World Academy of Science, Engineering and Technology
International Journal of Electrical and Computer Engineering
Vol:3, No:6, 2009

A Dynamically Reconfigurable Arithmetic Circuit for Complex Number and Double Precision Number

Haruo Shimada and Akinori Kanasugi

***Abstract*—**This paper proposes an architecture of dynamically reconfigurable arithmetic circuit. Dynamic reconfiguration is a technique to realize required functions by changing hardware construction during operations. The proposed circuit is based on a complex number multiply-accumulation circuit which is used frequently in the field of digital signal processing. In addition, the proposed circuit performs real number double precision arithmetic operations. The data formats are single and double precision floating point number based on IEEE754. The proposed circuit is designed using VHDL, and verified the correct operation by simulations and experiments.

***Keywords*—**arithmetic circuit, complex number, double precision, dynamic reconfiguration

I. INTRODUCTION

LSIs are embedded in almost all electronic equipments. Recently, LSIs become large scale dramatically, with miniaturization and speed-up of transistors. Reconfigurable architectures change constructions of circuit for miniaturization of LSIs. Especially, dynamically reconfigurable architectures attract attention, because they change circuits during operation [1, 2].

This paper proposes a complex number / real double precision number calculation circuit with dynamically reconfigurable architecture. The complex number calculations are used frequently in the field of digital signal processing [3, 4]. A multiplication circuit for complex number needs two real adders and four real multipliers. The circuit reconfigures dynamically by switching connections between the circuits and inputs. By reconfiguration, the proposed circuit can realize the following calculations; (i) complex number multiplication, (ii) complex number division, (iii) real number parallel calculation (single / double precision), (iv) real number MAC (Multiply-Accumulation; single / double precision).

The data formats are single and double precision floating point number based on IEEE754.

H. Shimada was with Tokyo Denki University, now with TAMURA Corporation, Tokyo, Japan (e-mail: island-paddy-field@hotmail.co.jp)

A. Kanasugi is with the Department of Electrical and Electronic Engineering, Tokyo Denki University, Tokyo, Japan (e-mail: kanasugi@eee.dendai.ac.jp).

This work was supported by the grant from the Foundation for Technology Promotion of Electronic Circuit Board, Japan.

II. DYNAMIC RECONFIGURATION

Usually, reconfiguration requires break of circuit in a few milli seconds. As a result, reconfiguration time prevents speed up. On the other hand, dynamic reconfiguration changes circuit construction during operation without breaking circuit. Thus, dynamic reconfiguration enables reconfiguration of circuit in one clock cycle or a few nano seconds.

In this paper, a dynamically reconfigurable arithmetic circuit is proposed. The arithmetic circuit consists of two adders, four multipliers, and one divider with single precision floating point number. As shown in Fig. 1, the proposed circuit performs complex number multiplication, real number parallel calculation, multiply-accumulation, and so on by reconfiguration.

In addition to that, the dynamically reconfigurable double precision floating point adder was designed (Fig. 2). The circuit is composed of two single precision adders, and achieves one double precision adder by dynamic reconfiguration. Similarly, the dynamically reconfigurable double precision floating point multiplier was designed.

