INTERNATIONAL JOURNAL OF RESEARCH
In Advanced Engineering Technologies

Volume 4, Issue 1 APR 2015

DESIGN AND IMPLEMENTATION OF FLEXIBLE MULTIPLIER USING RAZOR BASED DYNAMIC VOLTAGE SCALING FOR FILTER DESIGN

[1]L. Rajya Lakshmi, VLSI System Design, PG scholar,
[2]A. Balachandra Reddy Asst. Professor, ECE Department,
[1] rajyalakshmi453@gmail.com,
[2] balutest@gmail.com

Abstract: In this paper, we present flexible multi-precision multiplier that combined variable precision, parallel processing (PP), razor based dynamic voltage scaling (DVS), and dedicated MP operand scheduling to provide optimum performance for variety of operating conditions. All of the building blocks of proposed flexible multiplier can either work as independent small precision multiplier or parallel to perform higher-precision multiplier. While still maintain full through-put, the dynamic voltage and frequency scaling management unit configures the multiplier to operate at the proper precision and frequency.

Adapting to the run-time workload for targeted application, that flexible multiplier can used to design IIR filter for DSP application. Razor flip-flops together with a dithering voltage unit then configure the multiplier to achieve the lowest power consumption. The single-switch dithering voltage unit and razor flip-flops help to reduce the voltage margin and over head typically associated to DVS to lowest level. Finally, the proposed high speed flexible multiplier can further benefits from an operand scheduler that rearranges the input data, hence determine the optimum voltage and frequency operating conditions for minimum for power consumption. As well suited to design IIR filter design for DSP application.

Keywords- Computer arithmetic, dynamic voltage scaling, low power design, multi-precision multiplier.

I.Introduction

Consumer demand for increasingly portable yet high performance multimedia and communication products impose stringent constraints on the power consumption of individual internal components [1].of these multiplier perform one of the most frequently encountered arithmetic operation in digital signal processor(DSPs)[2].Multiplier is typically designed for a fixed maximum word-length to suit the worst case scenario. However, the real effective word-lengths of an application vary dramatically. The use of a non-proper word length may cause performance degradation or inefficient usage of the hardware resources. In addition, the minimization of the multiplier power budget requires the estimation of the optimal operating point including clock frequencies, supply voltage, and threshold voltage [1].In most VLSI system designs, the supply voltage is also selected based on the worst case scenario. In order to achieve an optimal power/performance ratio, a variable precision data path solution is needed to cater for various types of applications. Dynamic Voltage Scaling (DVS) can be used to match the circuit's real working load and further reduce the power consumption. Given their fairly complex structure and interconnections, multiplier can exhibits a large number of unbalanced paths, resulting in substantial glitch generation and propagation [8].this spurious switching activity can be

mitigated by balancing internal paths through a combination of architectural and transistor-level Optimization techniques. In addition to equalizing internal path delays, dynamic power reduction can also be achieved by monitoring the effective dynamic range of input operands so as to disable unused section of multiplier [6].therefore, an 8-bit multiplication computed on a 32-bit booth multiplier would result in unnecessary switching activity and power loss. Several works investigated this word-length optimization. [1], [2] proposed an ensemble of multiplier of different precision, with each pair of incoming operands is routed to the smallest multiplier that cancompute the result to take advantage of the lower energy consumption of the smaller circuit. This ensemble of point systems is reported to consume the least power but this came at cost increased chip area given the used ensemble structure. To address this issue,[3],[5] proposed to share and reuse some functional modules within the ensemble. In [3], an 8-bit multiplier is reused for the 16-bit multiplication adding scalability with out large area penalty. Reference [5] extended this method by implementing Pipelining to further improve the multiplier's performance. A more flexible approach is proposed in [15]. Combining multi-precision (MP) with dynamic voltage scaling (DVS) can provide a dramatic reduction in power consumption by adjusting the supply voltage according to –Circuit's run-time workload rather than fixing it to cater for the worst case scenario [2].when adjusting the voltage, the actual performance of multiplier running under scaled voltage has to be characterized to guarantee a fail-safe operation. Conventional DVS technique consist mainly of lookup table (LUT) the LUT approach tune the supply voltage according to predefined voltage frequency relationship stored in a LUT, which is formed worst case condition(process variation, power supply voltage droops, noise many more) therefore, large margin are necessarily added, which in turn necessary decrease effectiveness of DVS technique. Therefore, voltage could be scaled to the extent that the replica fails to meet the timing. However, safety margins are still needed to compensate for the intradie delay mismatch and address fast-changing transient effects [24].the aforementioned limitation of conventional DVS techniques motivated recent research efforts into error-tolerant DVS approaches [24]-[27], which can run-time operate the circuit even at a voltage level at which timing error occur, A recovery mechanism is then applied to detect and correct data. Because completely remove safety margins, error- tolerant DVS techniques can further aggressively reduce power consumption. In this paper, we propose a low power reconfigurable multiplier architecture that combined MP with an error-tolerant DVS approach based on razor flip-flops [25], the main contributions of this paper can be summarized follows. Silicon area is optimized by apply reduction technique that replace a multiplier by adders/subtractors. a silicon implementation of this multi-precision multiplier integrating error tolerant razor based dynamic DVS approach. The run–time adaption to actual workload condition and allow minimum supply voltage and frequency, while meet throughput operation. A dedicated operand scheduler that rearrange operation on input operands so as to reduce the number of transitions of the supply voltage and, in turn, minimize the overall power consumption of the flexible multi-precision multiplier.

