A Reconfigurable System for Digital Signal Processing

Huang Letian, Li Guangjun
School of Communication and Information Engineering, Electronics Science and Technology University of China, (UESTC) Chengdu, Sichuan P R China Post Code:610054
Email: huanglt@uestc.edu.cn gjli@uestc.edu.cn

Abstract

Commonality of various algorithms is analyzed based on the research of algorithms commonly used digital signal processing and a reconfigurable cell is proposed. A reconfigurable system with the cells is designed, which is widely used in a variety of digital signal processing. The principle and method of using this system are discussed with the basic algorithms of digital signal processing - multiplication and adder - as example. By means of simulation and implementation, it is found that this system has much higher processing speed and efficiency, compared with other kinds of GM multipliers.

Key words— reconfiguration system; digital signal processing; algorithms, cell; multiplier

1. Introduction

In recent years, research on reconfigurable computation is becoming a focus. Studies have been conducted in this field by researchers in different countries from various directions. One of the hot topics is the reconfigurable digital system. Reconfigurable digital system is basis of the related theories of reconfigurable computation, algorithms, and their applications. Reconfigurable systems are divided into two categories in accordance with types of the architectures — systems based on fine-grained architectures and those based on coarse-grained architectures respectively. Professor R. Hartenstein reviewed the research literature on systems based coarse-grained reconfigurable architecture from 1990 to 2000 and pointed out that their applications are heading from niche to mainstream, bridging the gap between ASICs and micro-processors. [1]

Most of the cells in reconfigurable digital systems based on coarse-grained architectures are actualized via simple microprocessors, such as ADRES [2] (Architecture for Dynamically Reconfigurable Embedded Systems) XPP [3] etc. Furthermore, most of the coarse-grained reconfigurable systems are in essence multi-core processors or multi- processor arrays. In fact such structures to certain extent waste a lot of hardware resources and lower their own efficiency. This violates the purpose of reconfigurable computation. In this paper，a new reconfigurable computation system applicable to digital signal processing is proposed to improve processing efficiency. By means of simulation and implementation, it is found that the system is much faster and more efficient in doing multiplications, compared with other multipliers.

978-1-4244-2179-4/08/$25.00 ©2008 IEEE

2. Analyses of Algorithms in Digital Signal Processing

2.1. Computations commonly used in digital signal processing

Algorithms commonly used in digital signal processing include the fast Fourier transform (FFT), the discrete cosine transform (DCT), the finite impulse response (FIR) digital filter, the infinite impulse response (IIR) digital filter, dot product etc. Almost all the algorithms are based on multiplication-addition operations. DCT is a case in point, whose definition can be expressed as the following formula.

$$X(k)=e(k)\sum_{n=0}^{N-1}x(n)\cos[\frac{(2n+1)k\pi}{2N}],k=0,1...,N-1 \quad (1)$$

$$x(n)=\frac{2}{N}\sum_{K=0}^{N-1}e(k)X(k)\cos[\frac{(2n+1)k\pi}{2N}],n=0,1,...,N-1 \quad (2)$$

$$e(k)=\begin{cases}\frac{1}{2},k=0\\1,k\neq 0\end{cases} \quad (3)$$

N point DCT transform can be derived from a $2N$ point discrete-time Fourier transform. A $2N$ point sequence y(n) can be formed via x(n) and its mirror symmetry as follows.

$$y(n)=x(n)+x(2N-n-1)=\begin{cases}x(n),0\le n\le N-1\\x(2N-n-1),N\le n\le 2N-1\end{cases} \quad (4)$$

The DFT of y(n) can be expressed as

$$Y_D=\sum_{n=0}^{2N-1}y(n)e^{-j\frac{2\pi}{2N}kn}=\sum_{n=0}^{N-1}x(n)e^{-j\frac{2\pi}{2N}kn}+\sum_{n=0}^{N-1}x(2N-n-1)e^{-j\frac{2\pi}{2N}kn} \quad (5)$$

Digital signal processing algorithms can be generally expressed as follows

$$\sum y(k)=\sum x_1(k)h_1(k)+\sum x_2(k)h_2(k)...+\sum x_n(k)h_n(k) \quad (6)$$

Therefore, it is reasonable to take a unit capable of multiplication-addition operation as a reconfigurable cell of a digital signal processing system.

