A Novel Register Organization for VLIW Digital Signal Processors

Tay-Jyi Lin, *Chen-Chia Lee*, *Chih-Wei Liu*, and *Chein-Wei Jen*

Department of Electronics Engineering
National Chiao Tung University, Taiwan

Abstract

This paper presents a novel register organization for VLIW DSPs. The simulation results show the performance of a DSP with the proposed register file is comparable with state-of-the-art DSPs. However, the proposed register file can save 89.7% area of a conventional centralized one, while reducing its access time by 68.6%.

1. Introduction

Today's wireless and multimedia applications require multi-billion operations per second, and owing to the advances in IC technology, the performance can be easily achieved by fabricating tens or hundreds of functional units (FU) on a hundred-MHz or GHz chip. The ***microprocessors*** have a more efficient and direct data exchange mechanism among the parallel FUs through the register file (RF) than the ***multi-processors***, where a monolithic and centralized RF provides storages for and interconnects to each FU in a general and homogeneous manner. In other words, each FU can directly read from or write to any register in a centralized RF. For N concurrent FUs, the silicon area, delay, and power dissipation of a centralized RF grow as N^3, $N^{3/2}$, and N^3 respectively [1]. As a result, the centralized RF will soon dominate the silicon area, the delay, and the power dissipation of the microprocessors as the number of FUs increases. The RF complexity can be significantly reduced with small performance penalty by restricting the communication between FUs and registers, such that each FU can only read from and write to a limited set of registers. We propose a novel ring-structure RF, where the centralized RF is partitioned into $2N$ sub-blocks, and each FU can simultaneously access two sub-blocks. One sub-block is private (i.e. dedicated to an FU) and the other is dynamically mapped to facilitate the inter-FU communication. In the proposed ring-structure RF, all sub-blocks require access ports only for a single FU, and the dynamically-mapped sub-blocks are organized as a ring to reduce the control overheads. We have integrated the proposed ring-structure RF in a four-way VLIW digital signal processor (DSP) to evaluate the performance penalty due to the access restrictions, where the port mapping of the shared register sub-blocks is under programmers' control. Our instruction set simulation shows that this DSP core has comparable performance with state-of-the-art high-end DSPs in the cycle counts. Moreover, the synthesis results of the proposed ring-structure RF saves 89.7% silicon area of a centralized RF while reducing the access time by 68.6%.

The rest of this paper is organized as follows. Section 2 gives the taxonomy of partitioned RF and summarizes the communication mechanisms among the RF partitions. Section 3 describes our proposed ring-structure RF and the VLIW DSP that integrates the RF. Both the performance evaluation and the silicon implementation of the proposed RF are available in Section 4. Finally, Section 5 concludes this paper and outlines our future research directions.

* This work was supported by the National Science Council, Taiwan under Grant NSC 93-2220-E-009-017

2. Taxonomy of Partitioned Register File

Fig. 1 shows a centralized register file (RF) and a register cell. The RF complexity depends on the number of registers and the number of access ports in the RF. Informally, an access port requires a word-line for control and a bit-line for data, and the RF grows quadratically with the number of access ports. Besides, both the number of registers and the number of access ports grow linearly with the number of functional units (FUs, either an arithmetic unit or a load/store unit). Therefore, the area of a centralized RF grows as the cubic of the number of FUs, and the access time and the power consumption also grow dramatically with the number of FUs, too [1]. The centralized RF will soon become the bottleneck of the microprocessor designs, as the number of integrated FUs grows rapidly.

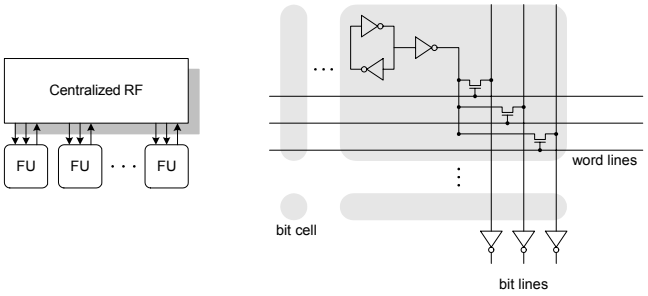

Fig. 1. Centralized register file

2.1 Register File Partitioning

A well-known technique to reduce the RF complexity is through partitioning. For example, we can implement a 12-port RF with four 3-port register banks [2] as shown in Fig. 2, where the port conflicts can be resolved either in software or hardware. The twelve virtual ports are dynamically mapped to the four banks, each of which has three physical ports respectively, through a crossbar router. The interconnection complexity between the RF partitions can be significantly simplified by exploring the spatial locality of computations. That is, we can divide the FUs into independent groups (or clusters [3]), where each group accesses its own RF partition and the inter-partition communication [4] is minimal. Fig. 3(a) shows the concept for FU grouping. A special case is to separate the load/store (L/S) units from the arithmetic units (AU) as Fig. 3(b), and the RF partition for the L/S units can be regarded as an additional memory hierarchy for AU, which is controlled by instructions directly [5].