A multiplier is an important part of digital signal processing systems, like frequency domain filtering (FIR and IIR), frequency-time transformations (FFT), Correlation, Digital Image processing etc. Multipliers have large area, long latency and consume considerable power. While many previous works focused on implementing high-speed multipliers, recently there have been many attempts to reduce power consumption [3]. This is due to the increased demand for portable multimedia applications, which require low power consumption as well as high speed. There is wide range of multipliers. Based on the way the data is processed, they are classified as serial, parallel and serial-parallel multipliers as shown in figure 1.

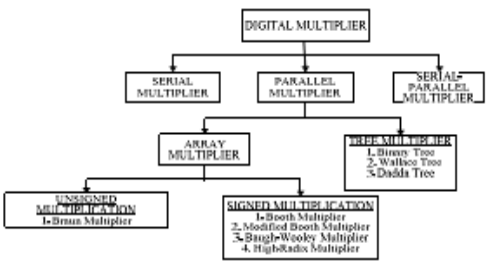


Figure 1. Classification of multipliers

In parallel multipliers, there are two main classifications. They are array and tree multipliers. Tree multipliers add as many partial products in parallel as possible and therefore, are very high performance architecture. Unfortunately tree multipliers are very irregular, hard to layout and hence large. Array multipliers, on the other hand, are very regular, small in size but suffer in latency and propagation delay. Booth multiplier is used for signed binary numbers. They are also called radix-2 multiplier. Their main advantage is that it involves no correlation cycles for signed terms. But they become inefficient for alternate zeros and ones as it involves large numbers of adders and subtractors, his result in area and speed limitation. The problem is overcome with modified booth multiplier (MBM)
or radix-4 multiplier which reduces the partial products by 50%. Thus it improves speed, reduce power consumption and also save multiplier layout area. MBM also has a regular structure.

II. System Overview and Operation

The proposed MP multiplier system (Fig. 2) comprises five different modules that are as follows:
1) The MP multiplier;
2)The input operands scheduler (IOS) whose function is to reorder the input data stream into a buffer, hence to reduce the required power supply voltage transitions;
3)The frequency scaling unit implemented using a voltage controlled oscillator (VCO). Its function is to generate the required operating frequency of the multiplier;
4)The voltage scaling unit (VSU) implemented using a voltage dithering technique to limit silicon area overhead. Its function is to dynamically generate the supply voltageso as to minimize power consumption;
5)The dynamic voltage/frequency management unit(VFMU) that receives the user requirements (e.g., throughput).

The VFMU sends control signals to the VSU and FSU to generate the required power supply voltage and clock frequency for the MP multiplier.
The MP multiplier is responsible for all computations. It is equipped with razor flip-flops that can report timing errors associated to insufficiently high voltage supply levels.The operation principle is as follows. Initially, the multiplier operates at a standard supply voltage of 3.3 V. If the razor flipflops of the multiplier do not report any errors, this means that the supply voltage can be reduced. This is achieved through the VFMU, which sends control signals to the VSU, hence to lower the supply voltage level. When the feedback provided by the razor flip-flops indicates timing errors, the scaling of the power supply is stopped.