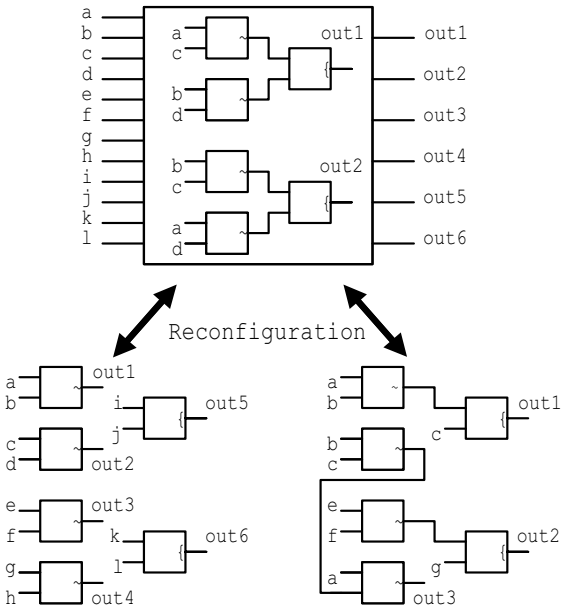

Fig. 1 Dynamic reconfiguration of arithmetic circuit

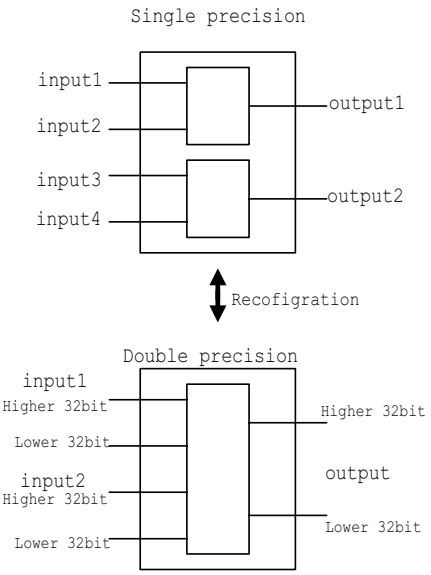

Fig. 2 Dynamic reconfiguration of single / double precision

III. Floating Point Number Based on IEEE754 Format

We use single and double precision IEEE754 formats which are adopted most widely at the calculations of the floating point number. It decides expression format, particular value, rounding method, precision, and so on. Fig. 3 shows expression format.

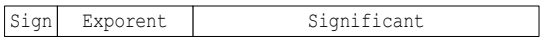

Fig. 3 Format of floating point number

TABLE I
BIT WIDTH OF EACH PART FOR SINGLE AND DOUBLE PRECISION

Symbol	Single	Double
Total	32 bit	64 bit
Sign	1 bit	1 bit
Exponent	8 bit	11 bit
Significant	23 bit	52 bit

A single precision floating point number is converted into a binary number as follows; where, C is sign part, E is exponent part, and S is significant part.

$$(-1)^C \times 1.S \times 2^{E-01111111} \quad (1)$$

The treatments in this paper are shown below.

1) Rounding process uses Unbiased and Towards Zero.
2) Underflow and un-normalized number are not used.
3) Exception processes are overflow, infinity, and NaN.

IV. Dynamically Reconfigurable Arithmetic Circuit

Dynamically reconfigurable arithmetic circuit consists of the single / double precision adders, multipliers, and a single precision divider. Fig. 4 shows the block diagram of proposed circuit. The circuit is designed using VHDL.

The reconfiguration is controlled by three bit "sel" (select functions). The proposed dynamically reconfigurable circuit reconfigures eight modes as indicated by the following.

(i) Single precision floating point arithmetic

1) Complex number multiplication (sel = "000")
2) Real number parallel calculation (sel = "001")
3) Real number MAC (a × b + c type) (sel = "010")
4) Real number MAC (a × b + c × d type) (sel = "011")
5) Complex number division real part (sel = "100")
6) Complex number division imaginary part (sel = "101")

(ii) Double precision floating point arithmetic

7) Real number parallel calculation (sel = "110")
8) Real number MAC (A × B + C × D type) (sel = "111")

The proposed circuit reconfigures by switching the multiplexers which connect between circuits in Fig. 4. In addition, the adder and multiplier reconfigure one double precision arithmetic circuit and two single precision arithmetic circuits, respectively. In this paper, rounding process of "Unbiased" and "Towards Zero" are used. "Towards Zero" is used in the dynamically reconfigurable double precision adder and multiplier, while "Unbiased" is used in the single precision divider.

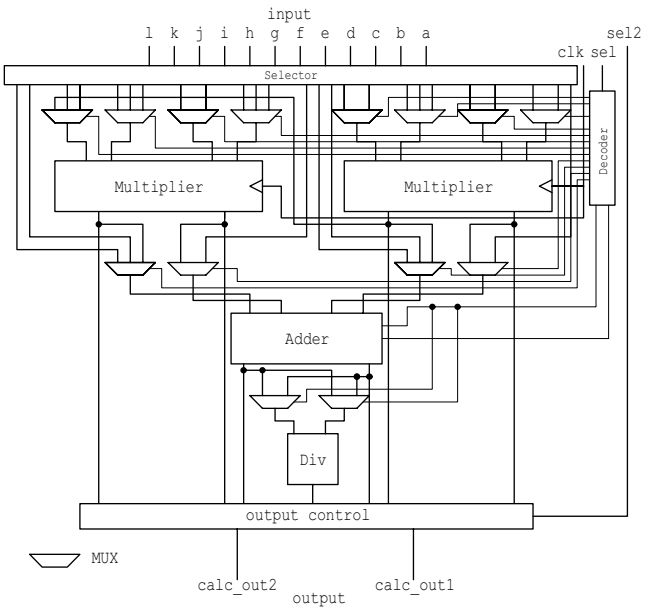

Fig. 4 Block diagram of dynamically reconfigurable arithmetic circuit

The detail of proposed circuit is described as follows.

