RESEARCH ARTICLE OPEN ACCESS

Dynamically Reconfigurable Processor for Floating Point Arithmetic

S. Anbumani*, R. Kavin**, D. Sudharsan***
*Assistant Professor/Ece, Excel Engineering College
** Assistant Professor/Eee, Excel College Of Engineering and Technology
*** Assistant Professor/Ece, Excel Engineering College

Abstract

Recently, development of embedded processors is toward miniaturization and energy saving for ecology. On the other hand, high performance arithmetic circuits are required in a lot of application in science and technology. Dynamically reconfigurable processors have been developed to meet these requests. They can change circuit configuration according to instructions in program instantly during operations.This paper describes, a dynamically reconfigurable circuit for floating-point arithmetic is proposed. The arithmetic circuit consists of two single precision floating-point arithmetic circuits. It performs double precision floating-point arithmetic by reconfiguration. Dynamic reconfiguration changes circuit construction at one clock cycle during operation without stopping circuits. It enables reconfiguration of circuits in a few nano seconds. The proposed circuit is reconfigured in two modes. In first mode it performs one double precision floating-point arithmetic or else the circuit will perform two parallel operations of single precision floating-point arithmetic. The new system design reduces implementation area by reconfiguring common parts of each operation. It also increases the processing speed with a very little number of clocks.

I. INTRODUCTION

Recently, development of embedded processors is toward miniaturization and energy saving for ecology. On the other hand, high performance arithmetic circuits are required in a lot of application in science and technology. Dynamically reconfigurable processors have been developed to meet these requests. They can change circuit configuration according to instructions in program instantly during operations. This technology constructs required functions with minimum area .This paper describes about dynamic reconfiguration to miniaturize arithmetic circuits in general-purpose embedded processor. We reconfigure floating-point arithmetic circuits.The proposed dynamically reconfigurable circuit is reconfigured two modes as described by the following: (i) One double precision floating-point arithmetic,(ii) Two parallel operations of single precision floating-point arithmetic. We have designed an embedded processor which reduces implementation area by reconfiguring common parts of each operation .The proposed processor conforms to the instruction set of "MIPS-I" that is a typical 32-bit microprocessor .We evaluate implementation area and processing speed for floating-point arithmetic. Then we show effectiveness of the proposed circuit.

II. DYNAMIC RECONFIGURATION

Conventional circuits require a lot of clocks for reconfiguration. As a result, time of few *milli* seconds order is required and speed-up is prevented. On the other hand,dynamic reconfiguration changes circuit construction at one clock cycle during operation without stoppingcircuits Thus,dynamic reconfiguration enables reconfiguration of circuits in a few *nano* seconds. In this paper, a dynamically reconfigurable circuit for floating-point arithmetic is proposed. The arithmetic circuit consists of two single precision floating-point arithmetic circuits. As shown in Fig. 1, the proposed circuit performs double precision floating-point arithmetic by reconfiguration

III. FLOATING-POINT BASED ON IEEE754 FORMAT

We use IEEE754 format which is adopted most widely at the calculation of the floating-point .It decides

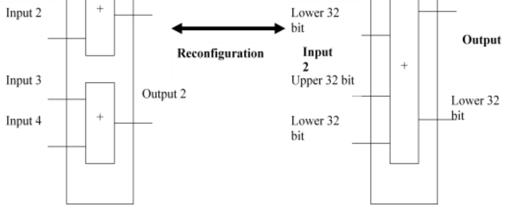

Fig. 1 Dynamic reconfiguration of arithmetic circuit

The detail format is shown in Table I.

Sign	Exponent	Significant

Fig. 2 Floating-point format

TABLE I
BIT WIDTH OF
EACH PRECISION

Symbol	Single precision	Double precision
Total	3	6
Sign	2	4
Expone		b
nt	b	it
Signific	i	1
ant	t	b

The floating point number is converted into a binary number as follows (single precision),

$(-1)^C \times 1.S \times 2^{E-127}$ (single precision) (1)
$(-1)C \times 1.S \times 2^{E-1023}$ (double precision) (2)

where, C is sign part, E is exponent part, and S is significant part. The treatments in this paper are shown below.

