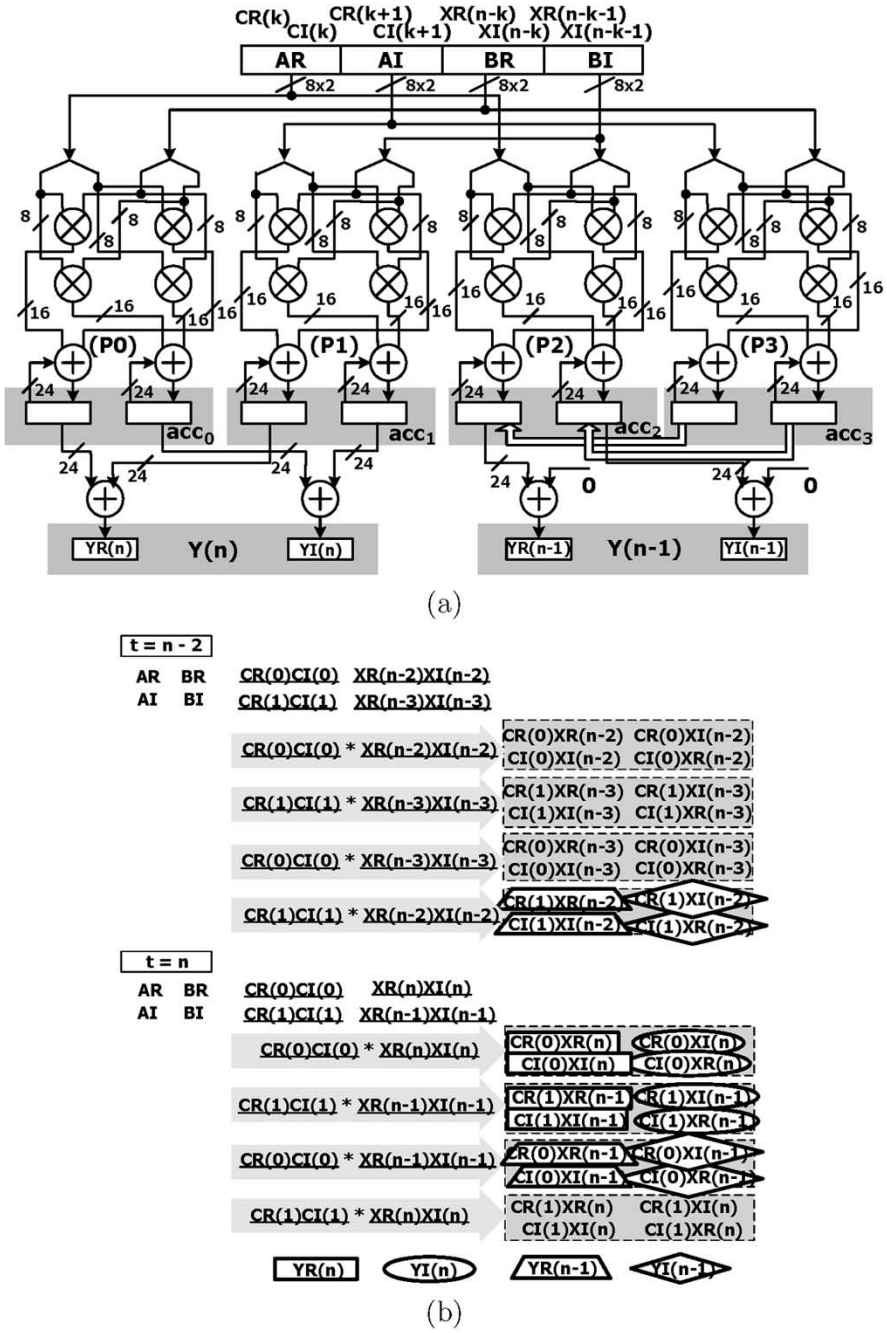


Fig. 6. (a) Single-precision complex-valued MAC architecture. (b) Processing schedule of a complex single-precision 2-tap FIR convolution.

operations, such as real-/complex-valued addition/subtraction/multiplication. Moreover, the complex-valued constants stored in the two internal $\cos/\sin$ ROMs can work as one operand to the complex multiplier in cases such as the phase rotator in a carrier recovery loop.

B. Complex-Valued MAC Architecture

A MAC circuit should preferably be able to accommodate real-/complex-valued as well as single-/double-precision signal format due to a large variety of signal formats in communication systems. This subsection introduces the detail of the architecture of a sub-word parallel complex-valued MAC for this purpose. The four real double-precision (16×16) multipliers in the BCMAC module can be properly configured to execute in one cycle: 1) one double-precision (16×16) complex-valued MAC operation; 2) four double-precision (16×16) real-valued MAC operations; 3) four single-precision (8×8) complex-valued MAC operations; or 4) 16 single-precision (8×8) real-valued MAC operations.

1) Double-Precision Complex-Valued MAC Operation: The double-precision (16×16) complex-valued MAC is composed of four real double-precision (16×16) multipliers and two accumulators as shown in Fig. 5(a). (ACC-AUX is not used in this case.) Fig. 5(b) shows an example of the execution of a 3-tap

Authorized licensed use limited to: National Taiwan University. Downloaded on February 26, 2009 at 23:21 from IEEE Xplore. Restrictions apply.

