Advanced Local-Conscious Global Register Allocator for VLIW DSP Processors with Distributed Register Files

K Kiran kumar [1], M Sathish Kumar[2],
1: student, UCEOU
3:Asst Professor,Vignan's Institute of Management and Technology for women,Hyderabad,India
2:Student,Princeton College of Engineering and Technology,Hyderabad,India

Abstract

The appearances of multi-banks of register files, distributed register clusters, and ping-pong architectures on embedded VLIW DSP processors such as PAC architectures present a great challenge for compilers to generate efficient codes for multimedia applications. In the literature, current research results in compiler optimizations for such problems have been limited to address issues for cluster-based architectures. It includes the work on partitioning register files to work with instruction scheduling [18], loop partitions for clustered register files [19], and cluster register files [16]. The work in [15] begins to address this complex optimization issue for embedded DSP processors, but only in the layer of copy propagation optimizations, and the work in [20] attempts to deal with software pipelining issues with distributed register files. In this paper, we address the issues dealing with global register allocations.

INTRODUCTION

Embedded processors developed in recent years have att-empted to employ novel hardware design to reduce ever-growing com-plexity, power dissipation, and die area. While using a distributed reg-ister file architecture with irregular accessing constraints is considered to be an effective approach rather than traditional unified register file structures, conventional compilation techniques are not adequate to uti-lize such new register file organizations for optimal performance. This paper presents a novel scheme for register allocation which composes of global and local register allocation, on a VLIW DSP processor withdistributed register files whose port access is highly restricted. In the scheme, a sub-phase prior to original global/local register allocation, named global/local RFA (register file assignment), is introduced to min-imize various register file communication costs. For featured register file structure where each cluster contains heterogeneous register files, con-ventional register allocation scheme with cluster assignment only have to be enhanced to cope both inter-cluster and intra-cluster communica-tions. Due to potential but heavy influences of global RFA on local RFA, a heuristic algorithm is proposed where global RFA manages to make suitable decisions on communication for local RFA. Experiments were done with a developing compiler based on the Open Research Compiler (ORC), and the results indicate that the compilation with the proposed approach delivers significant performance improvement, comparable to the solution using only the PALF scheme developed in our previous work

RFA, a heuristic algorithm is proposed where global RFA manages to make suitable decisions on communication for local RFA. Experiments were done with a developing compiler based on the Open Research Compiler (ORC), and the results indicate that the compilation with the proposed approach delivers sig-nificant

performance improvement, comparable to the solution using only thePALF scheme developed in our previous work.It has been suggested that a "picture is worth thousand words". This is all the more true in the modern era in which information has become one of the most valued of assets. A thousand words stored on a digital computer require very little capacity, but a single picture/image can require much more. The volume of data required to describe such images greatly slow transmission and makes storage prohibitively costly. The information contained in images must, therefore,be compressed by extracting only visible elements,which are then encoded . The quantity of data involvedis thus reduced substantially. Data compression algorithms are used in the standards such as 'JPEG' and'MPEG', to reduce the number of bits required forrepresenting an image or a video sequence, i.e.,compression is necessary and essential method forcreating image files withe manageable and transmittable sizes. A number of methods have been presented over the years to perform image compression. They all have one common goal: to alter the representation of information contained in an image so that it can be represented sufficiently well with less information.More recently, the wavelet transform has emerged as a cutting edge technology, within the field of image compression.For better performance ncompression, filters used in wavelet transforms should have the property of orthogonality, symmetry,short support and higher approximation order

.

Due toimplementation constraints scalar wavelets do not satisfy all these properties imultaneously.Multiwavelets can achieve better level of performance than scalar wavelets with scalar wavelets with similar computational complexity

.

Data compression is the technique to reduce the redundancies in data representation in order to decrease data storage requirements and hence communication costs. The objective of image compression is to reduceredundancy of the image data in order to be able to store or transmitdata in an efficient form , it is also used.Lossless compression methods may also be preferred for high value content, such asmedical imagery or image scans made for archival purposes.A com pression artifact is the result of an aggressive data compression scheme applied to an image, that discards some data that may be too complex to store in the available data-rate, or may have been incorrectly determined by an algorithm to be of little subjective importance, but is in fact objectionable to the viewer.with similar computational complexity. This paper is organized as follows. Section 2 highlights some key points on multiwavelets. Section3 provides the otivation for going into ultiwavelets for image compression. Section 4 presents the iteration of ecomposition in ultiwavelets. Section 5 discusses the coding of multiwavelet coefficients using modified SPECK.

