A PROGRAMMABLE PROCESSOR WITH MULTIPLE FUNCTIONAL UNITS AND BANKED REGISTERS FOR GENERAL PURPOSE NUMERICAL PROCESSING

Takafumi Morifuji, Yoshinori Takeuchi, and Masaharu Imai

Department of Informatics and Mathematical Science
Graduate School of Engineering Science, Osaka University,
1-3 Machikane-yama, Toyonaka, Osaka 560-8531, Japan
peasv@vlsilab.ics.es.osaka-u.ac.jp

ABSTRACT

We present an architecture of a General Purpose Numerical Processor(GPNP). The processor with this architecture is capable of running a wide variety of numerical processings and digital signal processings with its programmability. Flexibility and high-performance are achieved by multiple functional units and their data transfer parallelism. The prototype of a GPNP with five functional units can operate with 33-MHz clock frequency by simulation and its size corresponds with 230-kTr. and 34.5-kbytes on-chip memory.

1. INTRODUCTION

Advance of computer network technologies accelerate the wide use of multimedia data. Multimedia data consist of text data, audio data, image data, and the other different format digital data. Applying these digital data requires a large amount of numerical processing as a result. Moreover, high-speed calculation is required to deal with audio and image signals under the execution time constraint. Conventionally, numerical processings were executed on general purpose CPUs and DSPs (Digital Signal Processors) as software.

DSP can be classified into "application specific DSP" and "general purpose DSP." The former is designed for a specific digital signal processing, and the latter can deal with some domain of digital signal processings. Recent advance of semiconductor technologies enable us to integrate a much larger scale DSP with higher performance[1]. Considering general use, programmable DSP and reconfigurable DSP are also proposed[2][3][4].

In this paper, we propose the general purpose processor architecture for numerical processing indispensable to multimedia signal processing. In the next section, some examples of application are introduced and these features are considered for our processor architecture. Then, a General Purpose Numerical Processor(GPNP) architecture is explained being divided into each component unit. Next, a prototype of the GPNP is introduced. Finally, this paper is concluded with the discussion of the effectiveness of the GPNP.

2. NUMERICAL PROCESSING

In this section, some applications of numerical processing are introduced. The processor architecture is considered taking account of these features.

2.1. Matrix Operation

Matrix operation is utilized in manifold applications such as image signal processings. $N \times N$ matrix operation can be formulated as follows:

$$X(k) = A\sum_{n=0}^{N-1} x(n)a_{kn} + B\sum_{n=0}^{N-1} y(n)b_{kn} \qquad (1)$$

where $k = 0, 1, \cdots, N-1$, $x(n)$ and $y(n)$ denote an N-dimensional column vector, a_{kn} and b_{kn} denote an element of $N \times N$ matrix, and A, B denote a scalar multiplier, respectively.

Shown in (1), each multiply-accumulate operation for a element in the matrix is parallel-executable.

2.2. DCT

DCT (Discrete Cosine Transform) has its many application not only for audio and image compression but also for digital filters and filter banks. Each element of $N \times N$ transform matrix for N-point DCT-II, which is often utilized in image compression, is formulated as follows:

$$\begin{aligned} [C_N^{II}]_{m,n} &= \sqrt{\frac{2}{N}} k_m \cos\frac{m(n+\frac{1}{2})\pi}{N}, \\ k_j &= \begin{cases} 1, & j = 1, 2, \cdots, N-1 \\ \frac{1}{\sqrt{2}}, & j = 0 \end{cases} \end{aligned} \qquad (2)$$

where $m, n = 0, 1, \cdots, N-1$. DCT algorithms can be expressed using matrix operations as (2) shows. In the matrix implementation of DCT, an operation for an element is independent from the other operation as is matrix operation.