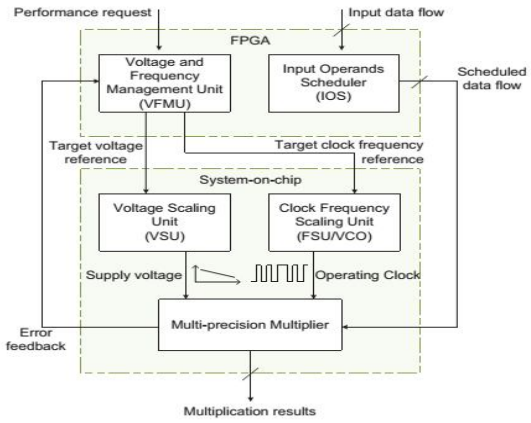


Fig. 2. Overall multiplier system architecture.

The proposed multiplier (Fig. 3) not only combines MP and DVS but also parallel processing (PP). Our multiplier comprises 8×8 bit reconfigurable multipliers. These building blocks can either work as nine independent multipliers or work in parallel to perform one, two or three 16×16 bit multiplications or a single-32×32 bit operation. PP can be used to increase the throughput or reduce the supply voltage level for low power operation.

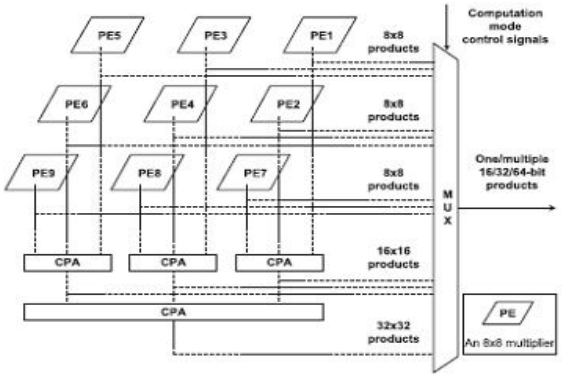


Fig. 3.Possible configuration modes of existing

MP multiplier

Proposed Block for Multi-Precision Multiplier

The following block diagram for proposed multiplier.

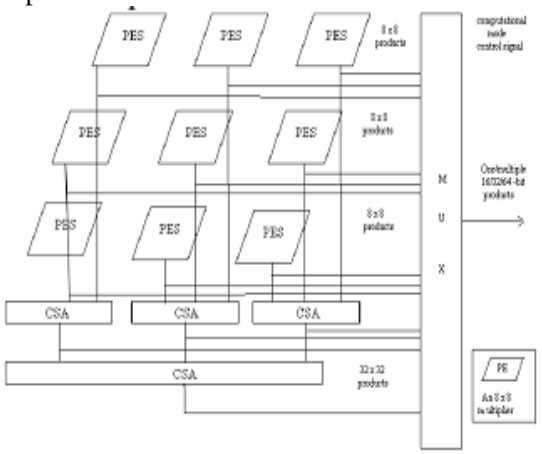


Fig. 3.Possible configuration modes of proposed

MP multiplier

The proposed multiplier (fig. 3) not only combines MP and DVS but also parallel processing (pp). This multiplier comprises 8 × 8 bit reconfigurable multiplier.

These building blocks can either work as nine independent multiplier or work in parallel to perform one, two or three 16 × 16 bit multiplication or a single-32×32bit operation.

PP can be used to increase the throughput or reduce the supply voltage levels for low power operation.

Mainly this multiplier having one drawback in adder circuit, here used carry propagate adder in which adder have a high propagation delay. So that purpose to proposed above fig.3

Carry Select Adder

Carry select adder (CSLA) is one of the fastest adders used in many data-processing processors to perform fast arithmetic functions. From the structure of the CSLA, it is clear that there is scope for reducing the area and power consumption in the CSLA.