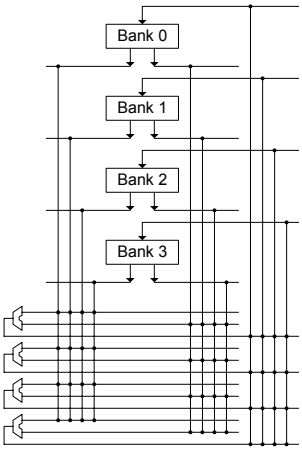

Fig. 2. Banked register file

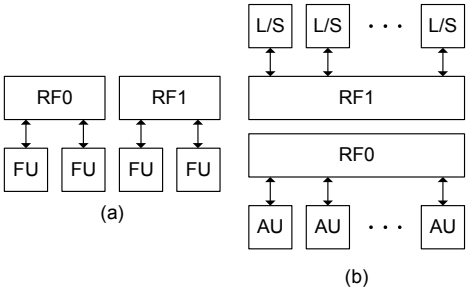

Fig. 3. (a) Clustered, and (b) hierarchical register files

2.2 Communication between RF Partitions

There are three ways for inter-partition communication. The first is through explicit copy instructions. It requires some extra ports of the RF partitions in each FU group as Fig. 4. One possible implementation is to use the existing issue slots [6] for copy instructions and thus to reuse the existing input (or output) ports of the RF partitions. Its major drawback is that the FU lies idle while executing the copy instructions. The other implementation is to use dedicated instruction slots at the cost of extra RF access ports [7]. The additional instruction slots may also significantly increase the program size. By the way, for hierarchical RF, the copy instructions can be regarded as special L/S instructions for the AUs to access the cache RF (i.e. the RF partition for the L/S units).

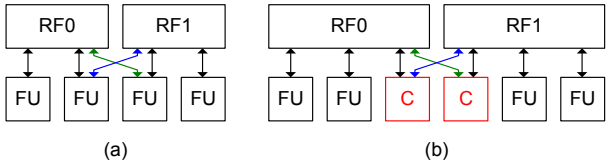

Fig. 4. Copy operations (a) without [6], and (b) with dedicated instruction slots [7]

The second way for inter-partition communication is by extended accesses, where the FU has limited read [8] or write [9] accesses to the RF partitions of other FU groups. Fig. 5 shows the communication scheme via extended reads and writes, where the RF partitions need to support the corresponding read ports with some external interconnection resources and control. The VelociTI architecture (C'6x DSP) [8] from Texas Instruments is one of the most famous examples in this category.

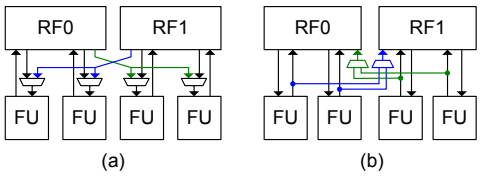

Fig. 5. (a) Extended reads [8], and (b) extended writes [9]

Fig. 6 illustrates the third inter-partition communication method, which is through a shared storage [10]. Each FU group has access ports directly connected to a common storage element, through which their data are exchanged. For hierarchical and clustered RF organizations [5], the cache RF can also function as the shared storage for inter-partition communication. Register permutation [11] is a special case of this approach, where each FU group is able to access an ***exclusive*** bank of the shared storage as depicted in Fig. 7. Note the hierarchical RF with register permutation for inter-partition communication works just like a ping-pong buffer.

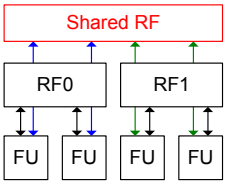

Fig. 6. Shared registers

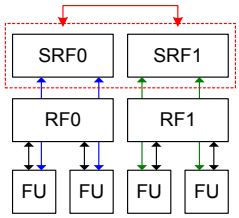

Fig. 7. Register permutation

3. Ring-Structure Register File

In this paper, we propose a ring-structure register file (RF), where a centralized RF for *N* functional units (FUs) is partitioned into 2*N* sub-blocks. Each FU can simultaneously access two sub-blocks, one of which is private (i.e. dedicated to the FU) and the other is dynamically mapped to facilitate inter-FU communication. Thus, each register sub-block only requires access ports for a single FU. Fig. 8 shows a 4-way VLIW digital signal processor (DSP) with the proposed ring-structure RF. We can imagine each of the four FUs is a RISC processor with an independent 16-element RF. Each RF is partitioned into a private and a dynamically-mapped sub-block with 8 registers and an FU has exclusive accesses to two sub-blocks with total 16 registers at any time. The four dynamically-mapped sub-blocks are concatenated as a ring with only four possible mappings to reduce the context for dynamic mapping. A 2-bit offset is attached to each instruction word to specify the mapping style, and the instruction dispatcher will take care of this and make it