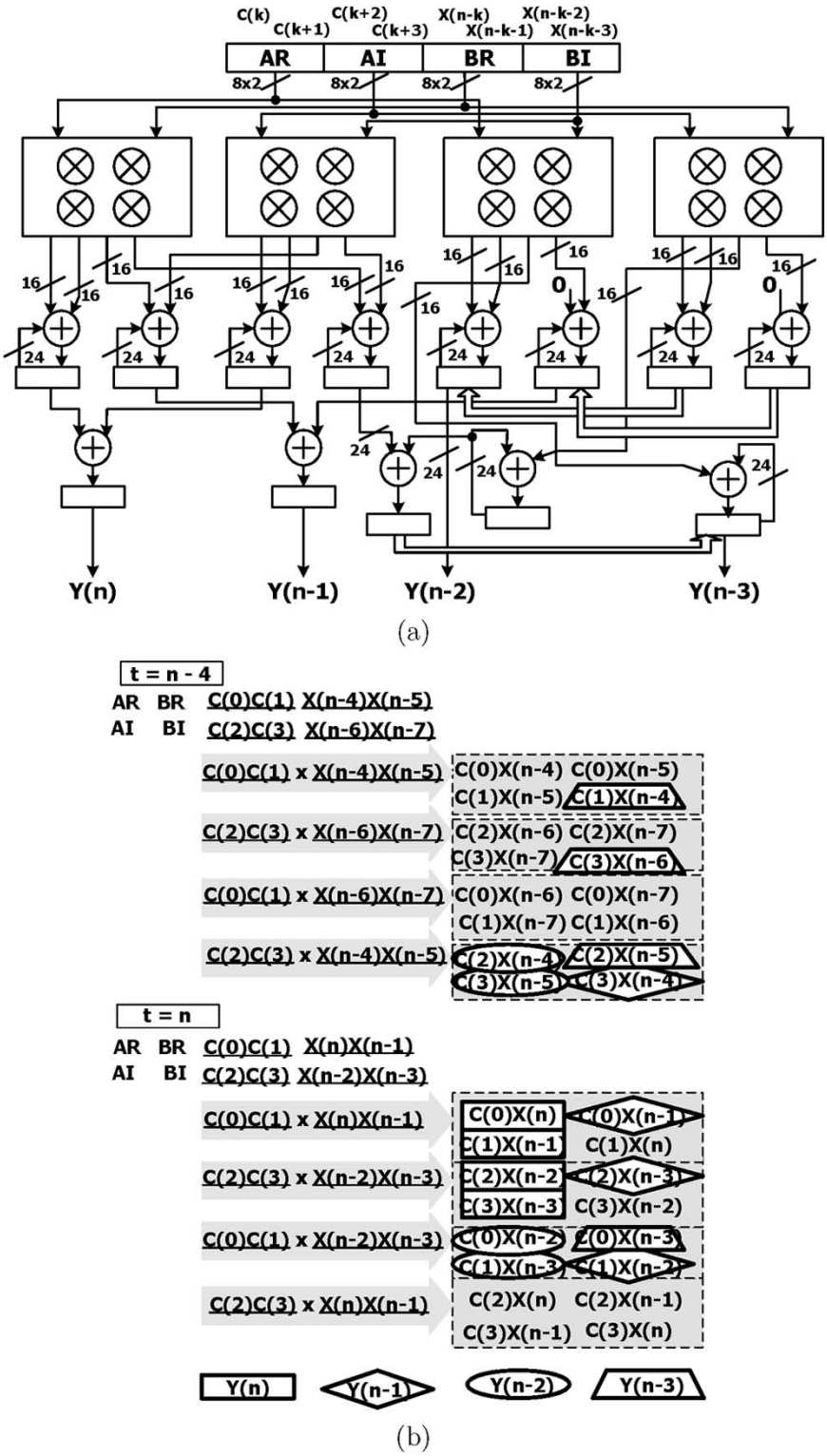


Fig. 7. (a) Single-precision real-valued MAC architecture. (b) Processing schedule of a real single-precision 4-tap FIR convolution.

complex FIR filter to compute two output samples in six cycles. In this case, the multiplexer in the MAC is switched to accumulate both P2 and P3 products in ACCI. Accordingly, the two 40-bit accumulators for the real part (ACCR) and the imaginary part (ACCI) together hold the accumulated complex-valued products. Note that a double-precision complex-valued MAC operation is executed in one cycle.

2) Double-Precision Real-Valued MAC Operation: To realize real-valued double-precision (16×16) MAC, let us first examine the FIR filter operation

$$Y(n) = \sum_{k=0}^{K-1} C(k) \cdot X(n-k). \tag{2}$$

Authorized licensed use limited to: National Taiwan University. Downloaded on February 26, 2009 at 23:21 from IEEE Xplore. Restrictions apply.

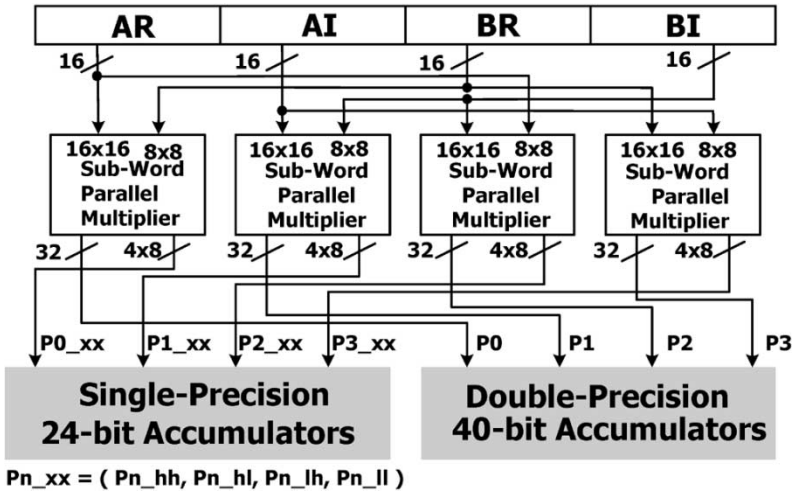


Fig. 8. Block diagram of the sub-word parallel multiplier based complex MAC. Each sub-word multiplier can execute either one 16×16 multiplication or four 8×8 multiplications.

In a typical order-K FIR filter, K pairs of 16-bit inputs $X(n)$ and 16-bit coefficients $C(n)$ are multiplied and accumulated as output $Y(n)$. In the complex MAC module, at each cycle two consecutive coefficients are loaded into the A register in Fig. 5(a), one in the real part and the other in the imaginary part to achieve maximum hardware utilization. Similarly, two consecutive input samples are loaded into the B register. The four multipliers and three accumulators then execute their operations. Fig. 5(b) shows an example of a 6-tap FIR filter with coefficients $C(0), C(1), \ldots, C(5)$. Note that in three cycles all six data samples and coefficients are loaded and product terms in three consecutive outputs, $Y(n-1)$ (ellipsoid), $Y(n)$ (rectangle), $Y(n+1)$ (rhombus), are computed. Therefore, in three $(K/2)$ cycles, $Y(n)$ can be computed in ACCR. Because now the multiplexer in Fig. 5(a) is switched to zero, half of the terms in $Y(n-1)$ and $Y(n+1)$ will be accumulated. At the end of each FIR iteration, $Y(n)$ is in ACCR and $Y(n-1)$ is in ACC-AUX. Before the next iteration, the content in ACCI, a partially accumulated sum of the next output $Y(n+1)$, must be moved to ACC-AUX. Accordingly, these four 16×16 multipliers compute all the time and $4\times$ speed-up can be achieved.