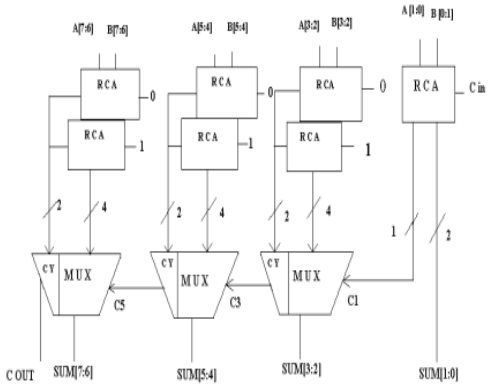


Fig.4. Basic block for 8bit carry select adder

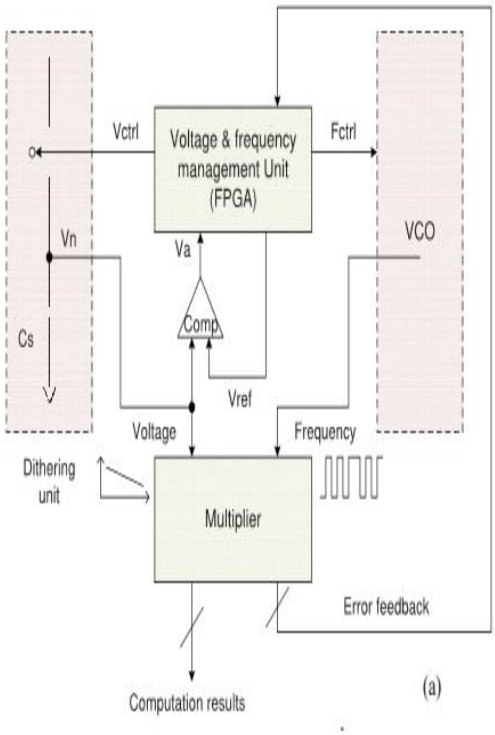
The CSLA is used in many computational systems to alleviate the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum. However, the CSLA is not area efficient because it uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and carry by considering carry input and, then the final sum and carry are selected by the multiplexers (mux). The basic idea of this work is to use Binary to Excess-1 Converter (BEC) instead of RCA with in the regular CSLA to achieve lower area and power consumption. Fig. 4 illustrates how the basic function of the CSLA is obtained by using the 4-bit BEC together with the mux. One input of the 8:4 mux gets as it input (B3, B2, B1, and B0) and another input of the mux is the BEC output. This produces the two possible partial results in parallel and the mux isused to select either the BEC output or the direct inputs according to the control signal Cin. The importance of the BEC logic stems from the large silicon area reduction when the CSLA with large number of bits are designed.

Dynamic voltage and Frequency Scaling Management

1. Dynamic Voltage Scaling (DVS) unit

In this implementation DVS unit shows a dynamic power supply and a VCO are employed to achieve real-time dynamic voltage and scaling can be achieved when using voltage dithering, which exhibits faster response time than conventional voltage regulator. Voltage dithering uses power switches to connect different supply voltage to the

load, depending on the time slots. Therefore, an intermediate average voltage is achieved.



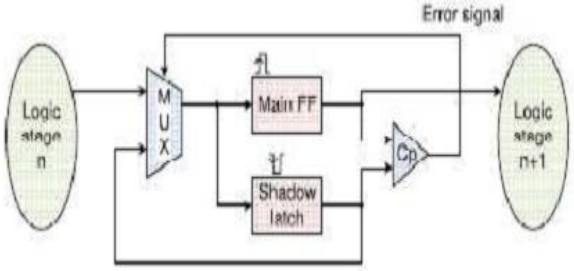
Proposed single-header voltage dithering unit

B. Dynamic Frequency Scaling Unit

In the proposed 32 ×32 bit MP multiplier, dynamic frequency tuning is used to meet throughput requirements. It is based on a VCO implemented as a seven-stage current starved ring oscillator. The VCO output frequency can be tuned from 5 to 50 MHz using four control bits (5 MHz/step).This frequency range is selected to meet the requirements of general purpose DSP applications. The reported multiplier can operate as a 32-bit multiplier or as nine independent 8-bit multipliers. For the chosen 5–50 MHz operating range, our multiplier boasts up to 9 ×50 =450 MIPS. The simulated power consumption for the VCO ranges from 85 (5 MHz) to 149μW (50 MHz), which is negligible compared with the power consumed by the multiplier. Fig. 7 shows experimental measurements showing the transient response for the worst case frequency switching (from 50 to 5 MHz). Clock frequency can settle within one clock cycle as required.

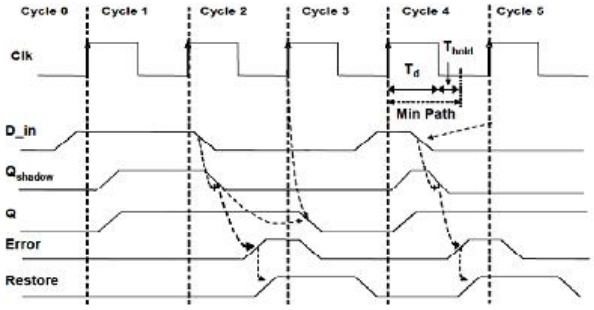
Implementation Of Razor Flip-Flops

Although the worst case paths are very rarely exercised, traditional DVS approaches still maintain relatively large safety margins to ensure reliable circuit operation, result in excessive power dissipated. The razor technology is breakthrough work, which eliminates the safety margin by achieving variable tolerance through in-situ error detection and correction ability [25]. This approach is based on a razor flip-flops, which detects and correct the delay error by double sampling.