2.2 Analyses of fixed-point multiplication

Multiplication can be divided into two categories — fixed-point multiplication and floating-point multiplication. Each category can be further divided into signed multiplication and unsigned multiplication. Fixed-point multiplication is a more basic computation. Under certain conditions, floating-point multiplication can be transformed into fixed-point multiplication, and signed multiplication can also be transformed into unsigned multiplication. For this reason, this article mainly discusses unsigned fixed-point multiplication. A binary constant B can be expressed as

$$B = b_n b_{n-1}...b_1 b_0 = b_n 2^n + b_{n-1}2^{n-1}...+ b_1 2^1 + b_0 2^0, \quad (b_n, b_{n-1}...b_1, b_0 \in \{0,1\}) \tag{7}$$

and the fixed-point multiplication of two binary constants A and B can be expressed as

$$AB = Ab_n 2^n + Ab_{n-1}2^{n-1}...+ Ab_1 2^1 + Ab_0 2^0. \quad (b_n, b_{n-1}...b_1, b_0 \in \{0,1\}) \tag{8}$$

If $b_k \in \{b_n, b_{n-1}..b_1, b_0\} = 0$,

it can be excluded for the product is 0. However, if $\{p_l, p_{l-1}...p_1, p_0\} \subset \{n, n-1...1, 0\}, b_{pl} = b_{pl-1} = ...b_{p1} = b_{p0} = 1$

Formula 8 can be revised as

$$AB = A2^{pl} + A2^{pl-1} + ...A2^{p1} + A2^{p0}. \tag{9}$$

Since $A \cdot 2^k (k \in \{n, n-1...1, 0\})$ can be realized by displacement A left to the position of k, it is revealed in Formula 9 that all unsigned fixed-point multiplications can be transformed as displacement-addition operations:

$$AB = A << p_l + A << p_{l-1}... + A << p_1 + A << p_0. \tag{10}$$

If only unsigned fixed-point multiplication is taken into consideration, the general formula of digital signal processing algorithms can be transformed into a displacement-addition formula as the following

$$\sum y(k) = \sum_{l=0}^{i} x_1(k) << p_l + \sum_{m=0}^{j} x_2(k) << p_m ... + \sum_{n=0}^{i} x_n(k) << p_n \tag{11}$$

3. Reconfigurable Digital Signal Processing System

3.1. Framework of reconfigurable digital system architecture

Although reconfigurable digital systems vary greatly from one another, all of them can be generally divided into two components, that is, reconfigurable cells and reconfigurable interconnection modules. Reconfigurable cells doing data processing are the basic function units of a reconfigurable digital system, while reconfigurable interconnection modules units function in data transmission, data exchange etc.

3.2. Design of reconfigurable cells

Since digital signal processing algorithms can be transformed into a displacement-addition formula, displacement-addition can be used as the basic operation of digital signal processing. Multiple algorithms can be realized by means of designing cells doing displacement-addition operations.

Such cells are made of two input units with displacement functions and an adder. Controlled by the displacement controlling signals, the two input units in turn control the displacement of the two input variables.

Figure 1 illustrates a general framework of reconfigurable cells.

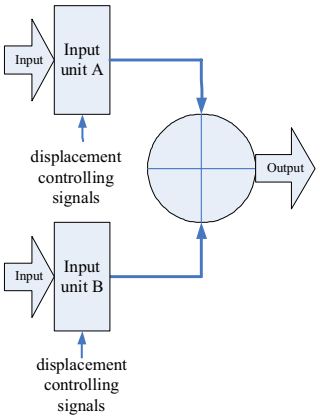

Fig 1. General framework of reconfigurable cells

In order to improve system performance, it is recommendable to add a stage of pile-line between the input units and the adder. It is also necessary to add FIFO before the input units to balance data flow and to avoid data loss.

3.3. Design of reconfigurable interconnection modules

The main structures of the reconfigurable interconnection modules include mesh, array, tree, and cube. Reconfigurable digital systems based on coarse-grained architectures require that all the inner processing units and interconnection modules must be on a same chip. Mesh and array are now more widely used because it is difficult to use tree and cube in IC design.