A. Decoder

Fig. 5 shows the decoder. The three bit input signal "sel" is decoded into 18 bit output signal "wire". One or two bit of "wire" become input select signals of each multiplexers. At the same time, decoder set output signal "pmflg" which selects calculations of addition or subtraction. The output signal "accuracy" selects the accuracy of the adder and the multiplier.

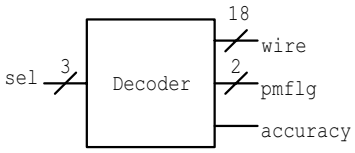

Fig. 5 Decoder

B. Dynamically reconfigurable adder

Fig. 6 shows the dynamically reconfigurable single / double precision floating point adder. This circuit operates as two adders of two inputs one output at single precision mode. That is, the circuit calculates;

add_in1 + add_in2 = add_out1

add_in3 + add_in4 = add_out2.

On the other hand, this circuit operates as one adder of four inputs two outputs at double precision mode. If two inputs of 64 bit double precision floating point numbers are assumed to be A and B; higher 32 bit of A, lower 32 bit of A, higher 32 bit of B, and lower 32 bit of B are input to add_in1, add_in2, add_in3, and add_in4, respectively. Similarly, higher 32 bit of the result of A + B is output to add_out1, and lower 32 bit is output to add_out2. Single precision and double precision are switched by the "accuracy" signal.

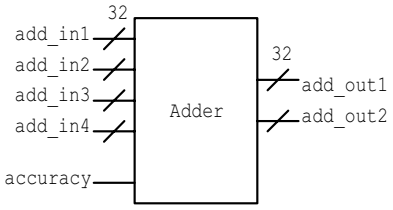

Fig. 6 Dynamically reconfigurable adder

C. Dynamically reconfigurable multiplier

Fig. 7 shows the dynamically reconfigurable single / double precision floating point multiplier. As similar as the adder, this circuit operates as two multipliers of two inputs one output at single precision mode. That is, the circuit calculates

add_in1 × add_in2 = add_out1

add_in3 × add_in4 = add_out2.

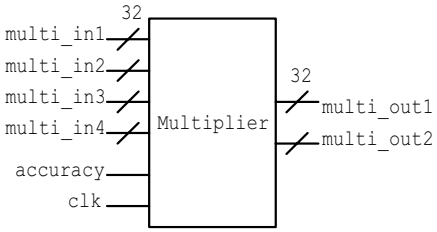

Fig. 7 Dynamically reconfigurable multiplier

On the other hand, this circuit operates as one multiplier of four inputs two outputs at double precision mode. If two inputs of 64 bit double precision floating point numbers are assumed to be A and B; higher 32 bit of A, lower 32 bit of A, higher 32 bit of B, and lower 32 bit of B are input to multi_in1, multi_in2, multi_in3, and multi_in4, respectively. Similarly, higher 32 bit of the result of A × B is output to multi_out1, and lower 32 bit is output to multi_out2. Single precision and double precision are switched by the "accuracy" signal. The double precision multiplication is calculated with a clock, divided into three portions.

D. Single precision floating point divider

Fig. 8 shows single precision floating point divider. This circuit calculates that div_in1 / div_in2 = div_out. This divider uses recovery method.

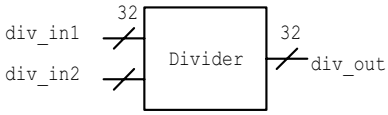

Fig. 8 Single precision floating point divider

E. Output controller

Fig. 9 shows output controller. The seven outputs of adders, multipliers, and a divider are inputs of this circuit. Signals data_in1 - 4 are four multiplier outputs. Signals data_in5 and 6 are two adder outputs. Signal data_in7 is one divider output. Signal "sel" and "sel2" control outputs.