2.3. FIR Filter

FIR(Finite Impulse Response) filter is a kind of digital filter, and utilized in digital signal processings such as audio, communication, and image processing. An algorithm of N-tap FIR filter is formulated as following expression:

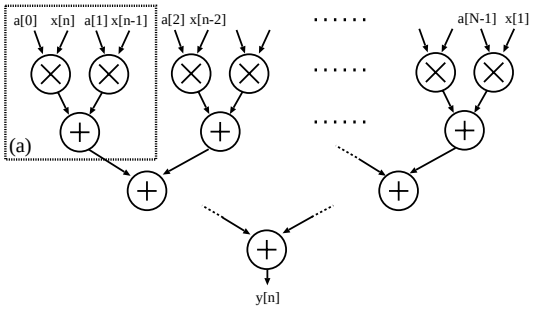

Figure 1: A Signal Flow of FIR Filter.

$$y[n] = \sum_{i=0}^{N} a[i]x[n-i] \tag{3}$$

where $x[n]$, $y[n]$ indicate input and output of FIR filter in time n, and each $a[i]$ indicates the filter coefficient, respectively.

A signal flow of FIR filter based on (3) is illustrated in Figure 1. Each term in (3) is independent from the other terms. Let the computation of a sum-of-product term corresponded to (a) in Figure 1 be defined as the basic block. The each basic block can be calculated in parallel within 2 clock cycles, for example, using multiple MAC (Multiply-ACcumulate) unit which can calculate a sum-of-product term in 1 clock cycle. Computational sequences of 16-tap FIR filter using four MAC units are shown in Table 1. The parallelism of the calculation in Table 1 is about the number of MAC units: These applications have the high potential parallelism of calculation.

2.4. Considerations

Implementing numerical processing, following processor architectures are considered: (a) consists of multiple functional units, ROM, RAM, and control unit, (b) with subordinate/parallel connection of functional units, and (c) considered shared functional units. Software oriented implementation (a) can have the more general purpose facility than hardware solutions. However, the amount of hardware including instruction memory increases as a result. On the other hand, hardware oriented implementation (b) and (c) can customize its architecture for its specific application at the expense of general purpose facility.

Considering the above-mentioned examples, an architecture model for numerical processing can be implemented using the combination of following units: Functional Unit (FU), Control Unit, Memory Unit, and Register Unit, where FU includes MAC unit and ALU, and Control Unit includes PCU (Program Counter Unit) and MCU (Memory Control Unit).

In numerical processing for multimedia, it is also important to perform multiply-accumulate operation in as few cycles as possible. Improving the performance of MAC unit, we can adopt following approaches: (1) improves clock frequency using pipelined MAC unit[5] and (2) utilizes multiple MAC units for parallel operation[1][6]. In the approach (1), whole units in the processor must be operated at higher clock frequency. On the other hand, the approach (2) needs larger chip area assigned to multiple MAC units

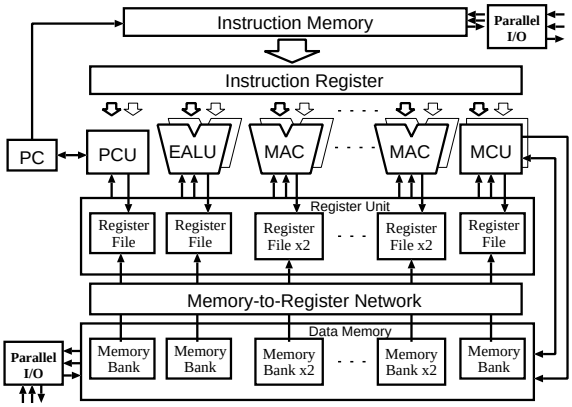

Figure 2: The Organization of the GPNP.

than the approach (1). However, the approach (1) requires higher-speed on-chip memories and larger power consumption. The approach (2) can be adopted as acceptable for the total chip size as the area of MAC units is, and would be effective to numerical processing as was shown in Table 1

In the next section, our processor architecture based on these considerations is explained.

3. ARCHITECTURE DESIGN

General purpose DSP is based on micro-processor with some peripherals for digital signal processing, such as a few independent memory banks, and hardware loops, addressing units[7][8][9].

We propose the architecture based on general purpose DSP taking account of general use to operate numerical processing.