As far as the present system is concerned, tree is the best interconnection structure, but tree is not adopted for it difficult to be used in IC design. Instead, mesh is adopted for it is more commonly used and more flexible than array in making interconnections. In this system since many changes are needed in making interconnections of cells, a greater flexibility in interconnections is a must, and therefore the best choice is to use mesh.

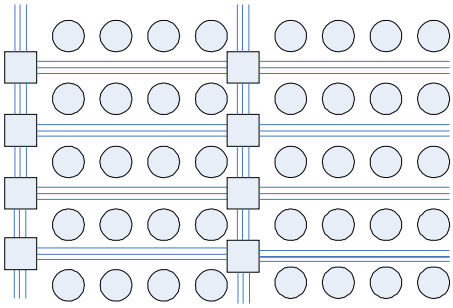

Fig 2 Architecture of reconfigurable digital signal processing system

In Figure 2 the circles represent cells and the squares represent interconnection modules. 4 cells are lined as one group and the input data buses are 2*4=8, while the output buses are 4. The horizontal buses are input buses and the vertical buses are output buses.

Multistage Interconnection Network (MIN) is used to interconnect all the buses to improve the system's flexibility in making interconnections. See Figure 3.

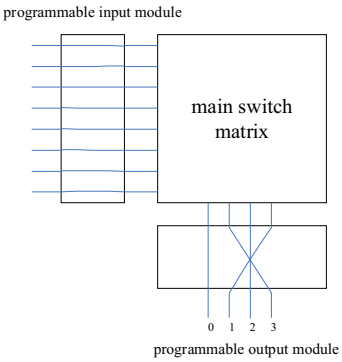

Fig 3 .Design of MIN

In Figure 3 MIN is made up of three components — a programmable input module, a main switch matrix, and a programmable output module. The inner connections of each can be changed via configurations. For instance, the original sequence of buses 0, 1, 2, and 3 of the output module is changed into the sequence of 0, 3, 2, and 1 post a configuration. Flexibility of interconnections among the modules can be greatly improved via adding only a little complexity, because the complexity of programmable input and output modules is linear.

Various implementation structures can be adopted in accordance with the requirements of specific algorithms and the situations of system resource consumption. Since all the cell units are the same, great flexibility is assured in making interconnections in actual algorithms implementation. Unconsumed resources can be flexibly used in the process of dynamic reconfiguration to realize random reconfiguration of the system and to make the best use of resources within the system.

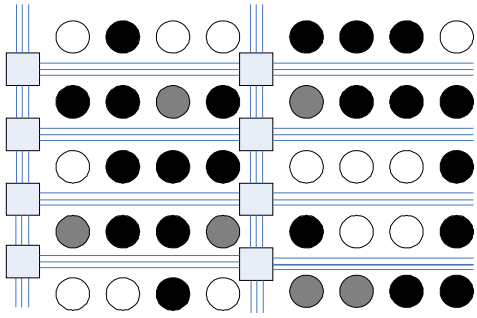

Fig 4 .A reconfigurable digital signal processing system in working

In Figure 4, the dark circles represent cells in working; the white ones represent those not in uses, while the gray ones represent those in the process of configuration. It takes little time to realize such configurations because only a few signals are changed during system configuration, which control displacement of the cells' input data and the interconnection modules respectively.

3.4. Analyses of system performance

It can be seen in Formula 10 that the number of steps needed to realize the number of digits in the multiplier with a value of 1 decides a multiplication via displacement-addition. Suppose N operations are needed, and A, B are expressed as the follows $A = a_n a_{n-1} \ldots a_1 a_0; B = b_n b_{n-1} \ldots b_1 b_0 (a_n, \ldots a_1, a_0; b_n, \ldots b_1, b_0 \in \{0,1\})$, then the number of operations for the binary multiplication of the two n digits can be expressed as

$$N = \sum_{i=0}^{n} b_i (b_n, b_{n-1} \ldots b_1, b_0 \in \{0,1\}) \qquad (12)$$

A GM Multiplier can do multiplication of any two numbers. It is very flexible but lowers performance of many other functions. No matter for an array multiplier, a multiplier with a STG controller, or a sequential multiplier based on ASMD, the product is computed by means of using each digit of the multiplier to control sequentially or simultaneously the steps of displacement-addition of the multiplied. The multipliers (like coefficients of a digital filter) are usually fixed during digital signal processing, and it is unnecessary to decide the value of each digit of the multiplier, but only to configure cells in terms of number and functions according to which of the multiplier's digits have a value of 1, to reduce the number of displacement-addition operations, and in turn to improve the performance of the system and to make better use of the resources.