DATA COMPRESSION

Data compression is the technique to reduce the redundancies in data representation in order to decrease data storage requirements and hence communication costs. Reducing the storage requirement is equivalent to increasing the capacity of the storage medium and hence communication bandwidth. Thus the development of efficient compression techniques will continue to be a design challenge for future communication systems and advanced multimedia applications. Data is represented as a combination of information and redundancy. Information is the portion of data that must be preserved permanently in ts original form in order to correctly interpret the meaning or purpose of the data. Redundancy is that portion of data that can be removed when it is not needed or can be reinserted to interpret the data when
needed. Most often, the redundancy is reinserted in order to generate the original data in its original form. A technique to reduce the redundancy of data is defined as Data compression. The redundancy in data representation is reduced such a way that it can be subsequently reinserted to recover the original data, which is called decompression of the data

IMAGE COMPRESSION

Image compression is the application of Data compression on digital images. The objective of image compression is to reduce redundancy of the image data in order to be able to store or transmit data in an efficient form. Image compression can be lossy or lossless. Lossless compression is sometimes preferred for artificial images such as technical drawings, icons or comics. This is because lossy compression methods, especially when used at low bit rates, introduce compression artifacts. Lossless compression methods may also be preferred for high 3 value content, such as medical imagery or image scans made for archival purposes. Lossy methods are especially suitable for natural images such as photos in applications where minor loss of fidelity is acceptable to achieve a substantial reduction in bit rate. The lossy compression that produces imperceptible differences can be called visuallylossless. Run length encoding and entropy encoding are the methods for lossless image compression. Transform coding, where a Fourier related transform such as DCT or the wavelet transform are applied, followed by quantization and entropy coding can be cit ed as a method for lossy image compression

BLOCK DIAGRAM OF SPECK IMAGE CODEC

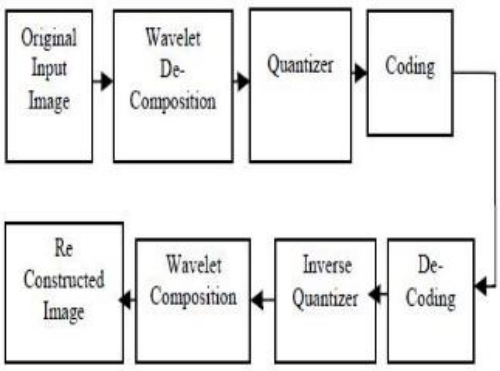

Figure4.1. Block diagram of SPECK image CODEC.

Fig. 1 shows typical image CODEC system. It comprises of three main stages: Transformation (Discrete Wavelet Decomposition), Quantization and Coding by rounding to the nearest integer.

A. Transformation
The principle of discrete wavelet image coding is based on the decomposition of an image into a number of frequency bands referred to as sub-bands.

B. Quantization
Each sub bands are quantized using uniform scalar quantizer, which is used to control the total bit rate

C. Coding The quantizer stage will be followed, by the scanning process of significant sets formation of binary bit stream. To reconstruct the image, the decoder basically performs the three main inverse operations in reverse order.

INTRODUCTION TO WAVELET TRANSFORM

The wavelet transform (WT) has gained widespread acceptance in signal processing and image compression,ωBecause of their inherent multi-resolution nature, waveletcoding schemes are especially suitable for applications where scalabilityand tolerable degradationare important, ω Wavelet transform decomposes a signal into a set of basis functions.ωThese basis functions are called Wavelets ω Wavelets are obtained from a single prototype wavelet ψ(t) called mother avelet by dilations and shifting:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}}\psi\left(\frac{t-b}{a}\right)$$

Where a is the scaling parameter and b is the shifting parameteris designed to be shared by the pair of M- and I-units in each cluster. The in-ternal of the D register file is further partitioned into two banks to utilize the instructional port switching technology in order to reduce more wire connec-tions between the M- and I-units. This technology, being referred to the name

as 'ping-pong register file structure', is that decreasing the register bank port connection which limits the accessibility of the two bank; in each cycle, the two functional units can only access to different banks. Each instruction bundle en- codes the information of which bank is to be accessed for each functional unit in the cycle so that the hardware can do port switching between D-register file banks and functional units, to attain the purpose of data sharing within a clus-ter. By using the concept of overlapping two different data-stream operations in a cluster, we may minimize the occasion that M- and I-units access the same data at the same time; therefore, the access constraints of 'ping-pong register file structure' should cause little impact on performance. The advantage of such a 'ping-pong register file structure' design is believed to consume less power due to its reduced read/write ports [11] while retaining the data communication ca-pability. Besides local register files and global register files, each cluster contains an additional constant register file which is shared by both M- and I-units as one of the read-only operand sources usable by certain instructions. Only M-units
can initialize the data in the constant register file

Register File Assignment Phases

The most variance between the register allocation for conventional unified regis-ter file architectures and the the irregular distributed register file structures used in the PAC DSP is the variety of the register files which may be allocatable for each variable operated by an instruction. With the irregular designs and various port access constraints in PAC DSP, the proper register file to be allocated can not be easily determined to produce the optimized results since the allocation heavily interferes with the instruction scheduling. Therefore, we prefer to sep-arate the register file assignment (RFA) from the register allocation, to simply the analysis and design of the proposed schemes, as well as the implementation to fit appropriately for the phase-ordering of our compiler infrastructures. The overall register allocation scheme currently involved in our developing compilers for PAC DSP is shown

This section reviews our previously proposed register allocation algorithm which, given a dependency DAG (Directed Acyclic Graph) [1] that describes the compi-lation regions, heuristically determines the proper register file/bank assignment and employ state-of-the-art graph-coloring register allocation for each assigned register file/bank in PAC architectures.