-Rounding process uses Unbased.
(Round to nearest even digit in the required position)
- Un-normalized number are not used.
-Exception processes are overflow, underflow, infinity, and NaN.

IV. DYNAMICALLY RECONFIGURABLE FLOATING-POINTARITHMETIC CIRCUIT

The floating-point arithmetic circuits contain three operations; sign part, exponent part, and significant part. In addition, normalization, rounding and exception handling are necessary. In this paper, the proposed circuit contains four floating-point arithmetics; addition, subtraction, multiplication, and division. Each circuit reconfigures two modes as indicated by the following:
(i) One double precision floating-point arithmetic,
(ii)two parallel operations of single precisionfloating-pointarithmetic. The reconfiguration of circuit is controlled by “*fmt*” (format signal). The input “*fmt*” selects single precision (*fmt* = '0') or double precision (*fmt* = '1').

A.Floating-point adder and subtracter Fig. 3 shows the floating-point adder and subtracter. This circuit calculates

Input_A + Input_B = Output,
or
Input_A - Input_B = Output.

The input “*sel*” selects addition (*sel* = '0') or subtraction (*sel* ='1'). The symmetrical circuit blocks perform single precision arithmetic in parallel. Two right shift circuits located at the center operate as one double precision circuit.

B. Floating-point multiplier

Fig. 4 shows the floating-point multiplier, which calculates

Input_A × Input_B = Output.

The adder and register located at the center add partial products of single precision for double precision. Two clock cycles are necessary for double precision arithmetic.

C.Floating-point divider

Fig. 5 shows the floating-point divider. This circuit calculates

Input_A / Input_B = Output.

This divider uses the recovery method.