3) Single-Precision Complex-Valued MAC Operation: Each 16×16 multiplier in the 16×16 complex-valued MAC can be configured as four 8×8 multipliers that can compute sub-word products. With two additional accumulators, the four single-precision multipliers can work as one independent single-precision (8×8) complex-valued MAC. In the operation of an order-K single-precision complex-valued FIR filter, the processing schedule is the same as that of the 16×16 real-valued MAC except that $C(n)$ and $X(n)$ are now combinations of 8-bit real parts ($CR(n)$ and $XR(n)$) and 8-bit imaginary parts ($CI(n)$ and $XI(n)$). Therefore, four 8×8 complex-valued MACs can be configured as shown in Fig. 6(a). Referring to Fig. 6(b), one sees that at the end of each iteration (one cycle in this example), complex accumulators acc_0 and acc_1 are summed as output $Y(n)$ and complex accumulator acc_2 contains output $Y(n-1)$. Before the next iteration, the content of acc_3 is moved to acc_2 for the next output $Y(n+1)$. Consequently, four 8×8 complex-valued MAC operations can be executed in one cycle. Moreover, the processing schedule in Fig. 6(b) shows that all access to $X(n)$ and $C(n)$ are word-aligned (even indices). Thus, the data misalignment problem encountered in most SWP-based DSPs is avoided and no index realignment is necessary.

4) Single-Precision Real-Valued MAC Operation: To compute single-precision real-valued MAC, the MAC module is configured as Fig. 7(a). To understand how this works, we take a 4-tap FIR filter as an example. The processing of this filtering is shown in Fig. 7(b). Note that $C(n)$ and $X(n)$ are 8-bit values, and four consecutive coefficients and four consecutive samples can be loaded into A and B registers, respectively at each cycle. During the execution of the current instruction $(t = n)$, all four terms in $Y(n)$ (rectangle) are computed. Additionally, several terms that contribute to $Y(n-1)$ (rhombus), $Y(n-2)$ (ellipsoid), and $Y(n-3)$ (trapezoid) are also computed. The remaining terms in $Y(n-1), Y(n-2)$, and $Y(n-3)$ were already computed in the previous instruction $(t = n-4)$. Thus, four output samples $Y(n), Y(n-1), Y(n-2)$, and $Y(n-3)$ can be computed in one cycle (one iteration). Before the next iteration, three partially accumulated sums are moved to the corresponding accumulators (bold arrows). Therefore, this MAC configuration can effectively execute 16 single-precision (8×8) real-valued MAC operations in one cycle. Most important of all, the data $X(n)$ and $C(n)$ are always aligned with index $n = 4k$, and hence there is no data misalignment problem.

IV. Circuit Design

As described in Section III, the butterfly processor can perform several 16-bit operations, including butterfly operations, the ACS operation, the sum of squared difference operation, and real/complex-valued operations. Furthermore, the structure of the accumulators in the complex MAC circuit must be carefully designed so that the four different MAC operations with real/complex-valued double-/single-precision operands can be executed with 100% hardware utilization and free of the data misalignment problem.

Authorized licensed use limited to: National Taiwan University. Downloaded on February 26, 2009 at 23:21 from IEEE Xplore. Restrictions apply.

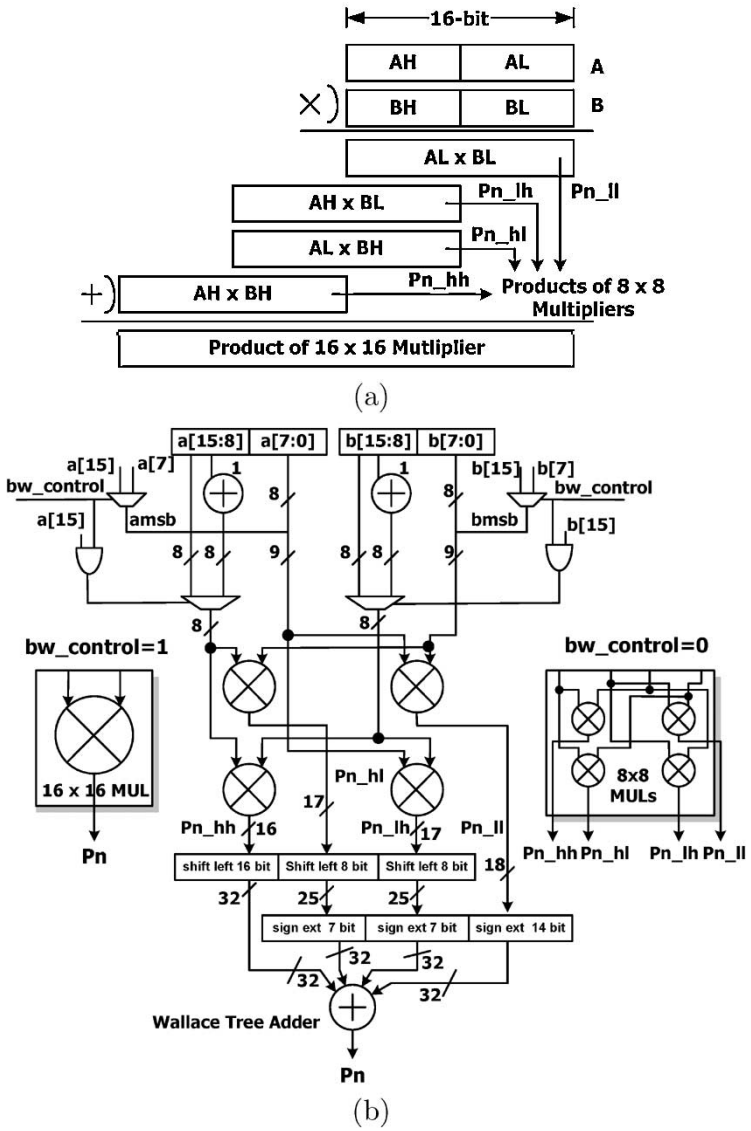


Fig. 9. (a) Breakdown of sub-word parallel multiplication and (b) the proposed sub-word parallel multiplier circuit. In the single-precision mode (bw_control is 0), the circuit generates four 16-bit products *Pn_hh, Pn_hl, Pn_lh*, and *Pn_ll*. In the double-precision mode (bw_control is 1), it generates a 32-bit product Pn.