The overall register allocation algorithm proposed is shown in Fig. 3. Our approach requires building an extended data dependence DAG, which preserves the information of the xecution and storage relationship for irregular constraint analysis, in addition to the original partial order imposed by instruction prece-dence constraints. Nodes in the extended data dependence DAG represent in- structions of the input code block, with the component-type association (that indicates which functional unit is preferred to be scheduled for this node) and the register-type association (that annotates the appreciated physical register file/bank, to where the operands/results will be allocated); the edges linked be-tween the nodes represent data dependency that serializes the execution order to be followed in the scheduled code sequence. The main PALF register allocation scheme could be organized into five phases as follows

Conclusion

This paper proposed a new scheme involving separated phases named register file
assignment from the typical register allocation, which includes two sub-phases, LRA-RFA and

GRA-RFA. Due to the irregular distributed register file struc-tures on PAC DSP architectures, where the conventional register allocation is not appropriate, register file assignment could become the core procedure of the scheme to boost performance on the architecture. Preliminary experimental re-sults showed that the proposed scheme can utilize the distributed register file architectures and deliver great performance. Our future work will include the complete exploration of the various issues involved in our proposed scheme and the full version of the implementations.

References

1. Aho, A. V., J. D. Ullman, and R. Sethi: Compilers Principles, Techniques, and Tools. Addison-Wesley, Reading, MA, 1986.
2. CEVA: CEVA-X1620 Datasheet. CEVA, 2004.
3. David Chang and Max Baron: Taiwan's Roadmap to Leadership in Design. Microprocessor Report, In-Stat/MDR, Dec. 2004. http://www.mdronline.com/mpr/archive/mpr2004.html
4. A. Capitanio, N. Dutt, and A. Nicolau: Partitioned Register Files for VLIW's: A Preliminary Analysis of Tradeoffs. Proceedings of the 25th Annual International Symposium on Microarchitecture (MICRO-25), pages 292–300, Portland, OR, December 1–4 1992.
5. R. Ju, S. Chan, and C. Wu: Open Research Compiler for the Itanium Family. Tutorial at the 34th Annual Int'l Symposium on Microarchitecture, Dec. 2001.
6. T.-J. Lin, C.-C. Lee, C.-W. Liu, and C.-W. Jen: A Novel Register Organization for VLIW Digital Signal Processors. Proceedings of 2005 IEEE International Symposium on VLSI Design, Automation, and Test, pages 335–338, 2005.
7. T.-J. Lin, C.-C. Chang. C.-C. Lee, and C.-W. Jen: An Efficient VLIW DSP Architecture for Baseband Processing. Proceedings of the 21th International Conference on Computer Design, 2003.
8. Tay-Jyi Lin, Chie-Min Chao, Chia-Hsien Liu, Pi-Chen Hsiao, Shin-Kai Chen, Li-Chun Lin, Chih-Wei Liu, Chein-Wei Jen: Computer architecture: A unified processor architecture for RISC & VLIW DSP. Proceedings of the 15th ACM Great Lakes symposium on VLSI, April 2005.
9. Yung-Chia Lin, Chung-Lin Tang, Chung-Ju Wu, Ming-Yu Hung, Yi-Ping You, Ya-Chiao Moo, Sheng-Yuan Chen, and Jenq Kuen Lee: Compiler Supports and Optimizations for PAC VLIW DSP Processors. Proceedings of the 18th International Workshop on Languages and Compilers for Parallel Computing, 2005.
10. R. A. Ravindran, R. M. Senger, E. D. Marsman, G. S. Dasika, M. R. Guthaus, S. A. Mahlke, and R. B. Brown: Increasing the number of effective registers in a low-power processor using a windowed register file. Proceedings of the 2003 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (CASES '03), 125–136, 2003.
11. S. Rixner, W. J. Dally, B. Khailany, P. Mattson, U. J. Kapasi, and J. D. Owens: Register organization for media processing. International Symposium on High Performance Computer Architecture (HPCA), pp.375-386, 2000.
12. Texas Instruments: TMS320C64x Technical Overview. Texas Instruments, Feb 2000.

.

.