completely transparent to the FUs. In our VLIW DSP, each register sub-block has two read and two write ports respectively (i.e. total 4 access ports). The ring registers (i.e. the dynamically-mapped sub-blocks) are identical and each sub-block has eight (R8~R15) 32-bit elements. The private register sub-blocks of the load/store (L/S) units have eight (R0~R7) 32-bit elements for general-purpose uses and memory addresses, while those of the ALU/MAC units (arithmetic units; AU) are composed of eight (R0~R7) 40-bit accumulators.

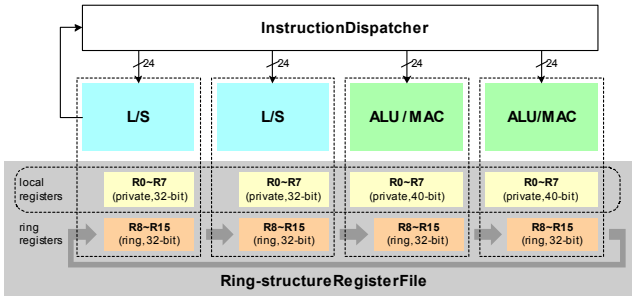

Fig. 8. A 4-way VLIW DSP with the proposed register file

```
1   0; MOV r0,COEF;        MOV r0,COEF;        MOV r0,0;       MOV r0,0;
2   0; MOV r1,X;           MOV r1,X+1;         NOP;            NOP;
3   0; MOV r2,Y;           MOV r2,Y+2;         NOP;            NOP;
4      RPT 512,8;
5   0; LW_D r8,r9,(r0)+2;  LW_D r8,r9,(r0)+2;  MOV r1,0;       MOV r1,0;
6      RPT 15,2;
7   2; LW_D r8,r9,(r0)+2;  LW_D r8,r9,(r0)+2;  MAC_V r0,r8,r9; MAC_V r0,r8,r9;
8   0; LW_D r8,r9,(r0)+2;  LW_D r8,r9,(r0)+2;  MAC_V r0,r8,r9; MAC_V r0,r8,r9;
9   2; LW_D r8,r9,(r0)+2;  LW_D r8,r9,(r0)+2;  MAC_V r0,r8,r9; MAC_V r0,r8,r9;
10  0; MOV r0,COEF;        MOV r0,COEF;        MAC_V r0,r8,r9; MAC_V r0,r8,r9;
11  0; ADDI r1,r1,-60;     ADDI r1,r1,-60;     ADD r8,r0,r1;   ADD r8,r0,r1;
12  2; SW (r2)+4,r8;       SW (r2)+4,r8;       MOV r0,0;       MOV r0,0;
```

Fig. 9. Example: 64-tap FIR filtering

The assembly syntax of our VLIW DSP starts with the ring offset, then followed by the four instructions to form an instruction word – ring offset; instr 0; instr 1; instr 2; instr 3; and the data memory subsystem is 16-bit addressed. Fig 9 shows an example for a 64-tap FIR filter that produces 1,024 outputs. The input data are 16-bit fractional and the output data are 32-bit fixed-point numbers respectively. `RPT` (the repeat instruction; line 4 and 6) is carried out by the instruction dispatcher and consumes no execution cycle of the datapath. Note that only three-level zero-overhead loop nesting is allowed in our implementation. The inner loop (line 7-8) loads four 16-bit inputs and four 16-bit coefficients into two 32-bit `r8` and two 32-bit `r9` respectively with the two L/S units (i.e. `r8`←MEM_{32}`[r0]`, and `r9`←MEM_{32}`[r1]` for each load/store unit), while the four address registers (`r0` and implicitly-specified `r1`) are updated (i.e. `r0`←`r0+2`, and `r1`←`r1+2` for each load/store unit) simultaneously. In the meanwhile, the two AUs perform 16-bit SIMD MAC for four taps (i.e. `r0`←`r0+r8.Hi×r9.Hi`, and `r1`←`r1+r8.Lo×r9.Lo` for each AU). After summing up the 32 32-bit products with 40-bit accumulators, `r0` are `r1` are added together and rounded back to the 32-bit `r8` ring registers. Finally, the two 32-bit outputs are stored in the memory subsystem by the two L/S units via `r8`. In the above 64-tap FIR example, the outer loop (line 5 and line 7-12) produces two filter outputs in 35 cycles. In other words, the proposed VLIW DSP is able to compute 3.66 taps per cycle.