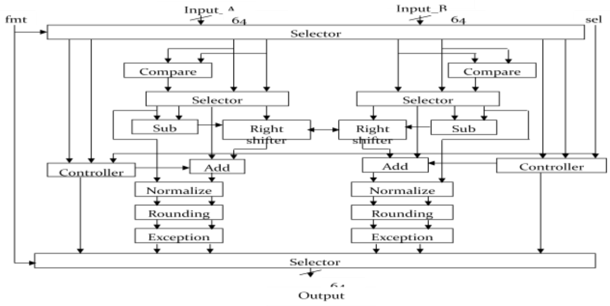

Fig. 3 Floating-point adder and subtracter

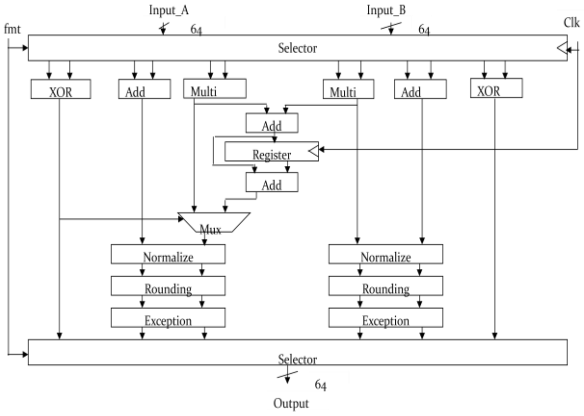

Fig. 4 Floating-point multiplier

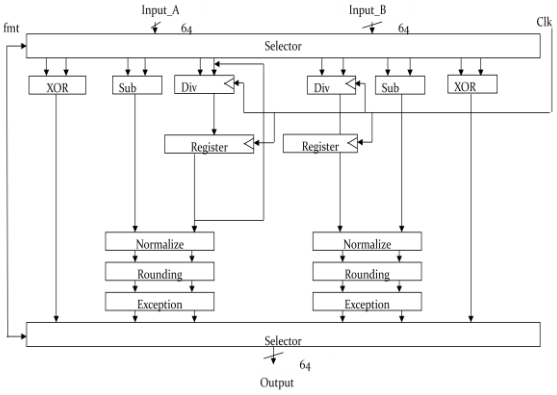

Fig. 5 Floating-point divider

V. PROCESSOR ARCHITECTURE

The instruction set architecture of the proposed processor conforms to MIPS-I (MIPS Technologies, Inc.) subset that is a typical microprocessor .MIPS-I has 32-bit RISC architecture. The pipeline consists of five stages (IF (instruction fetch), ID (instruction decode), EX (execution), MEM (memory access), and WB (write back)). The registers are PC (program counter), 32 GPR (general purpose register),and HI/LO (multiplication and division register). The size of instruction memory and data memory is 1 (KB). The integer arithmetic circuit has ALU (arithmetic logical unit) and MDU (multiplication and division unit). In addition, it has two CP (coprocessors). CP0 controls status management, exception, and interrupt. CP1 controls floating-point arithmetic. CP1 controls dynamically reconfigurable arithmetic circuit designed in this paper. Fig. 6 and Fig.7 show the block diagram of processor and co-processor, respectively. The "DR FPU" in Fig.7 means the dynamically reconfigurable floating-point unit. Table II summarizes the implemented instructions which cover 80% of instruction set for MIPS-I processor.

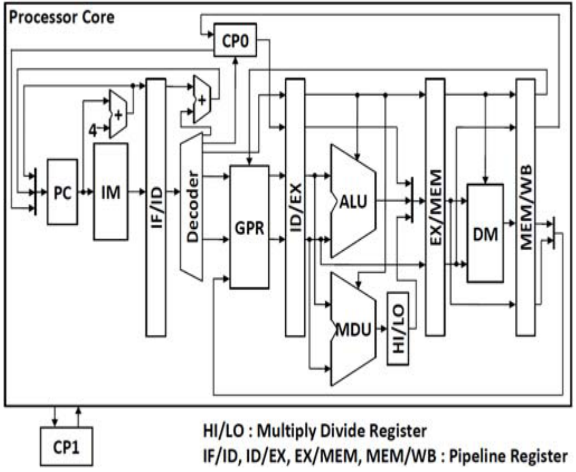
Processor Core
CP0
4
+
PC
IM
IF/ID
Decoder
GPR
ID/EX
ALU
MDU
HI/LO
EX/MEM
DM
MEM/WB
CP1
HI/LO : Multiply Divide Register
IF/ID, ID/EX, EX/MEM, MEM/WB : Pipeline Register

Fig. 6 Block diagram of processor core

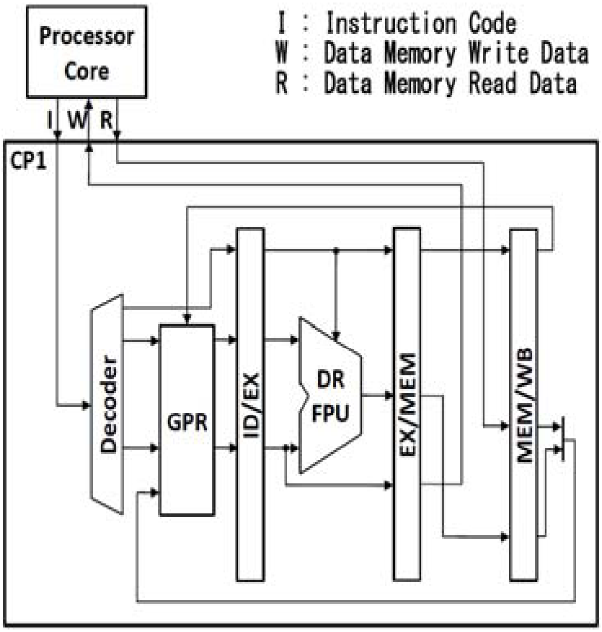

Fig. 7 Block diagram of floating-point co-processor

INSTRUCTION LIST

R (Register):