A. Complex MAC Circuit

Fig. 8 shows the block diagram of the complex MAC circuit. Each of the four sub-word parallel multipliers can be used as either a 16×16 multiplier or as a set of four 8×8 multipliers. In the single-precision mode, four sets of 16-bit products Pn_hh, Pn_hl, Pn_lh, and Pn_ll from the nth sub-word parallel multiplier are fed to ten 24-bit accumulators to implement two different types of single-precision MAC operations (real-valued operands or complex-valued operands). In the double-precision mode, four 32-bit products, $P0, P1, P2$, and $P3$, are fed to three 40-bit accumulators for two types of double-precision MAC operations. We will introduce the sub-word parallel multiplier circuit before detailing the complex MAC circuit.

B. Sub-Word Parallel Multiplier Circuit

The sub-word parallel multiplier is based on the typical multiplication chart illustrated in Fig. 9(a). The 16-bit data A can be divided into an 8-bit MSB part, AH, and an 8-bit LSB part, AL; likewise for B. The four 16-bit partial products (Pn_hh, Pn_hl, Pn_lh, and Pn_ll) can be aligned and summed to form the 32-bit product Pn. These four partial products can also be useful in single-precision MAC operations.

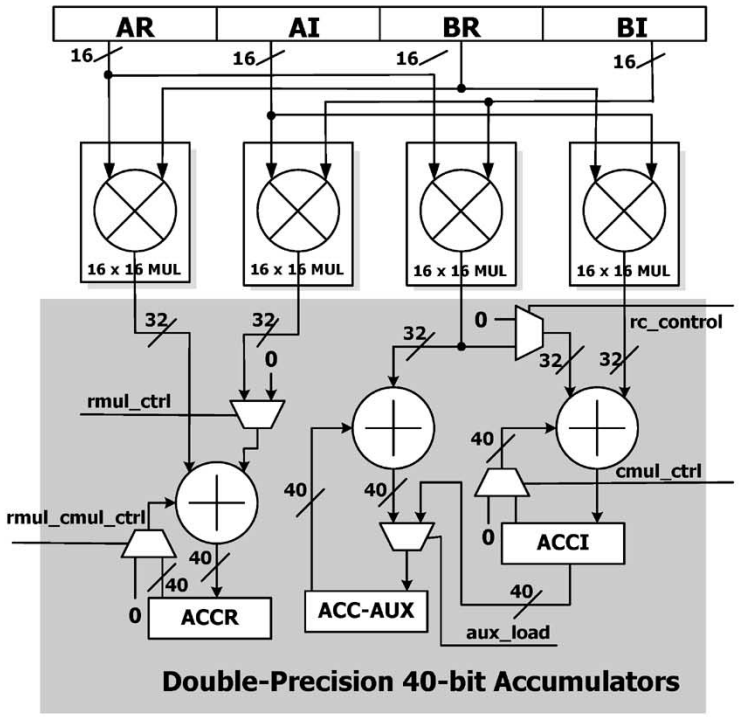
The signs of the double-precision and single-precision signals must be carefully treated in the sub-word parallel multiplier. As shown in Fig. 9(b), the control signal bw_control determines in what precision the multiplier operates. In practice, one 8×8, two 9×8, and one 9×9 Baugh–Wooley multipliers are used to compute the four single-precision products when bw_control is 0. In the double-precision mode (when bw_control is 1), the four product terms are properly shifted, sign-extended to 32-bit operands, and summed by a four-input adder to generate a 32-bit product.

1) Double-Precision MAC Circuit: The double-precision MAC architecture in Fig. 5(a) can execute one complex double-precision MAC operation or four real double-precision MAC operations in one cycle. An example of a K-tap FIR using the double-precision complex-valued MAC instruction

Authorized licensed use limited to: National Taiwan University. Downloaded on February 26, 2009 at 23:21 from IEEE Xplore. Restrictions apply.

	WCMAC (DACC_en=1) (rc_control=1)	WRMAC (DACC_en=1) (rc_control=0)	WRLD (aux_load=1)	CMUL (cmul_ctrl=1)	MUL (rmul_ctrl=1)	SQDSTA (square_distancea=1)
ACCR	AR x BR	AR x BR		AR x BR	AR x BR	AR x BR
	AI x BI	AI x BI		AI x BI	zero	AI x BI
	QACCR	QACCR		zero	zero	QACCR
ACC-AUX	×	AI x BR	QACCI	×	×	×
	×	QACC-AUX		×	×	×
ACCI	AI x BR	zero		AI x BR	×	AI x BR
	AR x BI	AR x BI		AR x BI	×	AR x BI
	QACCI	QACCI		zero	×	QACCI

(a)



(b)

Fig. 10. (a) Input signals to be summed by the accumulators of the double-precision MAC circuit in six different cases. (b) Proposed double-precision MAC circuit.

is demonstrated in the following, in which $K + 5$ cycles are needed.

- Double-precision complex-valued MAC routine:

$$\begin{array}{rl} & \text{LDSR } p\_addr \\ & \text{LDER } p\_addr \\ & \text{LDCX } \#K \\ p\_addr: & \text{WCMAC M[IR0++2]M[IR1++2]} \\ & \text{STACR M[IR2++2]} \\ & \text{STACI M[IR2++2].} \end{array}$$

For the real double-precision MAC operation, the value of the ACCI register must be loaded to the ACC-AUX register, after each convolution iteration, using an extra instruction. In this case, the MAC circuit needs $(K/2) + 6$ clock cycles to perform one iteration in a K-tap FIR convolution. Since two output samples are computed, an effective speed-up factor of four is achieved.

In addition, the complex double-precision multiplication (CMUL), the real double-precision multiplication (MUL), and the double-precision sum of squared difference (SQDSTA) instructions are also computed by this circuit. Fig. 10(a) lists the respective operands that need be summed for the three accumulators during the execution of the above six instructions. The 40-bit ACCR and ACCI accumulators both have to sum at most three operands while the 40-bit ACCR-AUX accumulator has to sum at most two inputs. The shadowed rectangles in Fig. 10(a) indicate that one 2-to-1 multiplexer is needed since each of these rows has two possible operands. It should be noted that the WRLD instruction moves the value in the ACCI register directly to the ACC-AUX register instead of the input of the adder before ACC-AUX. Thus, the multiplexer is placed before the input of ACC-AUX. Fig. 10(b) depicts the final double-precision MAC circuit.

2) Single-Precision MAC Circuit: The single-precision MAC circuit must accommodate the complex single-precision

Authorized licensed use limited to: National Taiwan University. Downloaded on February 26, 2009 at 23:21 from IEEE Xplore. Restrictions apply.