The ring-structure RF is redrawn in Fig. 10(a), where the four FUs communicate with each other by dynamically mapping the ring registers. Note that one L/S unit and one AU can form a computing engine with a ping-pong hierarchical RF as shown in Fig. 10(b) (see Section 2.2), where the private (R0~R7) registers are respectively the address registers for the L/S unit and the accumulators for the AU. Therefore, the proposed 4-way VLIW DSP can be viewed as a processor consisting of two independent clusters, which also communicate via the ping-pong registers.

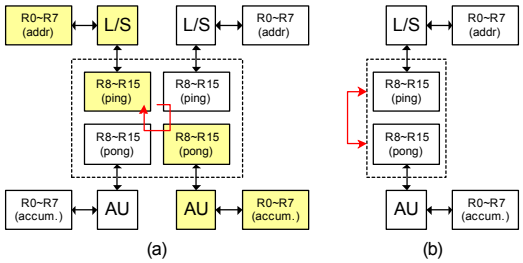

Fig. 10. (a) The ring-structure RF, which can be viewed as two clusters of (b) ping-pong hierarchical RF

4. Simulation & Implementation Results

In this section, we evaluate the performance degradations due to the access restrictions on the proposed ring-structure RF. We have constructed a cycle-accurate instruction-set simulator for the 4-way VLIW DSP. Note that the dynamic port mapping is under programmers' direct control as the ring offset attached to each instruction word. Table 1 summarizes the performance comparisons between state-of-the-art high-performance DSPs and our DSP with the proposed ring-structure RF. The second row shows the number of cycles required for *N*-sample and *T*-tap FIR filtering, which reveals the number of MAC units. The third and the fourth rows list the numbers of cycles to perform the 8-by-8 two-dimensional fast discrete cosine transform (DCT) and the radix-2 256-point fast Fourier transform (FFT) respectively. The fifth row compares the motion estimation under MAE (mean absolute error) criteria, where the block size is 16-by-16 and the search ranges is +/-15 pixels. The maximum ACS (add-select-compare) operations per cycle are given in the last row, which is the kernel of the Viterbi decoding algorithm. Moreover, the numbers in the parentheses indicate the results that consider the load/store overheads when the depth is 16. The performance of TI C'64s is not included because it contains a specific Viterbi coprocessor. Note that these results show the performance of our DSP is comparable with the high-performance DSPs if the dataflow is appropriately arranged.

We have implemented in Verilog RTL the ring-structure RF for the proposed 4-way VLIW DSP and a centralized one with the same number of registers. The two designs are synthesized using Synopsys Design Compiler with the 0.18μm cell library and automatically placed and routed by SoC Encounter in the 1P6M CMOS technology. Table 2 summarizes the results, where our approach reduces the delay and the area by factors of 3.18 and 11.71 respectively. Moreover, we have hand-crafted a full-custom design of the ring-structure RF, and the results are summarized in Table 3 and Fig. 11.

Table 1 Performance comparison of DSP cores

	TI C'55	TI C'64	NEC SPXK5	Intel/ADI MSA	Proposed
FIR	*NT*/2	*NT*/4	*NT*/2	*NT*/2	*NT*/4
DCT	238	126	240	296	112
FFT	4,786	2,403	2,944	3,176	2,340
ME	82,260	36,538	N.A.	90,550	57,731
ACS	1(0.4)	N.A.	1(1)	1(N.A.)	1(0.84)

Table 2 Comparison of cell-based register file implementations

	Centralized RF	Ring-structure RF
Delay	4.90 ns	1.54 ns
Gate count	272K	28K
Area	4,100,000 μm^2	350,000 μm^2

Table 3 Summary of full-custom implementation

Process	TSMC 0.18μm 1P6M CMOS
Delay	2ns
Area	193,600μm^2
Power	40.04mW@500MHz, 1.8V