ADD, ADDU, SUB, SUBU, MULT, ULTU, DIV, DIVU, SLT, SLTU, AND, OR, NOR, XOR, MFHI,MFLO, MTHI, MTLO, SLL, SLLV, SRL, SRLV, SRA, SRAV, JR, JALR

I (Immediate):

LW, SW, LB, LBU, SB, LH, LHU, SH, BEQ, BNE, BLEZ, BGEZ, BLTZ, BGTZ, GEZAL, BLTZAL,ADDI, ADDIU, SLTI, SLTIU, ANDI, ORI, XORI

J (Jump):

J, JAL

floating-point:

LWC1, SWC1, ADD.fmt, SUB.fmt, MUL.fmt,DIV.fmt, ABS.fmt, MOV.fmt, NEG.fmt, C.EQ.fmt,C.LT.fmt, C.LE.fmt, BC1T, BC1F
(fmt: S = Single-precision, D = Double-precision, P = Pair Single-precision)

VI. VERIFICATION

A. Simulation

The dynamically reconfigurable arithmetic circuit was synthesized using ISE11.2 CAD software (Xilinx Inc.). It was simulated using ModelSim XE III 6.4b logic simulator (Mentor Graphics Corp.). The circuit was verified with a sample program that solves simultaneous linear equation by Gauss-Jordanian method in double precision floating-point.

In the simulation, the following simultaneous linear equations are solved.

$$2x + 3y + z = 4$$
$$4x + y - 3z = -2$$
$$-x + 2y + 2z = 2$$

Then, solution matrix is stored in data memory. The results in double precision floating-point format are obtained as shown in table 3. For example,

$$3FF0\ 0000\ 0000\ 0000_{(16)} \qquad (5)$$

equals

$$0011\ 1111\ 111\ 1\ 0..\ .0_{(2)} \qquad (6)$$

Then, we have

$$C = 0 \qquad (7)$$

$$£ = 011\ 1111\ 1111(2) = 1023_{(10)} \qquad (8)$$

$$S = 0 \qquad (9)$$

Therefore

$$(-l)^{0}xl.0x2^{1023-1023} =1 \qquad (10)$$

Fig.8 shows the part of sample program. The object program size is 456 byte.

Fig. 9 shows the simulation result. The result showed that the value of data memory is equal to the solution in Table III. Thus, the correct operations of proposed circuit were verified.

TABLE III

FLOATING-POINT VALUE OF SOLUTION

row	column	value	Floating-point (hex)
	1	1.0	3FF00000 00000000
	2	0.0	00000000 00000000
	3	0.0	00000000 00000000
	4	2.0	40000000 00000000
	1	0.0	00000000 00000000
	2	1.0	3FF00000 00000000
	3	0.0	00000000 00000000
	4	-1.0	BFF00000 00000000
	1	0.0	00000000 00000000
	2	0.0	00000000 00000000
	3	1.0	3FF00000 00000000
	4	3.0	40080000 00000000

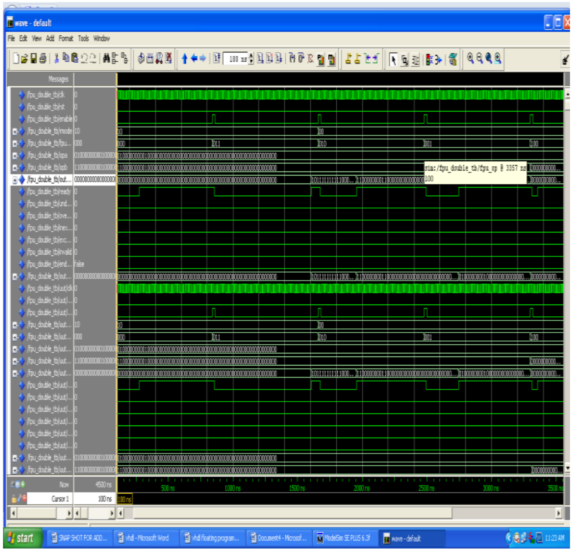

Fig. 9 Simulation result for processor reconfiguration

Table IV shows the circuit size of proposed processor and the number of total clock cycles of verification program. The conventional floating-point arithmetic circuit was measured for the comparison. Table IV showed that the proposed circuit was reduced the circuit size more than the conventional circuit, with a very little number of clocks (1.1%).