Conceptual view of razor flip-flops

The razor flip-flops are constructed out of a standard positive Edge triggered flip-flops (DFF) augmented with a shadow latch which samples at the negative clock edge. Thus, the input data is given additional time, equal to the duration of the positive clock phase, to settle down to its correct state before being sampled by the shadow latch. In order to ensure the shadow latch always capture the correct data, the minimum allowable supply voltage needs to be constrained during design time such that step-up time at the shadow latch. A comparator flags a timing error when it detects a discrepancy between the speculative sampled at the main flip-flops and the correct data sampled at the shadow latch.



Timing diagram for razor flip-flops

An error correction mechanism, based on global clock gating, is implemented in the proposed multiplier [25]. In this correction scheme, error and clock signals are used to determine when the entire pipeline needs to be stalled for a singleclock cycle. Fig. 1 shows that a

global error signal is fed to the VFMU so as to alert the controlling unit whenever the current operating voltage is lower than necessary. The VFMU will then increase the voltage reference. This will in turn result in the VSU generating a new supply voltage level based on the new target voltage reference. When an error occurs, results can be recomputed at any pipeline stage using the corresponding input of the shadow latch. Therefore, the correct values can be forwarded to the corresponding next stages. Given that all stages can carry out these recomputations in parallel, the adopted global clock gating can tolerate any number of errors within a given clock cycle [25]. After one clock cycle, normal pipeline operation can resume. The actual implementation of razor flip-flops requires careful design to meet timing constraints and avoid system failure. For example, the use of a delayed clock for the shadow latch (Fig. 8) makes it possible for a short-path in the combinational logic to corrupt the data in the shadow latch [25]. This imposes a short-path delay constraint at the input of each razor flip-flop of our multiplier. To meet these constraints across all corners, we inserted delay buffers through all short paths found by Cadence
silicon-on-chip (SOC) Encounter and validated them through Prime Time. In addition, precautions are used to mitigate meta stability by inserting a meta stability detector at the output of each main flip-flop. The outputs of the meta stability detector and the error comparator (Fig. 8) are ORED to generate the error signal of individual razor flip-flops [25], [26]. These razor error signals are OR-ED together to form a global error signal used to ensure that all valid data in the shadow latches is restored into the main flip-flops before the next clock cycle.. This meta stability detector relies on skewed inverters, which require careful simulation through all process corners to ensure proper operation.
When implementing razor-based DVS, it is essential that the resulting power/delay overhead be kept to a minimum, hence not to severely limit the benefits brought by aggressive supply voltage scaling. In the case of our multiplier, only 13 out of a total of 144 flip-flops that is 9% of the flip-flops are found not to meet timing constraints under worst case level of the supply voltage (Table II). Therefore, only these 13 critical paths are equipped with razor flip-flops. These 13 near-critical paths are identified through Cadence SOC Encounter and validated using Prime Time. At a supply voltage of 3.3 V and operating frequency of 50 MHz, the razor flip-flop is found to consume 1.2 times more static/switching power (70/57μW) when no timing errors are detected. In the other case, it consumes 4.2 times more static/switching power (239/57μW). However, for a conservative activity factor of 1%, the power overhead due to razor flip-flops was estimated to be less than 2.3% of the nominal chip power because only 9% of the flip-flops were made razor flip-flops. Therefore, both the silicon area and power overheads associated to razor flip-flops are found to be negligible. In regard to therazor flip-flop's delay overhead, it is mainly because of the additional multiplexer at its input as well as the increased fan-out resulting from the introduction of comparator, metastability detector, andORgates at the output. At a supply voltage of 3.3 V and operating frequency of 50 MHz, delay overheads are found to be 1.20% and 3.58% for error-free and error-occurring cases, respectively. These delay overheads constitute a small penalty for the massive power reduction enabled by razor-based DVS.