3.1. Structure of Processor

Our processor architecture, GPNP, consists of multiple MAC unit, EALU (extended ALU), PCU, MCU, Memory Unit, and Register Unit. Figure 2 illustrates the organization of the GPNP.

The GPNP has following features: (1) Control Unit adopts microprogram control system for generality. (2) The GPNP can calculate double format or extended format operation using multiple FU. (3) Data transfer parallelism can be achieved by the data/instruction memory with memory bank system and Register Unit with multiple register files.

The functionality of the GPNP can be parameterized as follows:
(1)bit-width of I/O port, (2)basic internal precision,
(3)number of MAC units, (4)capacity of memory bank,
(5)number of memory banks, (6)number of registers, and
(7)number of register files.

These parameters will be defined according to the area constraint, the execution time constraint, and target applications.

3.2. Functional Unit

The structure of MAC unit utilized in the GPNP is shown in the left part of Figure 3. MAC unit consists of following components: N-bit fixed-point multiplier (MUL) with 2n-bit output can execute multiplication within 1 clock cycle, which enables to execute MAC

Table 1: Computational Sequences of 16-tap FIR Filter Using Multiple MAC Units.

time	MAC3	MAC2	MAC1	MAC0
T				$a[0]x[t]$
$T+1$			$a[5]x[t-5]$	$\sum_{i=0}^{1} a[i]x[t-i]$
$T+2$		$a[9]x[t-9]$	$\sum_{i=5}^{6} a[i]x[t-i]$	$\sum_{i=0}^{2} a[i]x[t-i]$
$T+3$	$a[13]x[t-13]$	$\sum_{i=9}^{10} a[i]x[t-i]$	$\sum_{i=5}^{7} a[i]x[t-i]$	$\sum_{i=0}^{3} a[i]x[t-i]$
$T+4$	$\sum_{i=13}^{14} a[i]x[t-i]$	$\sum_{i=9}^{11} a[i]x[t-i]$	$\sum_{i=5}^{8} a[i]x[t-i]$	$\sum_{i=0}^{4} a[i]x[t-i]$
$T+5$	$\sum_{i=13}^{15} a[i]x[t-i]$	$\sum_{i=9}^{12} a[i]x[t-i]$	$\sum_{i=0}^{8} a[i]x[t-i]$	
$T+6$	$\sum_{i=13}^{16} a[i]x[t-i]$	$\sum_{i=0}^{12} a[i]x[t-i]$		
$T+7$	$\sum_{i=0}^{16} a[i]x[t-i]$			

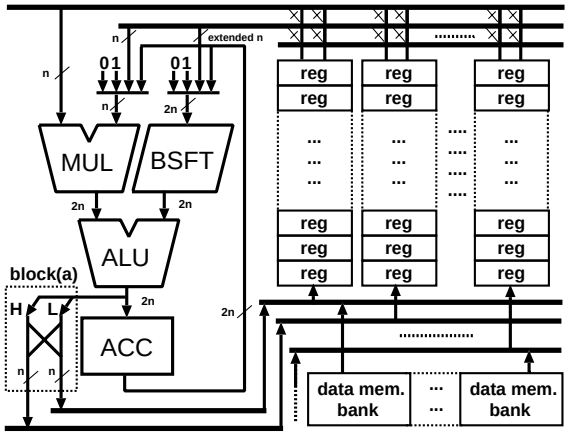

Figure 3: The Structure and Peripherals of MAC unit.

operation in 1 clock cycle. 2n-bit ALU can also execute saturation arithmetic. Saturation arithmetic can decrease the number of comparison and branch instructions. 2n-bit barrel shifter (BSFT) can shift its input data to left/right direction ranging from 1-bit to 2n-bit, which can be utilized in rounding and normalizing.

2n-bit accumulator (ACC) can vary its output data in format of "HH", "LH", "HL", and "LL", where "H" indicates upper n-bit of input data, and "L" indicates lower n-bit of input data, hence the destination of output data in data transfer can be decided flexibly as is shown in block(a) of Figure 3. On the other hand, EALU has an n-bit input and have the same facility as ALU in MAC unit. The GPNP can provide flexible calculation in parallel at the other precision format, such as double format, controlling multiple FU's by its microprogram.