TABLE IV

	Static configuration	Dynamic configuration	Reduction rate
slice	8,467	8,009	-5.3%
Flip flop	3,347	3,221	-3.8%
Clock cycles	1,628	1,646	1.1%

VII. CONCLUSION

This paper proposed a general-purpose processor with dynamically reconfigurable floating-point arithmetic. The correct operation was verified by simulations and experiments. In addition, the implementation result showed that the proposed circuit could reduce the circuit size, with a very little number of clocks.

REFERENCES

[1] Yukinari Minagi , Akinori Kanasugi , "A Processor With Dynamically Reconfigurable Circuit For Floating-Point Arithmetic". World Academy of Science, Engineering and Technology 68, 2010.

[2] H. Shimada, Y. Hayakawa and A. Kanasugi, "An Architecture of Dynamically Reconfigurable Arithmetic Circuit", Proc. of 2009 Int. Conf. on Electronics Packaging, pp.963-966, 2009.

[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] T. J. Todman, et al, "Reconfigurable computing: architectures and design methods", IEEE Proc.-Computers & Digital Techniques, vol. 152, no. 2, pp. 193 - 207, 2005.

[5] Xilinx, San Jose, CA, "Virtex-4 family overview," Literature NumberDS112, Feb. 2007, vol. 1.5 [Online].

[6] H. Zhang et al., "A 1-V heterogeneous reconfigurable DSP IC for wireless baseband digital signal processing," IEEE J. Solid-State Circuits, vol. 35, no. 11, pp. 1697–1704, Nov. 2000.

[7] P. Heysters and G. Smit, "Mapping of DSP algorithms on the MOTIUM architecture," in Proc. Int. Parallel Distrib. Process.

Symp.,2003, pp. 180–185.

[8] PACT Informations Technologie, “GmbH the XPP white paper,” Mar.2002, vol. 2.1.

[9] B. Plunkett and J. Watson, “Adapt2400 ACM architecture overview,” QuickSilver Technology, Inc., San Jose, CA, 2004 [Online].Available:http://www.qstech.com/pdfs/Adapt2000_Overview.pdf

[10] S. C. Goldstein et al., “PipeRench: A reconfigurable architecture and compiler,” Computer, vol. 33, no. 4, pp. 70–77, Apr. 2000.

[11] J. Becker, T. Pionteck, C. Habermann, and M. Glesner, “Design and implementation of coarse-grained dynamically reconfigurable hardware architecture,” in Proc. IEEE Computer Soc. Workshop VLSI, 2001, pp. 41–46.

[12] M. Myjak and J. Delgado-Frias, “A two-level reconfigurable architecture for digital signal processing,” Microelectron. Eng., vol. 84, no. 2, pp. 244–252, Feb. 2007.

[13] M. J. Myjak, “A medium-grain reconfigurable architecture for digital signal processing,” Ph.D. dissertation, School Elect. Eng. Comput. Sci., Washington State Univ., Pullman, May 2007.

[14] M. J. Myjak, J. K. Larson, and J. G. Delgado-Frias, “Mapping and performance of DSP benchmarks on a medium-grain reconfigurable architecture,” in Proc. Int. Conf. Eng. Reconfigurable Syst. Algorithms, 2006, pp. 123–129.

[15] M. J. Myjak and J. G. Delgado-Frias, “Medium-grain cells for reconfigurable DSP hardware,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 54, no. 6, pp. 1255–1265, Jun. 2006.