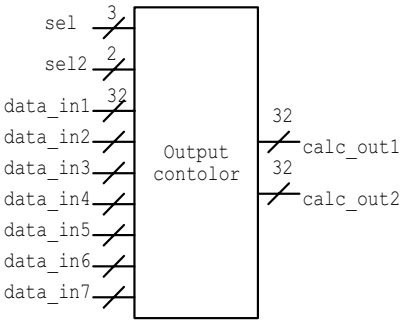

Fig. 9 Output controller

The result of the selected calculation by the "sel" signal is the output. If the output value becomes two or more pieces, two pieces are separately output by the "sel2" signal. In the case of real number parallel operation, the six pieces of outputs are sent divided into three times.

V. Reconfiguration

At first, the single precision calculations are shown as follows ($A - F$).

A. Complex number multiplication (sel = "000")

The multiplication of two complex number $Z_1 = a + jb$ and $Z_2 = c + jd$ becomes $Z_1 \times Z_2 = (ac - bd) + j\,(bc + ad)$. The complex number is multiplied by switching the multiplexer to obtain a correct calculation result.

Fig. 10 shows the connection diagram for complex number multiplication.

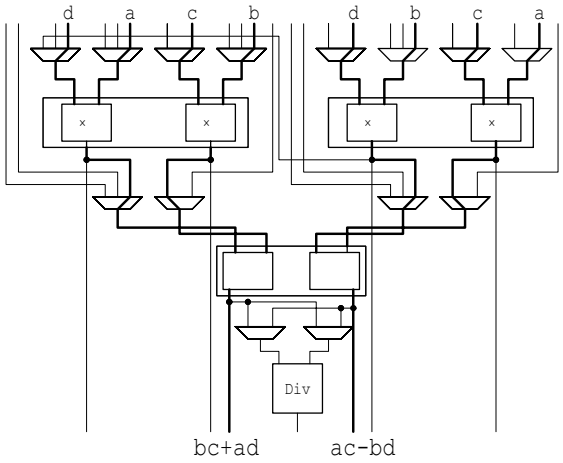

Fig. 10 Connection diagram for complex number multiplication

B. Real number parallel calculation (sel = "001")

Each arithmetic circuit is independently operated for twelve real numbers inputs (*a* - *l*). Six operations are done in parallel as follows.

Multiplication: $a \times b$, $c \times d$, $e \times f$, $g \times h$

Addition: $i + j$, $k + l$

Fig. 11 shows the connection diagram for real number parallel operations.

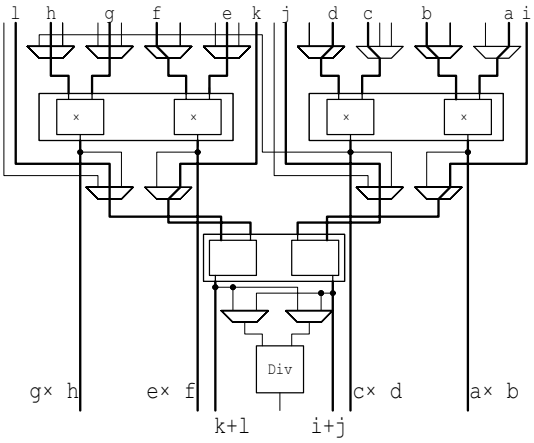

Fig. 11 Connection diagram for real number parallel operations

C. Real number MAC (a × b + c type) (sel = "010")

Real number MAC (multiply-accumulation, $a \times b + c$ type) is executed. Two MAC ($a \times b + c$, $e \times f + g$) can be executed, because one MAC operation uses one multiplier and one adder. In addition, one multiplier of three inputs like $a \times b \times c$ can be composed, because two multipliers are not used.

Fig. 12 shows the connection diagram for real number MAC ($a \times b + c$ type).

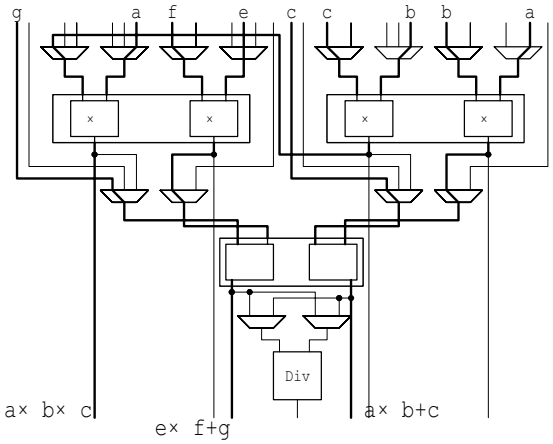

Fig. 12 Connection diagram for real number MAC ($a \times b + c$ type)

D. Real number MAC (a × b + c × d type) (sel = "011")

Real number MAC ($a \times b + c \times d$ type) is executed. Two MAC ($a \times b + c \times d$, $e \times f + g \times h$) can be executed, because one MAC operation uses two multipliers and one adder.