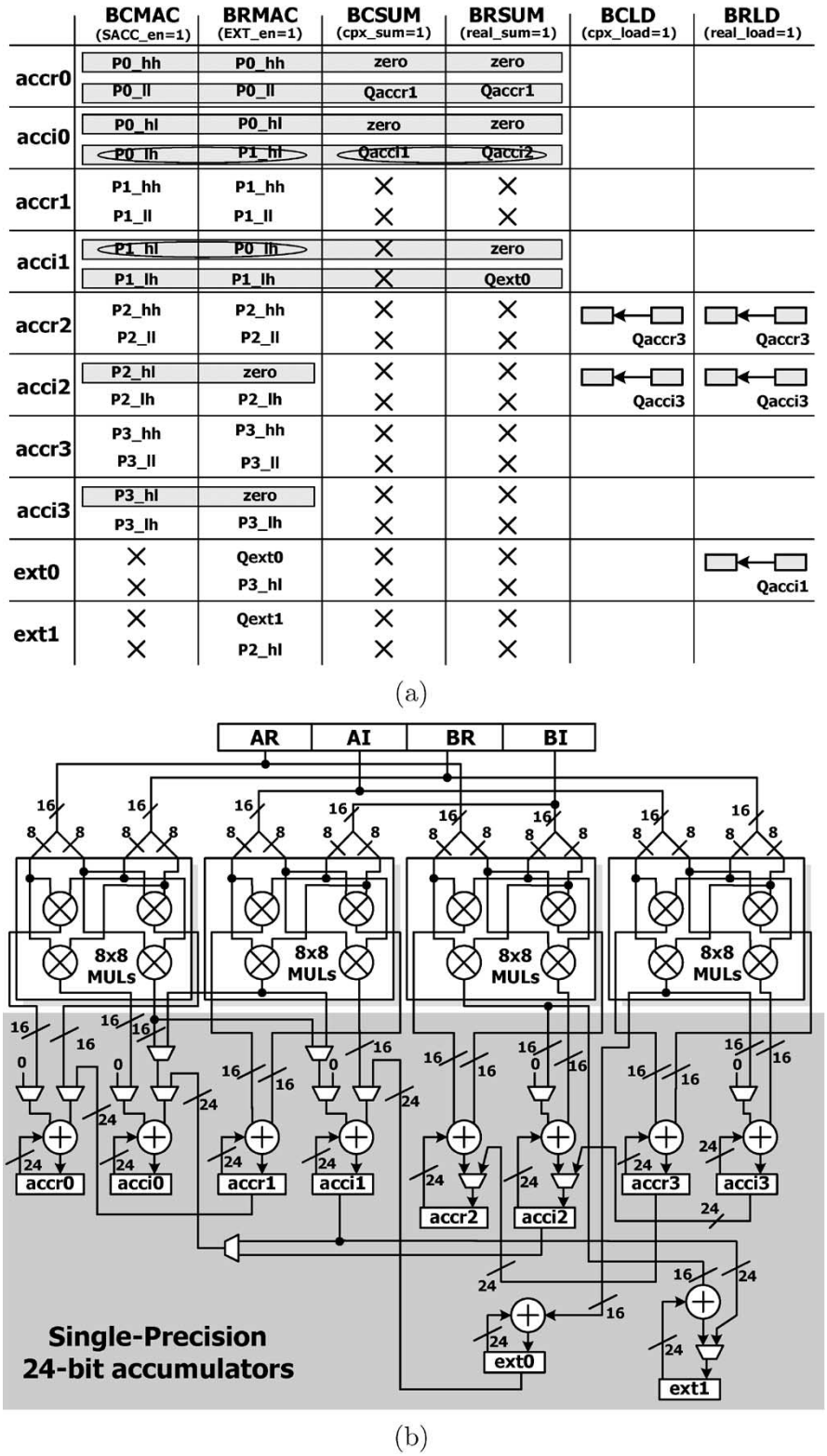


Fig. 11. (a) Input signals to be summed by the ten real accumulators of the single-precision MAC circuit in different instructions. (b) Proposed single-precision MAC circuit.

MAC operation and the real single-precision MAC operation. In the complex single-precision MAC mode, referring to Fig. 6, accumulators $acc0$ and $acc1$ are used to compute the partially accumulated values using the single-precision complex-valued MAC instruction (BCMAC). After each iteration, the contents in the two accumulators are then added to generate the output $Y(n)$ of the current iteration. Before the next iteration, the complex accumulator $acc3$ must be moved to the accumulator $acc2$. A routine that compute the K-tap FIR convolution on complex single-precision signals in $(K/2)+7$ clock cycles is

Authorized licensed use limited to: National Taiwan University. Downloaded on February 26, 2009 at 23:21 from IEEE Xplore. Restrictions apply.

TABLE II
COMPLEX MAC INSTRUCTIONS

Instruction	Description
MUL	Real Multiplication
CMUL	Complex Multiplication
CADD	Complex Addition
CSUB	Complex Subtraction
CCON	Complex Conjugate
WCMAC	Double-Precision Complex MAC
WRMAC	Double-Precision Real MAC
WRLD	ACC Load for Double Real MAC
BCMAC	Single-Precision Complex MAC
BCSUM	ACC Sum for Single Complex MAC
BCLD	ACC Load for Single Complex MAC
BRMAC	Single-Precision Real MAC
BRSUM	ACC Sum for Single Real MAC
BRLD	ACC Load for Single Real MAC
ACS	Add-Compare-Select
SQDSTA	Squared Distance Accumulation
SQDST	Squared Distance

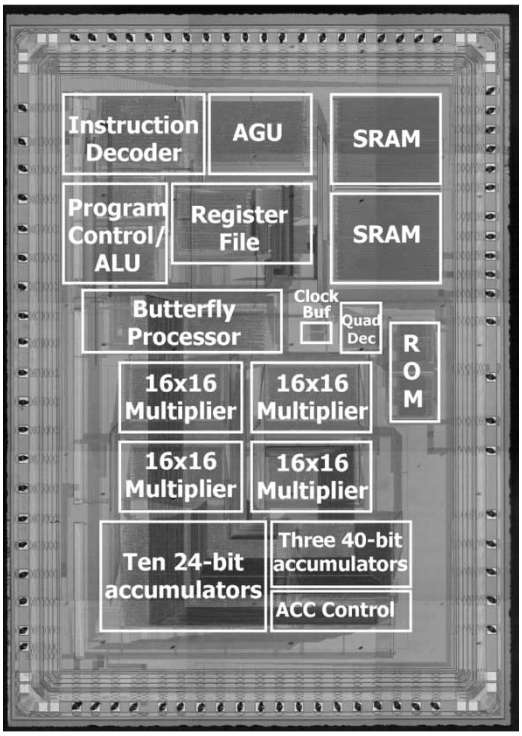


Fig. 12. Chip microphotograph.

listed below. Note that two output samples are computed on average, thus a speed-up factor of four can be achieved.