Input Operand Scheduler Unit

A. Motivation and Operation Principle

Main motivation and operating principle of input operand scheduler that rearranges operations on input operands so as to reduces the number of transition of the supply voltage and, in turn, minimize the overall power consumption of the multiplier. Whose function is reordering the input data stream into a buffer, hence to reduce the required power supply voltage transitions. The multiplier provide three different precision modes (32×32bit, 16×16, 8×8-bits), the supply voltage would to transit dynamically between the minimum required voltage levels Vmin32, Vmin16, Vmin8, required for 32, 16, 8-bit operands, respectively, we propose an IOS that will perform following task:

1. Reorder the input data stream such that sameprecision operands are grouped together into a buffer.

2. Find the minimum supply voltages (Vmin32, Vmin16, Vmin8), and operating frequencies (f32,f16, f8) for three different-precision data grouped to minimize the overall power consumption while still meeting the specified throughput.

Algorithm for IOS

There are three different algorithms to reduce this overall power consumption, algorithm A, B and C each of these algorithms constitutes a different approach to process the mixed-precision data held in the operands buffer. The performance of each algorithm is evaluated using a mixed precision data set with the corresponding to each precision (8, 16, and 32-bit).

Algorithm A

The algorithm A states the level of voltage and frequency is varying for different (8, 16, 32) bit precision.

Algorithm-B

This algorithm removes all transitions of the power supply voltage by making Vmin32, Vmin16, and Vmin8 equal and adjusting f32, f16, and f8 such that the overall throughput is kept unchanged.

Algorithm-C

Only one modification has been considered for compare algorithm-A with algorithm-C is inversely changing the frequency and voltage is remaining same.

B. Problem Formulation

Given a random mixed-precision (32-, 16-, or 8-bit) input data stream and specified throughput Tp, our goal is to determine the voltages (Vmin32, Vmin16, Vmin8) and frequencies (f32, f16and f8) at which each

precision data group should be processed such that the total power consumption is minimized. In the following analysis, we consider the following four components of the total power consumption: 1) the resistive power loss Pdith_resistic_loss of the dithering unit; 2) the switching power loss Pdith_switching_loss of the dithering unit; 3) the dynamic power consumption Pcomputation associated to the multiplication computation; and 4) finally, Pcompu_overhead that corresponds to the power consumption of the latter computation when carried out at voltage levels higher than the nominal Vmin. The equations of the aforementioned four components of the total power consumption are given below

$$P_{\text{dith\_resistic\_loss}} = I_{\text{char}}^2 R_{\text{on}}$$

Where I_{char} is the charge current of the dithering unit, and Ron is the equivalent resistance of the dithering switch

$$P_{\text{dith\_switching\_loss}} = \frac{C_g V_{\text{dd}}{}^2 f}{N}$$

whereCg is the gate capacitance of the dithering switch, Vdd is the 3.3 V standard voltage, and Nis the number of input data patterns

$$P_{\text{computation}} = C_m V_{\text{min}}{}^2 f$$

whereCmis the effective capacitance of the multiplier, Vmin is the applied minimum supply voltage, and f is the applied operating frequency

$$P_{\text{compu\_overhead}} = \frac{\int P dt}{T} = \frac{\int C_m V^2 f dV}{T}$$

whereVis the dithering unit output, which fluctuates around Vmin,andT is the charge time period, which is inversely proportional to the operating frequency.

The overall power consumption is thus given by

$$P_{\text{overall}} = P_{\text{computation}} + P_{\text{compu\_overhead}} + P_{\text{dith\_resistic\_loss}} + P_{\text{dith\_switching\_loss}}.$$

In the following, we present three different algorithms to reduce this overall power consumption. Each of these algorithms constitutes a different approach to process the mixed-precision data held in the operands buffer . The performance of each algorithm is evaluated using a mixed precision data set of 120 000 randomly operands, with a third corresponding to each precision (8-, 16-, and 32-bit). In the following, the specified throughput Tp for the proposed 32 ×32 bit multiplier is 64 F(Mbits/s), where F is the multiplier's operating frequency.