Fig. 13 shows the connection diagram for real number MAC ($a \times b + c \times d$ type).

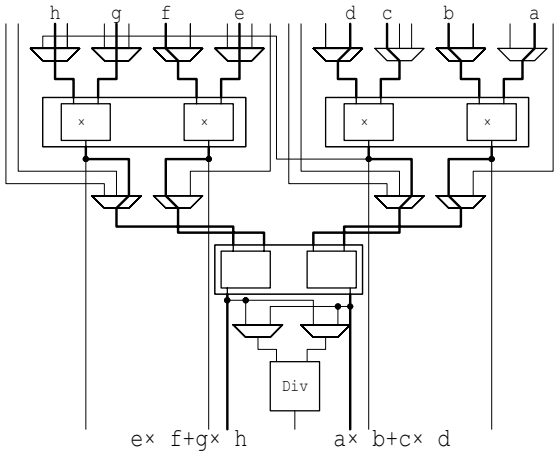

Fig. 13 Connection diagram for real number MAC ($a \times b + c \times d$ type)

E. Complex number divider real part (sel = "100")

The division of two complex number $Z_1 = a + jb$ and $Z_2 = c + jd$ becomes $Z_1 / Z_2 = (ac+bd) / (c^2+d^2) + j\,(bc-ad) / (c^2+d^2)$. A real part of the complex number division ($(ac+bd) / (c^2+d^2)$) is calculated in this mode.

The connection diagram is shown in Fig. 14. Two multipliers and one adder of the left side calculate denominator $c^2 + d^2$ in Fig. 14. Similarly, two multipliers and one adder of the right side calculate numerator $ac + bd$.

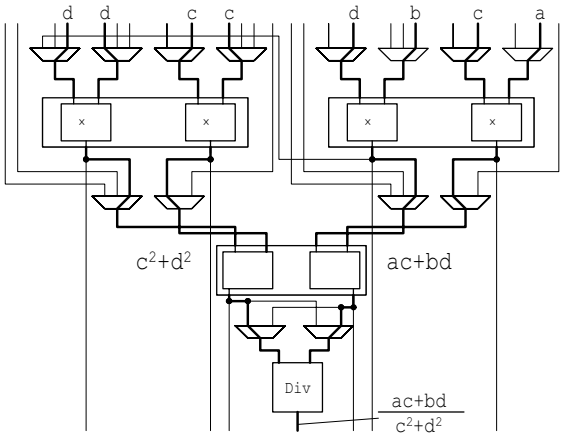

Fig. 14 Connection diagram for complex number division (real part)

F. Complex number divider imaginary part (sel = "101")

A imaginary part of the complex number division $((bc-ad)/(c^2+d^2))$ is calculated. The connection diagram is shown in Fig. 15. Two multipliers and one adder of the left side calculate denominator $c^2 + d^2$ in Fig. 15. Similarly, two multipliers and one adder of the right side calculate numerator $bc - ad$.

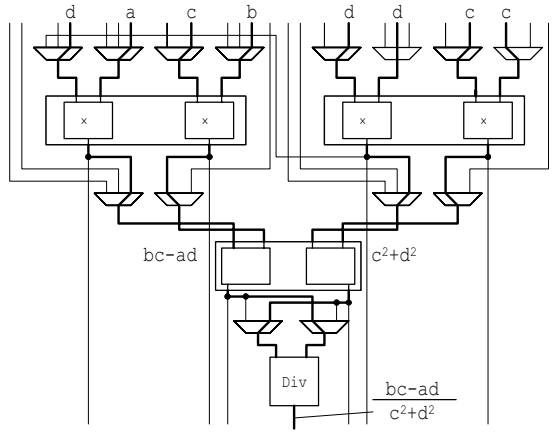

Fig. 15 Connection diagram for complex number division (imaginary part)

Next, two calculations (G and H) in double precision are described. As for the double precision input, the data that two single precision input data were simply connected is used. For instance, higher 32 bit of A (= a & b) in double precision data is 'a' and lower 32 bit of A is 'b', where '&' means a connection of bit strings. The inputs of double precisions are as follows.