- Single-precision complex-valued MAC instruction:

$$
\begin{aligned}
&\text{LDSR } p\_addr\\
&\text{LDER } p\_addr\\
&\text{LDCX } \#(K/2)\\
p\_addr:\ &\text{BCMAC M[IR0++2] M[IR1++2]}\\
&\text{BCSUM}\\
&\text{STAC0 M[IR2++]}\\
&\text{STAC2 M[IR2++]}\\
&\text{BCLD.}
\end{aligned}
$$

In the real single-precision MAC mode, three new instructions, BRMAC, BRSUM, and BRLD, are similarly defined. One iteration of a K-tap real single-precision convolution takes only $(K/4)+8$ clock cycles. Since four output samples are computed in each iteration, a speed-up of 16 can thus be obtained.

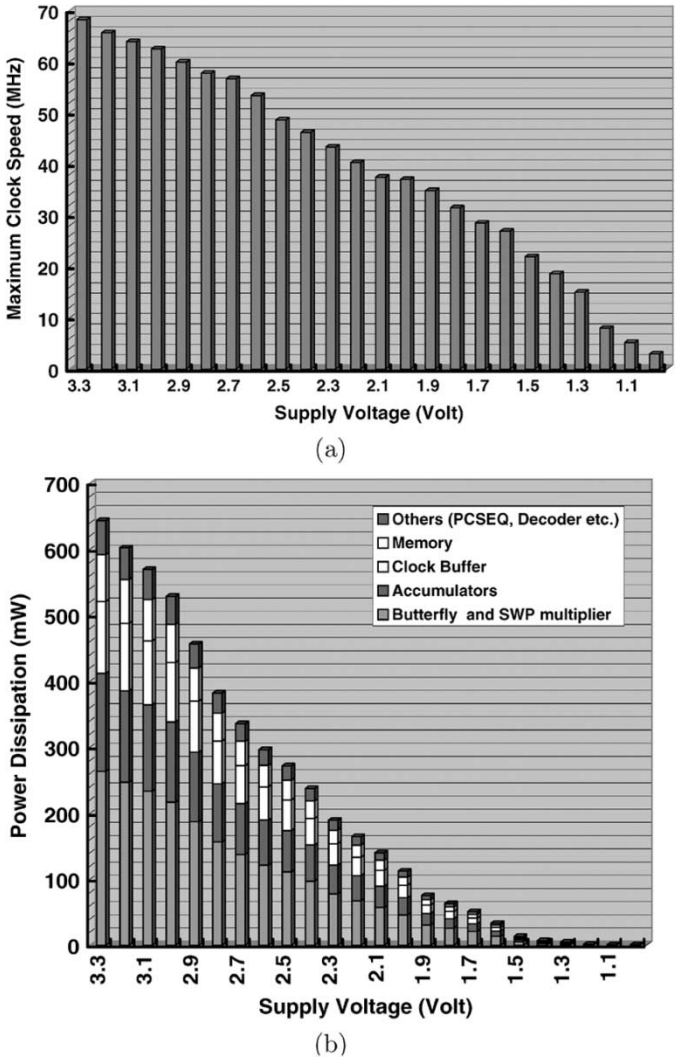


Fig. 13. (a) Clock speed versus supply voltage. (b) Power dissipation versus supply voltage.

Following similar analysis in the double-precision case, operands to be summed by the ten real accumulators in the single-precision MAC circuit during different instructions are listed in Fig. 11(a). Note that in addition to the four complex accumulators ($acc0$, $acc1$, $acc2$, and $acc3$), there are two extra real accumulators: $ext0$ and $ext1$. Each shadowed rectangle or ellipsoid represents one multiplexer. The overall single-precision MAC circuit is depicted in Fig. 11(b). Finally, a list of CMAC-related instructions is given in Table II.

V. EXPERIMENTAL RESULTS

The proposed DSP was fabricated using a 0.35-μm n-well one-poly four-metal CMOS technology. The chip occupies 4200×6000 μm^2 silicon with 3200×5000 μm^2 active area. Fig. 12 shows the microphotograph of the chip. The functionality of the DSP chip was verified and its performance measured using a test station. We developed a simple software to compile the instructions into machine code. The looping

Authorized licensed use limited to: National Taiwan University. Downloaded on February 26, 2009 at 23:21 from IEEE Xplore. Restrictions apply.

TABLE III
COMPARISON OF THE PROPOSED DSP AND SEVERAL MODERN DSPS

	TI C55x [4]	LODE [5]	DSP16000 [6]	MDSP-II [8]	CDSP [9]	Hiroyuki JSSC98 [19]
Complex MAC	No	No	No	Yes	Yes	No
Sub-Word Parallel MAC	No	No	No	No	Yes	No
Memory Alignment During Access	N/A	N/A	N/A	N/A	No	N/A
Double-precision complex MAC	(2 cycles)	(2 cycles)	(2 cycles)	2 cycles	(4 cycles)	(2 cycles)
Double-precision real MAC	1/2 cycle	1/2 cycle	1/2 cycle	1 cycle	1 cycle	1/4 cycle
Single-precision complex MAC	(2 cycles)	(2 cycles)	(2 cycles)	(2 cycles)	1 cycle	(2 cycles)
Single-precision real MAC	(1/2 cycle)	(1/2 cycle)	(1/2 cycle)	(1 cycle)	1/2 cycle	1/4 cycle
Butterfly	(6 cycles)	(4 cycles)	(4 cycles)	(5 cycles)	(8 cycles)	(3 cycles)
ACS operation	(4 cycles)	2 cycles	2 cycles	2 cycles	1/2 cycle	(2 cycles)
Square Distance	1 cycle	1 cycle	1 cycle	1 cycle	(2 cycles)	(1/2 cycle)
Technology	-	-	0.35 μm	0.6 μm 1P3M	0.35 μm 1P4M	0.25 μm 1P3M
Area	-	-	20.0 mm^2	9.7×9.8 mm^2	5.6×5.6 mm^2	9.2×9.2 mm^2
Clock Speed	-	-	160MHz(@3.3V)	50MHz(@5.0V)	60MHz(@3.3V)	50MHz(@1.5V)
MAC Performance (MAC/sec)	-	-	320M (SIMD)	100M (SIMD)	240M (SIMD)	200M (4-way MIMD)
Power Dissipation	-	-	160mW	-	165mW	110mW
MAC/sec/mW	-	-	0.7M	-	1.46M	1.8M