C. Algorithm A

In the first algorithm, the multiplier throughput Tp=64 F is kept constant by fixing the operating frequencies (f32−, f16−, or f8) of each precision-data group (32-, 16-, or 8-bit) to

$$f_{32} = F,\ f_{16} = \frac{F}{2},\ f_8 = \frac{F}{4}$$

whereF is the multiplier's operating frequency. This is because the throughput in 8×8 bit multiplication mode is four times that of the 32 ×32 bit multiplication mode and double that of the 16 ×16 bit multiplication mode,

as a result of the multiplier PP. The minimum supply voltage (Vmin32, Vmin16 or Vmin8) associated to each operating frequency (f32, f16 or f8) is determined through a Vmin–f LUT. Algorithm A shows its limitations when 32-bit operands are processed initially. As shown in Fig. 14, once all N32operands of the data block are processed, the supply voltage (Vn) needs to decrease rapidly from point A (Vmin32) to point B (Vmin16) at which all N16 16-bit operands of the data block should be processed. If N16 is too small, most 16-bit operands will be actually processed in Sections A and B, that is at a voltage possibly much higher than the minimalVmin16level. Similarly 8-bit operands of the data block could be processed in Sections C and D, B-C, or even A-B for the worst case. This contributes to increasing Pcompu_overhead. The overall power performance of

algorithm A is shown in Table IV. Compared with the fixedwidth 32×32 bit standard multiplier (32×32 bit mode must be chosen given that a third of operands are 32-bit), 77.7% total power reduction is achieved with a total silicon area overhead of only 11.1%, when considering DVS, razor, RAM, and dedicated circuitry for scheduling algorithm A.

D. Algorithm B

This algorithm removes all transitions of the power supply voltage by makingVmin32, Vmin16,andVmin8equal and adjusting f32, f16,andf8 such that the overall throughput is kept unchanged. We thus need to have the following:

$$\frac{64N_{32}+128N_{16}+256N_8}{\frac{N_{32}}{f_{32}}+\frac{N_{16}}{f_{16}}+\frac{N_8}{f_8}}=64\,F. \qquad (11)$$

From a LUT, we can obtain the V_{min}–f relationship as follows:

$$V_{min32}=\psi_{32}(f_{32}) \qquad (12)$$

$$V_{min16}=\psi_{16}(f_{16}) \qquad (13)$$

$$V_{min8}=\psi_8(f_8). \qquad (14)$$

As algorithm B keeps the supply voltage constant

$$\psi_{32}(f_{32})=\psi_{16}(f_{16})=\psi_8(f_8)=V \qquad (15)$$

the operating frequencies f32, f16,andf8 can be determined by using (11) and (15). For example, when F is set to 50 MHz, the values for V, f32, f16,andf8 are found to be 1.35 V, 20 MHz, 25 MHz, and 35 MHz, respectively. The overall power consumption of algorithmBis shown in Table IV. Due to the complete removal of voltage transitions, the Pcompu_overhead is reduced. Simultaneously, because of holistic planning, the dynamic computation power is also optimized to a lower level. Compared with the fixed-width 32×32 bit standard multiplier, 81.5% power reduction is achieved with a total silicon area overhead of only 11.9%, when considering DVS, razor, RAM, and dedicated circuitry for scheduling Algorithm B.

E. Algorithm C

Although AlgorithmBremoves power supply voltage transitions by setting a single-voltage level V,theremaybe better power saving combinations of power supply voltages and operating frequencies: (Vmin32, f32), (Vmin16, f16), and (Vmin8, f8). The aim of algorithmCis to find such an optimum for reduced power consumption. To limit complexity, we will only seek to minimize the dynamic power dissipated as a result of the computation

$$\begin{aligned}P &= CV^2f\\ &= C_{m32}V_{\min32}^2f_{32}+C_{m16}V_{\min16}^2f_{16}+C_{m8}V_{\min8}^2f_8\\ &= \chi(f_{32},f_{16}).\end{aligned}$$

Given that the Vmin–f relationships are known (12)–(14), one could find the minimum of the above equation for the specified throughput (11). For example, when F is set to 50 MHz, the values for (Vmin32, f32), (Vmin16, f16), (Vmin8, f8) are found to be (1.15 V, 15 MHz), (1.30 V, 20 MHz), and (1.75 V, 45 MHz), respectively. The overall power performance of algorithm C is shown in Table IV. When considering DVS, razor, RAM, and dedicated scheduling circuitry, algorithm B exhibits the least power consumption, with an overall power reduction of 86.3%, compared with the standard 32×32 bit fixed-width multiplier. However, it requires two additional dithering units to generate all three discrete power supply levels Vmin32, Vmin16,andVmin8 and thus remove transitions among these different supply levels. This increases the total silicon area overhead to 27.1%.

REFERENCES

Therefore, algorithm B provides the most attractive tradeoff with 81.5% reduction and a silicon area overheard of just 11.9%.