$A = a \,\&\, b = 5026.321297352928,$

$B = c \,\&\, d = 10.713167202879671$

$C = e \,\&\, f = 3554.900876960291$

$D = g \,\&\, h = 355030.06326240901$

$E = i \,\&\, j = 657661.12784750015$

$F = k \,\&\, l = 4.0309885854732644 \times 10^{14}$

G. Real number parallel calculation (sel = "110")

The three double precision real number calculations shown as follows are executed in parallel.

Multiplication: $A \times B,\ C \times D$

Addition: $E + F$

Fig. 16 shows the connection diagram for double precision real number parallel calculations.

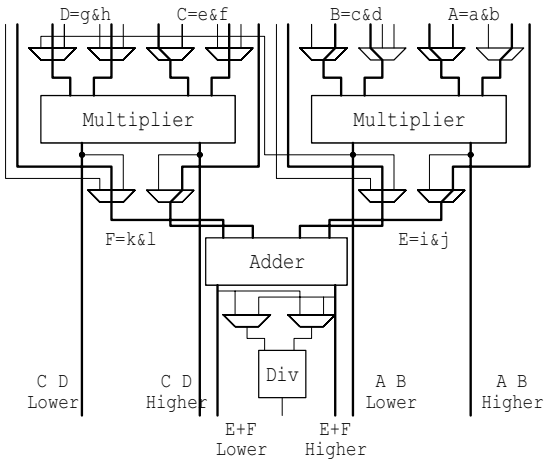

Fig. 16 Connection diagram for double precision real number parallel calculations

H. Real number MAC (A × B + C × D type) (sel = "111")

The double precision real number MAC (A × B + C × D type) is executed. Fig. 17 shows the connection diagram for double precision real number MAC (A × B + C × D type).

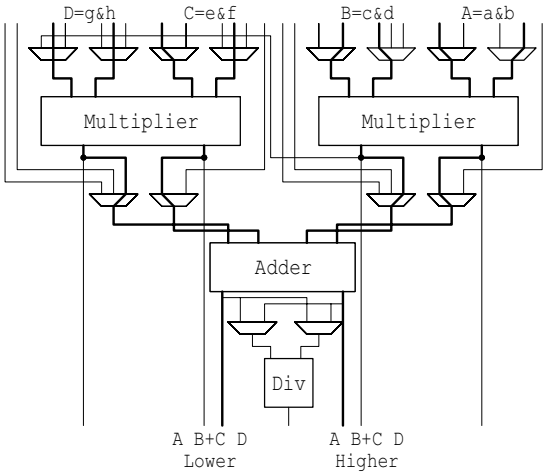

Fig. 17 Connection diagram for double precision real number MAC (A×B + C×D type)

VI. Simulations

The proposed dynamically reconfigurable circuit was synthesized by ISE 10.1 (Xilinx Inc.). The circuit was simulated using ModelSim XE III 6.3c (Mentor Graphics

Corp.).

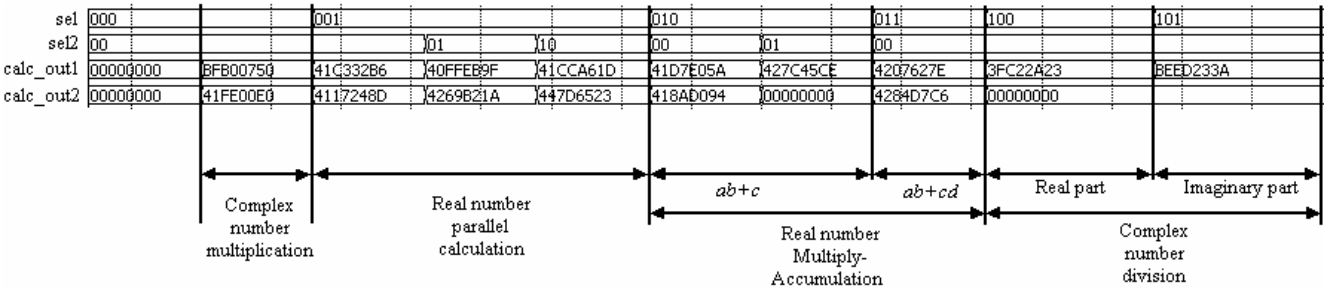

Fig. 18 Simulation result of single precision calculation

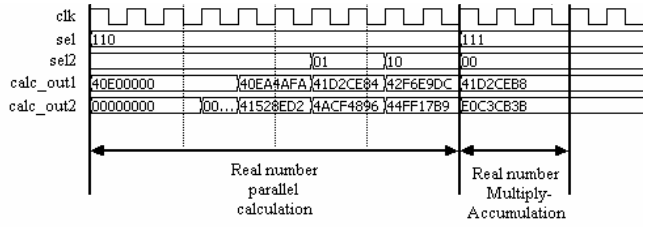

Fig. 19 Simulation result of double precision calculation

The twelve inputs of the simulation are as follows;