	Hinrichs JSSC2000 [20]	Ackland JSSC2000 [21]	Olofsson ISSCC2002[22]	Agarwala ISSCC2002[23]	This work High-Speed	This work Low-Power
Complex MAC	No	No	No	No	Yes	
Sub-Word Parallel MAC	No	Yes	No	Yes	Yes	
Memory Alignment During Access	N/A	No	N/A	No	Yes	
Double-precision complex MAC	(2 cycles)	(2 cycles)	(4 cycles)	(2cycles)	1 cycle	
Double-precision real MAC	1/4 cycle	1/4 cycle	1 cycle	(1/4 cycle)	1/4 cycle	
Single-precision complex MAC	(2 cycles)	(1 cycle)	1/4 cycles	(1/4 cycle)	1/4 cycle	
Single-precision real MAC	1/4 cycle	1/8 cycle	1/8 cycles	(1/8 cycle)	1/16 cycle	
Butterfly	(3 cycles)	(3 cycles)	(8 cycles)	(3 cycles)	1 cycle	
ACS operation	(2 cycles)	(4 cycles)	1/4 cycle	(2 cycles)	1 cycle	
Square Distance	(1/2 cycle)	(1/2 cycle)	(2 cycles)	(2 cycles)	1/2 cycle	
Technology	0.5 μm 3M	0.25 μm 1P4M	0.13 μm 8M	0.13 μm 6M	0.35 μm 1P4M	
Area	13.8×13.6 mm^2	200 mm^2	10×10 mm^2	8.5×8.5 mm^2	4.2×6.0 mm^2	
Clock Speed	40MHz(@3.3V)	100MHz(@3.3V)	250M(@1V)	600MHz(@1.2V)	68MHz(@3.3V)	22MHz(@1.5V)
MAC Performance (MAC/sec)	380M (SIMD)	1.6G (4-way MIMD)	4.32G (8-way MIMD)	4.8G (8-way VLIW)	1.1G (SIMD)	352M
Power Dissipation	650mW	4W	1W	718mW	644mW	14mW
MAC/sec/mW	0.6M	0.4M	4.32M	3.3 M	1.69M	25M

MAC instructions are executed to measure the maximum clock speed and power dissipation of the DSP chip. Fig. 13(a) shows the maximum clock speeds at various supply voltages. The chip can operate up to 68, 40, and 22 MHz at 3.3, 2.2, and 1.5 V, respectively. Fig. 13(b) plots the power dissipation at different supply voltages. The chip consumes 644, 165, and 14 mW at 3.3, 2.2, and 1.5 V, respectively. The DSP chip operates in two modes, high-speed mode and low-power mode. In the high-speed mode, the DSP chip can operate at its maximum frequency of 68 MHz at 3.3 V, achieving 1.1 G MAC operations per second and 68 M butterfly operations per second. In the low-power mode, the DSP chip dissipates 14 mW power running at 22 MHz from a 1.5-V power supply. The 100% hardware utilization in all four MAC configurations makes the chip capable of executing 25 M MAC operations per second per milliwatt.

The performance of the proposed chip is listed and compared with several recently published DSPs in Table III. The parenthesized entries represent the number of cycles required to compute the operations for which the chip in question was not specifically designed. As is clear from the table, the proposed DSP outperforms all other DSPs in almost all categories due to its special datapath architecture and flexible configuration. In the high-speed mode, the proposed DSP chip has a computational capability of 1.1 G MAC operations per second, outperforming all other DSP chips except the DSP chips in [21]–[23].

Authorized licensed use limited to: National Taiwan University. Downloaded on February 26, 2009 at 23:21 from IEEE Xplore. Restrictions apply.

However, these DSPs have MIMD or VLIW architecture with four-way or eight-way parallelism. If we compare the performance based on only one processing core, the proposed DSP chip has the best performance. On the other hand, these chips dissipate more power than the proposed chip. In terms of power efficiency (MAC operations per second per milliwatt), the proposed chip, in its low-power mode, is better than most other DSP chips. Furthermore, if the proposed DSP architecture were implemented in a more advanced technology as the three DSPs, much better performance can be attained. Therefore, we believe that the chip is suitable for use in portable devices with a high computing capacity, as is the case for the next-generation wireless communication.

VI. Conclusion

This work presents a programmable sub-word parallel digital signal processor specially designed for communication and multimedia devices. The processor supports basic butterfly operations, single/double-precision and real/complex-valued MAC, sum of squared difference computation, and ACS operations. These operations can compute FFT, Viterbi decoding, and other functions more efficiently and do so with 100% hardware utilization in filter and convolution processing. The chip is fabricated in a 0.35-μm n-well one-ploy four-metal CMOS technology. The DSP chip can achieve 1.1 G MAC/second at 3.3 V in the high-speed mode and 25 M MAC/s/mW at 1.5 V in the low-power mode. Therefore, this processor can significantly improve the computing performance of communication signal processing and thus lay a solid foundation for future wireless communication systems.

Acknowledgment

The authors would like to thank the Chip Implementation Center of the National Science Council of Taiwan for chip fabrication. They also thank the anonymous reviewers for their valuable suggestions that greatly improved this paper.