3.3. Data Transfer

Data transfer in the GPNP is classified into (a) between FU and Register Unit, and (b) from Memory Unit to Register Unit. The data transfer between Memory Unit and the outside of GPNP is achieved through a parallel I/O.

In the case (a), MAC unit can write 2n-bit data to two n-bit register files. MCU, PCU, and EALU can write n-bit data to one n-bit register file. In reading data from register file to FU, FU can read data from any register files, which keeps flexibility of data transfer: The GPNP adopts heterogeneous register system in writing data to register file as is shown in Figure 3. In the case (b), other units can read/write from/to data memory through MCU fundamentally, which means the GPNP adopts load/store architecture with the exception of following data transfer system. The GPNP can transfer data directly from data memory to Register Unit by no way of MCU decreasing the data transfer overhead: Starting to operate numerical processing, the amount of data such as coefficients needs not to be preloaded in Register Unit. Moreover, each data transfer from data memory bank to register file can be executed concurrently.

The GPNP will achieve high performance by implementation of this data transfer parallelism.

4. PROTOTYPING

In this section, we introduce the prototyping of the GPNP for investigating effectiveness of proposed architecture.

4.1. Preliminary

We prototype the GPNP with following parameters:

(1)bit-width of I/O port	16-bit,
(2)basic internal precision	32-bit,
(3)number of MAC units	4,
(4)capacity of memory bank	2-kbyte,
(5)number of memory banks	16,
(6)number of registers	12, and
(7)number of register files	11.

Bit-width of I/O port is 16-bit considering common audio signal processing. Basic internal precision, the bit-width of output in MAC unit, is 32-bit.

4.2. Features of the Prototype

All the functionality of the GPNP is described in VHDL for short design time: Control Unit and FU is described in RT level, and Memory Unit is described in behavior level for simulation. The amount of all the description is about 17,000 lines. Control Unit and FU is logic-synthesized. These logic-synthesized netlists are connected with Memory Unit in layout level using COMPASS DA

Table 2: Features of the Prototype.

Technology	$0.6\mu m$ CMOS 3LM
Chip Scale	230-k Tr.+34.5-kbytes memory
Clock Frequency	33-MHz
Chip Size & Package	$9mm \times 9mm$, 208-pin QFP

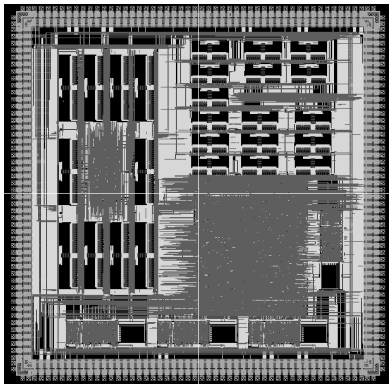

Figure 4: Layout Pattern of the Prototype.

TOOL. The features of the prototype is shown in Table 2. The layout pattern of the prototype is illustrated in Figure 4. We confirmed the prototype of the GPNP could operate at not less than 33-MHz clock frequency by simulation.

4.3. Prototyping Results

The execution time comparison for 256-point complex radix-2 FFT is illustrated in Figure 5. The data needed to execute the application assume to be stored in the data memory at the initial status of the application. Higher-speed execution time can be achieved by parallel-executable MAC units than the other DSPs relatively.

However, the GPNP architecture needs large amount of memory capacity, which means the area ratio of on-chip Memory Unit in the prototype is dominant. The number of global wires, in specially between Memory Unit and Register Unit, is very large because of the generic network between them. It is also one of reasons that the prototype used $0.6\mu m$ CMOS with 3 layer metal. This problem is expected to be solved using the deep-submicron technology with more multiple wiring layer.

Further, in the actual design, more deliberated parameter setting and the subset of our proposed architecture will be required considering the trade-off between performance and area.