Simulation Result of Multiprecsion Reconfigurable:

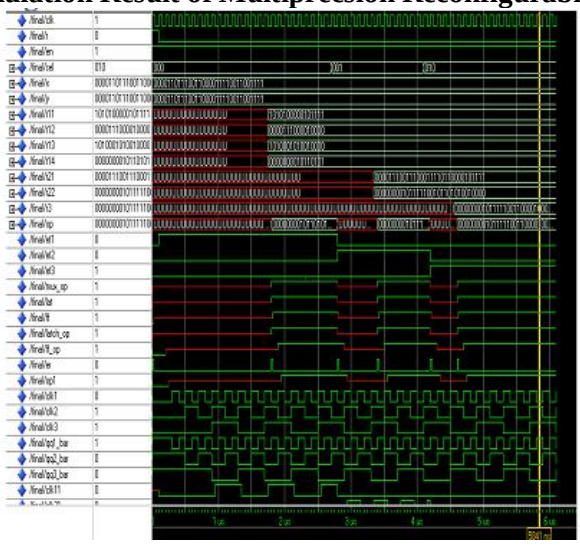


Fig.11shows the simulation result of multiprecision reconfigurable multiplier, in which dynamic voltage scaling and razor based error detection unit is used to provide full computational flexibility and low power application

Conclusion

Variable latency functional units using adaptive operation precision can allow aggressive supply voltage scaling and clock frequency scaling for improved power efficiency. In this paper a flexible multiplier combining variable precision processing, scaled voltage and clock frequency are used efficiently to reduce circuit power consumption. Various algorithms and topologies are explored to obtain high performance. Reported result show that variable precision multiplier enables a reduction of power dissipation compared to fixed precision multiplier. When operating under different precision, the multi-precision multiplier is used in attractive various general purpose low power application. Design of area and power-efficient high-speed data path logic systems are one of the most substantial areas of research in VLSI system design. Related work multi-precision multiplier can be implemented using high speed adders such as carry select adder for improving the performance of high speed flexible multiplier for design IIR filter application.

[1] R. Min, M. Bhardwaj, S.-H. Cho, N. Ickes, E. Shih, A. Sinha, A. Wang,and A. Chandrakasan, "Energy-centric enabling technologies for wireless sensor networks," IEEE Wirel. Commun., vol. 9, no. 4, pp. 28–39,Aug. 2002.

[2] M. Bhardwaj, R. Min, and A. Chandrakasan, "Quantifying and enhancing power awareness of VLSI systems," IEEE Trans. Very Large ScaleIntegr. (VLSI) Syst., vol. 9, no. 6, pp. 757–772, Dec. 2001.

[3] A. Wang and A. Chandrakasan, "Energy-aware architectures for a realvalued FFT implementation," in Proc. IEEE Int. Symp. Low PowerElectron. Design, Aug. 2003, pp. 360–365.

[4] T. Kuroda, "Low power CMOS digital design for multimedia processors," inProc. Int. Conf. VLSI CAD, Oct. 1999, pp. 359–367.

[5] H. Lee, "A power-aware scalable pipelined booth multiplier," in Proc.IEEE Int. SOC Conf., Sep. 2004, pp. 123–126.

[6] S.-R. Kuang and J.-P. Wang, "Design of power-efficient configurablebooth multiplier," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57,no. 3, pp. 568–580, Mar. 2010.

[7] O. A. Pfander, R. Hacker, and H.-J. Pfleiderer, "A multiplexer-based concept for reconfigurable multiplier arrays," inProc. Int. Conf. Field Program. Logic Appl., vol. 3203. Sep. 2004, pp. 938–942.

[8] F. Carbognani, F. Buergin, N. Felber, H. Kaeslin, and W. Fichtner,"Transmission gates combined with level-restoring CMOS gates reduce glitches in low-power low-frequency multipliers," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 16, no. 7, pp. 830–836, Jul. 2008.

[9] T. Yamanaka and V. G. Moshnyaga, "Reducing multiplier energy by data-driven voltage variation," inProc. IEEE Int. Symp. Circuits Syst.,May 2004, pp. 285–288.

[10] W. Ling and Y. Savaria, "Variable-precision multiplier for equalizer with adaptive modulation," inProc. 47th Midwest Symp. Circuits Syst.,vol.1. Jul. 2004, pp. I-553–I-556.