References

[1] J. Mitola, "The software radio architecture," *IEEE Commun. Mag.*, vol. 33, pp. 26–38, May 1995.
[2] H. Nie, A. K. Salkintzis, and P. T. Mathiopoulos, "ADC and DSP challenges in the development of software radio base stations," *IEEE Commun. Mag.*, vol. 37, pp. 47–55, Aug. 1999.
[3] (2001, Mar.) TMS32C54x DSP Reference Set Volume 3: Algebraic Instruction Set. Texas Instruments Inc., Dallas, TX. [Online]. Available: http://www.ti.com/
[4] (2000, Feb.) TMS32C55x Technical Overview. Texas Instruments Inc., Dallas, TX. [Online]. Available: http://www.ti.com/
[5] I. Verbauwhede, M. Touriguian, K. Gupta, J. Muwafi, K. Yick, and G. Fettweis, "A low power DSP engine for wireless communications," in *Proc. Workshop VLSI Signal Processing*, vol. IX, 1996, pp. 471–480.
[6] M. Alidina, G. Burns, C. Holmqvist, E. Morgan, D. Rhodes, S. Simanapalli, and M. Thierbach, "DSP16000: A high performance, low power dual-MAC DSP core for communication applications," in *Proc. IEEE Custom Integrated Circuits Conf.*, May 1998, pp. 119–122.
[7] T. S. Rappaport, *Wireless Communications, Principles and Practice*, 2nd ed. Englewood Cliffs, NJ: Prentice Hall, 2002.
[8] B. W. Kim, J. H. Yang, C. S. Hwang, Y. S. Kwon, K. M. Lee, I. H. Kim, Y. H. Lee, and C. M. Kyung, "MDSP-II: A 16-bit DSP with mobile communication accelerator," *IEEE J. Solid-State Circuits*, vol. 34, pp. 397–403, Mar. 1999.
[9] C. K. Chen, P. C. Tseng, Y. C. Chang, and L. G. Chen, "A digital signal processor with programmable correlator array archetecture for third generation communication system," *IEEE Trans. Circuits Syst. II*, vol. 48, pp. 1110–1120, Dec. 2001.
[10] J. Fridman, "Sub-word parallelism in digital signal processing," *IEEE Signal Processing Mag.*, vol. 17, pp. 27–35, Mar. 2000.
[11] Y. H. Huang and T. D. Chiueh, "A sub-word parallel digital signal processor for wireless communication systems," in *Proc. IEEE Asia-Pacific Conf. ASIC*, Aug. 2002, pp. 287–290.
[12] C. F. Hsu, Y. H. Huang, and T. D. Chiueh, "Design of an OFDM receiver for high-speed wireless LAN," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 4, May 2001, pp. 558–561.
[13] H.-P. Ma, S. H. Hsu, and T.-D. Chiueh, "An uplink baseband receiver architecture and FPGA implementation for W-CDMA systems," in *Proc. 6th Asia-Pacific Conf. Communications*, Seoul, Korea, Nov. 2000, pp. 774–778.
[14] R. B. Lee, "Multimedia extensions for general-purpose processors," in *Proc. IEEE Workshop Signal Processing Systems (SiPS)*, Nov. 1997, pp. 9–23.
[15] E. O. Brigham, *The Fast Fourier Transform and its Application*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
[16] A. M. Despain, "Very fast Fourier transform algorithms hardware for implementation," *IEEE Trans. Computers*, vol. C-28, pp. 333–341, May 1979.
[17] S. He and M. Torkelson, "Designing pipeline FFT processor for OFDM (de)modulation," in *Proc. Int. Symp. Signals, Systems, and Electronics*, Sept./Oct. 1998, pp. 257–262.
[18] M. A. Richards, "On hardware implementation of the split-radix FFT," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP–36, pp. 1575–1581, Oct. 1988.
[19] H. Igura, Y. Naito, K. Kazama, I. Kuroda, M. Motomura, and M. Yamashina, "An 800-MOPS, 110-mW, 1.5-V, parallel DSP for mobile multimedia processing," *IEEE J. Solid-State Circuits*, vol. 33, pp. 1820–1828, Nov. 1998.
[20] W. Hinrichs, J. P. Wittenburg, H. Lieske, H. Kloos, M. Ohmacht, and P. Pirsch, "A 1.3-GOPS parallel DSP for high-performance image-processing applications," *IEEE J. Solid-State Circuits*, vol. 35, pp. 946–952, July 2000.
[21] B. Ackland *et al.*, "A single-chip 1.6-billion 16 b MAC/s multiprocessor DSP," *IEEE J. Solid-State Circuits*, vol. 35, pp. 412–424, Mar. 2000.
[22] A. Olofsson and F. Lange, "A 4.32 GOPS 1 W general-purpose DSP with an enhanced instruction set for wireless communication," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, vol. 2, 2002, pp. 36–389.
[23] S. Agarwala *et al.*, "A 600 MHz VLIW DSP," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, vol. 2, 2002, pp. 38–389.

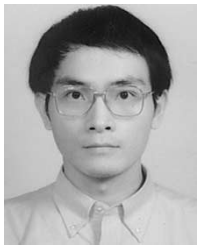
Yuan-Hao Huang (S'98–M'02) was born in Taiwan in 1973. He received the B.S. and Ph.D. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 1995 and 2001, respectively.

He is currently a Member of Technical Staff at VXIS Technology Corporation, Hsin-Chu, Taiwan. His primary research interests include speech and audio coding, digital signal processing system design and VLSI design of speech, audio, and telecommunication systems.

Hsi-Pin Ma (M'98) was born in Nantou, Taiwan, in 1973. He received the B.S. and Ph.D. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 1995 and 2002, respectively.

Since 2003, he has been with the Department of Electrical Engineering, National TsingHua University, Hsin-Chu, Taiwan, R.O.C., where he is currently an Assistant Professor. His primary research interests include baseband signal processing of communication systems and related VLSI design.

Authorized licensed use limited to: National Taiwan University. Downloaded on February 26, 2009 at 23:21 from IEEE Xplore. Restrictions apply.



Ming-Luen Liou was born in Taipei on February 15, 1973. He received the B.S.E.E. degree from the National Taiwan University, Taipei, Taiwan, R.O.C., in 1994. He was in military service from 1996 to 1998. He received the Ph.D. degree in electrical engineering from National Taiwan University, Taipei, in 2001.

He is currently a Member of Technical Staff with MediaTek Inc., Hsin-Chu, Taiwan. His research interests include baseband signal processing algorithm and low-power digital ASIC implementations of communication systems.

Tzi-Dar Chiueh (S'87–M'90–SM'03) was born in Taipei on July 3, 1960. He received the B.S.E.E. degree from the National Taiwan University, Taipei, Taiwan, R.O.C., in 1983 and the M.S. and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena, in 1986 and 1989, respectively.

Since 1989, he has been with the Department of Electrical Engineering, National Taiwan University, where he is currently a Professor. His research interests include IC design for digital communication systems and analog neuromorphic systems.

Authorized licensed use limited to: National Taiwan University. Downloaded on February 26, 2009 at 23:21 from IEEE Xplore. Restrictions apply.