5. CONCLUSIONS

In this paper, we proposed an architecture of a General Purpose Numerical Processor(GPNP). The processor with this architecture is capable of running a wide variety of numerical processings and digital signal processings with its programmability, multiple FU's, and their data transfer parallelism. The prototype of a GPNP can operate at 33-MHz clock frequency by simulation and achieve high-speed execution of 256-point complex FFT.

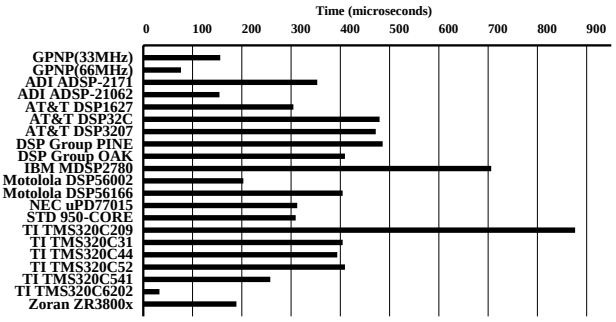

Figure 5: Execution Time for 256-Point FFT[10].

6. ACKNOWLEDGMENT

The authors would like to express their special thanks to Semiconductor Technology Academic Research Center (STARC) and the members of VLSI system design laboratory in Osaka Univ. for their helpful comments and discussions. The VLSI chip in this study has been fabricated in the chip fabrication program of VLSI Design and Education Center(VDEC), the University of Tokyo with the collaboration by Rohm Corporation, and Toppan Printing Corporation.

7. REFERENCES

[1] K.Nadehara, H.Lieske and I.Kuroda, "Software MPEG-2 Video Decoder on a 200-MHz, Low Power Multimedia Micoroprocessor," *Proc. of ICASSP'98*, vol.V, pp.3141-3144, 1998.

[2] An-Yeu Wu, K.J.Ray Liu, "System Architecture of an Adaptive Reconfigurable DSP Computing Engine," *IEEE Trans. on Circuits and Systems for Video Technology*, pp.54-73 1998.

[3] T.C.Denk, C.J.Nicol, P.Larsson, K.Azadet, "Reconfigurable Hardware for Efficient Implementation of Programmable FIR Filters,"*Proc. of ICASSP'98*, vol.V, pp.3005-3008, 1998.

[4] M.M.Cai, D.A.Luthi, P.A.Ruetz, and P.H.Ang, "A 40 MHz programmable and reconfigurable filter processor," *Proc. of IEEE Custom Integrated Circuits Conf.*, pp.13.3.1-13.2.4, 1990.

[5] H.Kabuo, M.Okamoto, I.Tanaka, H.Yasoshima, S.Marui, M.Yamasaki, T.Sugimura, K.Ueda, T.Ishikawa, H,Suzuki, and R.Asahi, "An 80-MOPS-peak high-speed and low-power-consumption 16-b digital signal processor," *IEEE J. of Solid-State Circuits*, vol.31, no.4, pp.494-503, Apr.1996.

[6] T.Shiraishi, K.Kawamoto, K.Ishikawa, H.Sato, F.Asai, E.Teraoka, T.Kengaku, H.Takata, T.Tokuda, K.Nishida, and K.Saitoh, "A 1.8V 36mW DSP for the half-rate speech CODEC," *Proc. of IEEE Custom Integrated Circuits Conf.*, pp.371-374, 1996.

[7] P.Lapsley, J.Bier, A.Shoham, and E.A.Lee, *DSP Processor Fundamentals: Architectures and Features*, Berkeley Design Technology,Inc., 1997.

[8] V.K.Madisetti, *Digital Signal Processors*, IEEE Press, 1995.

[9] M.Dolle, S.Jhand, W.Lehner, O.Muller, and M.Schlett, "A 32-b RISC/DSP Microprocessor with Reduced Complexity," *IEEE J. of Solid-State Circuits*, vol.32, no.7, pp.1056-1066, Mar.1997.

[10] "Evaluating DSP Processor Performance," Berkeley Design Technology, Inc., 1996, which is available on the world-wide web at http://www.bdti.com/articles/wpeval.htm.