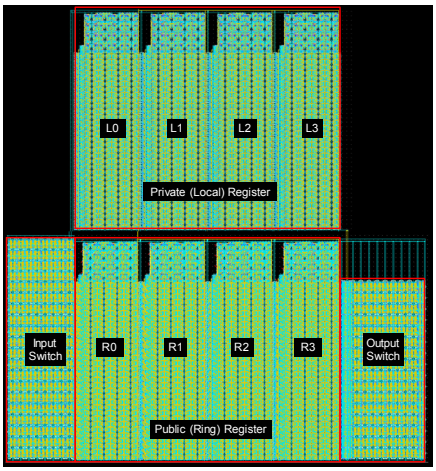

Fig. 11. Layout of the proposed ring-structure register file

5. Conclusions

As the number of concurrent functional units (FUs) in media processors increase continuously to meet the performance needs, the access ports of a centralized register file (RF) multiply rapidly and cannot be efficiently implemented in silicon. For N FUs, the silicon area grows as N^3, the access time as $N^{3/2}$, and the power dissipation as N^3. This paper presents a ring-structure RF, which is composed of $2N$ register sub-blocks, and each sub-block has only access ports for a single FU. The data exchanges among the FUs are explicit through the ring registers and an N-by-N switch network. The proposed ring-structure RF has been successfully integrated into a 4-way VLIW DSP to demonstrate its effectiveness to perform popular DSP kernels. The synthesis result shows that our proposed ring-structure RF saves 89.7% silicon area of the centralized one, while reducing its access time by 68.6%.

We are now developing specific compiler techniques for the proposed ring-structure RF, instead of just posing the restrictions of exclusive accesses on the ping-pong structures. Moreover, we are now studying some ambitious methods to further reduce the number of physical ports by lowering the register access demands in software [15][16]. In the future, we will systematically explore the optimal RF organization in terms of the access time, the cycle count (to run the application software), the silicon area, and the power consumption, based on some RF energy models [17].

References

[1] S. Rixner, W. J. Dally, B. Khailany, P. Mattson, U. J. Kapasi, and J. D. Owens, "Register organization for media processing," in *Proc. HPCA-6*, 2000, pp.375-386

[2] J. H. Tseng and K. Asanovic, "Banked multiported register files for high-frequency superscalar microprocessors," in *Proc. ISCA*, 2003, pp.62-71

[3] R. E. Kessler, "The Alpha 21264 microprocessor," *IEEE Micro*, vol.19, pp.24-36, Mar.-Apr.1999

[4] A. Terechko, E. L. Thenaff, M. Garg, J. Eijndhoven, and H. Corporaal, "Inter-cluster communication models for clustered VLIW processors," in *Proc. HPCA-9*, 2003, pp.354-364

[5] J. Zalamea, J. Llosa, E. Ayguade, and M. Valero, "Hierarchical clustered register file organization for VLIW processors," in *Proc. IPDPS*, 2003, pp.77-86

[6] P. Faraboschi, G. Brown, J. A. Fisher, G. Desoll, and F. M. O. Homewood, "Lx: a technology platform for customizable VLIW embedded processing," in *Proc. ISCA*, 2000, pp.203-213

[7] E. F. Barry, G. G. Pechanek, and P. R. Marchand, "Register file indexing methods and apparatus for providing indirect control of register file addressing in a VLIW processor," *International Application Published under the Patent Cooperation Treaty (PCT)*, WO 00/54144, Mar. 9 2000

[8] *TMS320C6000 CPU and Instruction Set Reference Guide*, Texas Instruments Inc., 2000

[9] K. Arora, H. Sharangpani, and R. Gupta, "Copied register files for data processors having many execution units" U.S. Patent 6 629 232, Sep. 30, 2003

[10] A. Kowalczyk et al., "The first MAJC microprocessor: a dual CPU system-on-a-chip," *IEEE J. Solid-State Circuits*, vol. 36, pp.1609-1616, Nov. 2001

[11] T. J. Lin, et al., "Performance evaluation of ring-structure register file in multimedia applications," in *Proc. ICME*, 2003, pp.121-124

[12] *TMS320C55x DSP Programmer's Guide*, Texas Instruments Inc., July 2000

[13] T. Kumura, M. Ikekawa, M. Yoshida, and I. Kuroda, "VLIW DSP for mobile applications," *IEEE Signal Processing Mag.*, pp.10-21, July 2002

[14] R. K. Kolagotla, et al, "High-performance dual-MAC DSP architecture," *IEEE Signal Processing Mag.*, pp.42-53, July 2002

[15] M. Franklin and G. S. Sohi, "Register traffic analysis for streamlining inter-operation communication in fine-grain parallel processors," in *Proc. MICRO-25*, 1992, pp.236-245

[16] I. Park, M. D. Powell, and T. N. Vijaykumar, "Reducing register ports for higher speed and lower energy," in *Proc. MICRO-35*, 2002, pp.171-182

[17] V. Zyuban and P. Kogge, "The energy complexity of register files," in *Proc. ISLPED*, 1998, pp.305-310