a =5.6135645, b =4.3465721, c =2.5847865, d =3.6546244,

e =5.3678954, f =1.4898785, g =9.3543321, h =6.2456545,

i =10.254389, j =15.326721, k =123.45676, l =890.12346

Fig.18 and 19 show the simulation results of the proposed single and double precision circuits, respectively. Output values are shown in hexadecimal format.

These results show that the circuit performs reconfiguration and various calculations by "sel" signal. The single precision floating point calculation results converted decimal numbers have almost seven digit accuracy. The double precision calculation results have almost fifteen digit accuracy. Thus, the correct operations of proposed circuit were verified.

VII. Experiments

The proposed circuit was implemented on FPGA (Xilinx Inc., Virtex-4) circuit board shown in Fig. 20 [5]. Xilinx ISE Design Suite 10.1 was used for implementation. "MicroBlaze" [6] is a soft IP core processor which works as a CPU in FPGA. The operation of "MicroBlaze" can be programmed by C language. "MicroBlaze" is connected to the proposed dynamically reconfigurable calculation circuit with the peripheral circuits. The block diagram is shown in Fig. 21. The peripheral circuits are registers and buffers which are designed by VHDL.

The communication between host PC and the proposed circuit is done through RS232C port. After converting the input decimal number into the floating point number, the host PC transmits the converted data and the selection signal to "MicroBlaze". "MicroBlaze" writes the received selection signal in the register and input data are written in the input buffer.

Fig. 20 FPGA circuit board

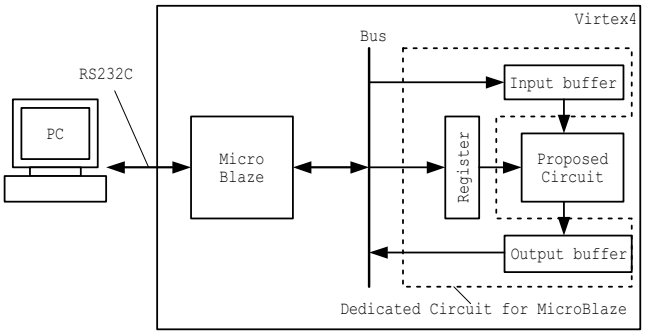

Fig. 21 Circuit configuration for hardware implementation

The proposed circuit reads the input data from the input buffer, operates according to the selection signal of the register, and writes the results in the output buffer. "MicroBlaze" reads out data from the output buffer, and transfers them to the host PC. The received data are converted into the decimal numbers, and the host PC displays them on the PC screen. As a result, decimal number input and decimal number output is realized.

The input data for single precision calculation is as similar as the simulation. The double precision input data is as follows.

A = 254.2657811245723

B = 562.2487956427975

C = 753.5610331726043

D = 341.5645322624046

E = 6576.627847498471

F = 4498.513984982131

The photograph of experiments on Virtex-4 FPGA board is shown in Fig. 22.