Suppose the number of digits of both the multiplier and the multiplied is N, the number of digits in the multiplier with a value of 1 is p, then $p \leq N$. The data on performance of multiplier implemented by the cells and other multipliers are listed in Table 1.

Table 1. Performance of different multipliers

	by cells	Array multiplier	with a STG controller	based on ASMD
number of addition operation	*p*	*N*-1	*p*	*p*
number of displacement operation	*p*	*N*-1	*N*-1	*N*-1
cycle needed for each multiplication	*1*	*1*	*related to number of states*	*N+1*
hardware resource needed	related to *p*	Many	few	a few

A simulation comparison is made on the timing diagram of the multiplier implemented by cells and that based on ASMD. Figure 5 represents the operation sequence of the former, and Figure 6 represents the operation sequence of the latter. In Figure 5 variable A is the multiplied and variable B is the product (the outcome of the computation), while in Figure 6 word1 is the multiplied and word2 is the multiplier, and product is the outcome of the computation. Variable A, word1, word2 are all 8-bit unsigned numbers.

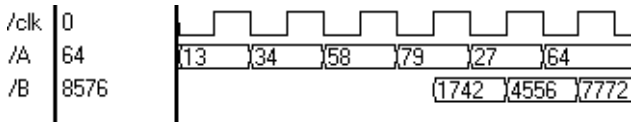

Fig 5. Operation sequence of multiplier implemented by cells

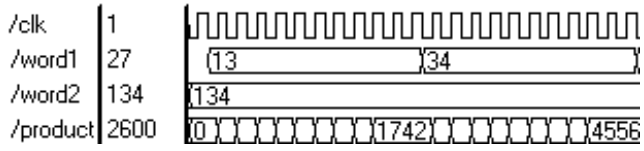

Fig 6. Operation sequence of GM Multipliers based on ASMD

Comparing Figure 5 and Figure 6, we can see that it take only 1 cycle to do the multiplication if a multiplier implemented by cells is used. In contrast, 9 cycles are needed to do the same thing by a multiplier based on ASMD.

Table 2 lists the data gained from synthesizing the two programs under the condition of using FPGA type of Cyclone II EP2C5F256C8.

Table 2. Performance of multiplier implemented by cells and multiplier based on ASMD

	highest frequency of clock	*cycle needed*	*time needed for each multiplication*
multiplier implemented by cells	240.73 MHz	1	4.154 ns
multipliers based on ASMD	231.00 MHz	9	38.961 ns

In Table 2 it can be seen that the computation of multiplication implemented by multiplier using cells is 9 times faster than that by multiplier based on ASMD. Under the condition of using ASIC to realize the system, the performance of the system will be improved still more greatly.

4. Conclusion

A kind of reconfigurable system, which is widely used in a variety of digital signal processing, is described in this article. Comparison on performance of this system and different multipliers shows that the system is more efficient. Unconsumed resources can be flexibly configured in the process of dynamic reconfiguration of the system and the hardware resources can be best used. In conclusion, this system is of great value for further research.

References

[1] Reiner Hartenstein. “A Decade of Reconfigurable Computing: a Visionary Retrospective.” Design, Automation and Test in Europe Conference (DATE), Munchen, Germany, 642-649 2001

[2] B. Mei, S. Vernalde, “ADRES: An architecture with tightly coupled VLIW processor and coarse-grained reconfigurable matrix.” The International Conference on Field Programmable Logic and Applications (FPL’03), 61-70. 2003.

[3] V. Baumgarte, G. Ehlers, “PACT XPP–A Self-Reconfigurable Data Processing Architecture.”The Journal of Supercomputing, 167-184. 2003, 26(2):

[4] Keshab K. Parhi. “VLSI Digital Signal Processing System Design and Implementation.” Beijing：Publishing House of Electronic Industry, 2004.

[5] Michael D Ciletti. “Advanced Digital Design with the Verilog HDL.” Beijing：Publishing House of Electronic Industry, 2007.