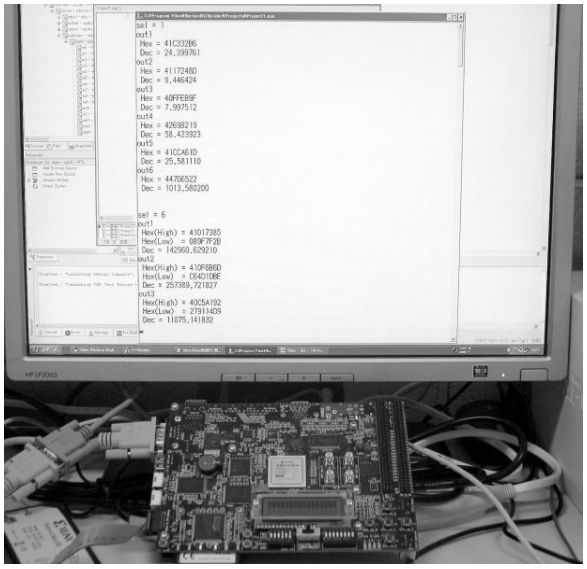

Fig. 22 Experiments on FPGA board

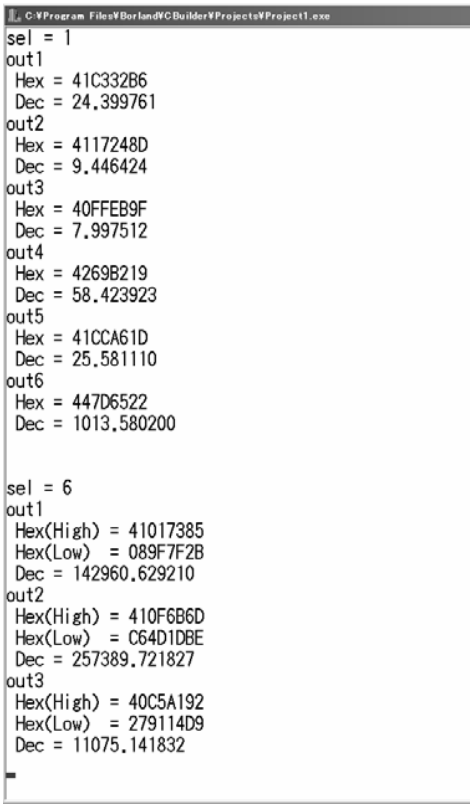

Fig. 23 Close up of PC screen

Fig. 23 is an example of close up of PC screen, where the single precision real number parallel operation and the double precision real number parallel operation are executed. It was confirmed that the proposed circuit operated correctly from the simulation and experiment results.

VIII. Conclusion

In this paper, an architecture of dynamically reconfigurable arithmetic circuit which can deal with complex number and double precision real number. The simulation results showed that eight kinds of operations were possible according to the selection signal. The designed arithmetic circuit was implemented on the Xilinx Virtex-4 evaluation circuit board. The arithmetic circuit was controlled by Xilinx "MicroBlaze" (soft IP processor), and the results were displayed on the screen of host PC. The results of experiment with the FPGA board showed that the proposed circuit operated correctly.

References

[1] T. J. Todman, G. A. Constantinides, S. J. E. Wilton, O. Mencer, W. Luk and P. Y. K. Cheung, "Reconfigurable computing: architectures and design methods", *IEE Proc.-Computers & Digital Techniques*, vol. 152, no. 2, pp. 193 - 207, 2005.

[2] T. Sato, H. Watanabe, K. Shiba, "Implementation of dynamically reconfigurable processor DAPDNA-2", *VLSI Design, Automation and Test, 2005 IEEE VLSI-TSA International Symposium*, pp. 323-324, 2005.

[3] M J. Myjak, J. G. Delgado-Frias, "A Medium-Grain Reconfigurable Architecture for DSP: VLSI Design, Benchmark Mapping, and Performance", *IEEE Trans. on VLSI Systems*, vol.16, no.1, pp.14-23, Jan 2008.

[4] Monte Tull, et al., "High-Speed Complex Number Multiplier and Inner-Product Processor", *IEEE Trans. on Circuits and Systems,* vol. 3, pp.III-640 - III-643, Aug. 2002.

[5] Xilinx, Inc., *ML401/ML402/ML403 Evaluation Platform User Guide UG080 (v2.5)*, May 2006

[6] Xilinx, Inc., *MicroBlaze Processor Reference Guide UG081 (v9.0)*, 2008.

Haruo Shimada was born at Tokyo in June 22, 1986. He received the B.E. from Tokyo Denki University in 2009. He is currently in TAMURA Corporation, Tokyo, Japan. His research interests include VLSI design for dynamically reconfigurable circuits.

Akinori Kanasugi was born at Tokyo in July 2, 1960. He received the B.E., M.E. and Ph. D from Saitama University, Japan, in 1983, 1985 and 1994, respectively.

After a research associate in Saitama University, he moved to Tokyo Denki University in 2002, where he is currently a professor in the faculty of engineering. His current research interests are in the development of VLSI systems such as reconfigurable processor, GA processor, and rough sets processor.

Prof. A. Kanasugi is a member of the Institute of Electronics, Information and Communication Engineers, Information Processing Society of Japan, the Japanese Society for Artificial Intelligence, and Japan Institute of Electronics Packaging.