Journal of Computer Science 3 (10): 803-809, 2007
ISSN 1549-3636

A Novel Reconfigurable Execution Core for Merged DSP Microcontroller

[1]A. K. Rath and [2]P.K.Meher
[1]Department of Computer Science and Engineering,
College of Engineering Bhubaneswar, Bhubaneswar, Orissa, India
[2]School of Computer Engineering Nanyang Technological University,
Nanyang Avenue, Singapore

Abstract: The study presents an execution core which can be reconfigured either for calculation of digital convolution or for computation of discrete orthogonal transform by appropriate local buffer initialization of processing cells. It is shown that the data flow pattern can be changed by a single bit control signal. The proposed core can be connected to port 1 of Intel 8051 to derive the necessary control signals for reconfiguration. The core can be used as a pluggable module with existing microcontroller when DSP algorithms are required to be implemented. Using such execution core the computational load of the processor can be significantly reduced as the math-intensive components of the DSP algorithm is relegated to the execution core. The use of such pipelined core will not only caters to the need of real-time performance, but also it will facilitate scalability, reusability and flexibility for wide varieties of DSP functionalities.

Key words: Digital Signal Processor, microcontroller, merged architecture, embedded system, core-based DSP

INTRODUCTION

Digital signal processors (DSPs) are special purpose devices, designed especially to handle computation intensive digital signal processing algorithms[1,2]. A DSP may consist of I/O, data memory, program and control memory, address generators, ALU, and multiply accumulate (MAC) unit/ barrel shifter. The MAC unit, address generator and barrel shifter are used in DSPs to realize faster implementation of digital convolution and filtering applications[3-6]. Many a low cost general-purpose processors called as microcontroller which are basically designed to execute control-oriented tasks efficiently are widely available now. These processors are used in control applications where the computational requirements are modest. A microcontroller is a single integrated circuit that contains all the elements of complete computer systems, which includes CPU, memory, input/output ports and other constituent components. DSPs and microcontrollers have several commonalities in their architecture and application domain. Many applications require a mixture of control oriented as well as DSP functionalities. An example of such a system is digital cellular phone, which must implement both supervisory tasks and voice-processing tasks. A DSP can be used as microcontroller and a microcontroller can also be used for executing DSP algorithms. But, using a DSP for simple microcontroller application is not a cost-effective choice and a microcontroller in general may not be able to provide the desired real-time math-intensive DSP functionalities[7]. In general, microcontrollers provide good performance in controller tasks and poor performance in DSP tasks. DSP processors have the opposite characteristics. Hence, combination of control and signal processing applications were typically implemented using two separate processors: a microcontroller and a DSP processor. In the recent years, high performance microcontrollers are available which support DSP functionalities by adding fast multipliers, MAC units, or adding separate DSP units or coprocessors. A number of microcontroller vendors have begun to offer DSP-enhanced versions of their microcontrollers as an alternative to the dual-processor solution. Using a single processor to implement both types of functionalities is attractive, because it can potentially simplify the design task, save total chip area, reduce total power

Corresponding Author: Amiya Kumar Rath, Department of Computer Science and Engineering, College of Engineering Bhubaneswar, Bhubaneswar, India

consumption and reduce overall system cost[8,9].Microcontroller vendors such as Hitachi, ARM (Advanced RISC Machines) and Lexra have taken a number of different approaches for adding DSP functionality to the existing microprocessor design, borrowing and adapting the architectural features common among DSP processors. The DSP units in these microcontrollers contain fast MAC components, barrel shifters, registers, on-chip memory and bit-parallel interfaces to accommodate fast execution of DSP algorithms. However, as the amount of workload increases a single CPU cannot provide the desired performance. So DSP processor comes in to picture to handle the added load. Embedded microcontrollers can be designed where an existing microcontroller is integrated with the added DSP capability. The loosely connected combination of microcontroller and DSPs was successful, since it performs wide variety of applications. A single merged architecture gives distinct advantages of better and efficient performance and processing power in both application and system development[10]. Many of these hybrid processors achieve signal processing performance that is comparable to that of low-cost or mid range DSP processors while allowing re-use of software written for the original microcontroller architecture. The fully merged architecture provides simplicity of the single instruction stream and, with various forms of parallelism. The merged hybrid architecture i.e. integration of DSPs capability and microcontroller unit utilizes shared memory and data buses[7].

It has however potential threat of access conflict leading to detrimental effect in real-time supervisory and DSP functions. In this paper, we aim at examining the scope of merging of these two popular computing components in embedded devices for cost-effective, size sensitive, appropriately responsive to the environment in real-time and on-line applications using Reconfigurable execution core.

Core based system is given much importance in the recent years for embedded DSP system applications. Cores can be said to be complex building blocks to be used as functional entities in embedded system environment. With a rich cell library of predesigned, preverified circuit blocks, cores provide an attractive means to transfer technology to a system integrator and to develop new products by leveraging intellectual property advantages. Most importantly, the use of cores shortens the time to market for new system designs through design reuse. A core may be soft, firm, or hard. A soft core consists of a synthesizable HDL (Hardware Description Language) description that can be retargeted to different semiconductor processes. A firm core contains more structure, commonly a gate-level netlist that is ready for placement and routing. Often, core vendors design a firm core for a given process technology to get an estimate of the expected performance for that technology. A hard core includes layout and technology-dependent timing information, and is ready to be dropped into a system[11]. Examples of such cores include processor cores, memory cores, communication cores and bus-interface cores. The core-based implementation of system-on-chip (SOC) is gaining popularity in the recent years to minimize design cycle time in view of short-time-to market and so also for development of transient products under evolutionary technology. Other advantages of core-based approach are reusability and portability to other applications, facility for digital circuit abstraction for upgradation and correction[12]. The use of such cores is, therefore, rapidly increasing for design and implementation of embedded system-on-chips.

DESIGN ASPECTS OF THE PROPOSED EXECUTION CORE

Most of the DSP applications involve operations like filtering, encoding/decoding, interpolation, estimation of power spectral density and filter bank realization etc. which can be realized through calculation of finite digital convolution or discrete orthogonal transforms like discrete Fourier transform (DFT) and the discrete cosine transform (DCT)[13-16]. Calculations of convolutions and orthogonal transforms, however, are highly math-intensive and are required to be performed at a speed determined by the temporal constraint of the application for a real-time and on-line digital signal processing. As for example, in a discrete multi-tone modulation (DMT)-based digital subscriber line (DSL) transceiver, it is necessary to compute transforms of the order as high as 4096 at sampling rate up to 44.16 MHz. Similarly, in video encoder/decoders it is necessary to compute O (10^6) of 8-point transform samples per second. The image filtering operation also involves computation of similar magnitude for convolution operation. Never the less, there is a strong need of suitable reconfigurable processors for high-speed computation of the transform coefficients/ digital convolution to meet the requirements of real-time signal processing and digital multimedia communication systems[17-19]. Keeping the above behaviour of the DSP algorithms we envisage an ideal reconfigurable execution core to have the following features:

- The core should be dynamically reconfigurable during run time either for calculation of digital convolution or for computation of discrete orthogonal transform by appropriate control signals and local buffer initialization of processing cells.
- The core should switch from one configuration to another without temporal overhead such that switching from one configuration to other will be fast enough.
- The core should not demand substantial hardware for facilitating the reconfigurations.
- The hardware components of the reconfigurable system should be utilized optimally.
- The execution core should yield high throughput for real-time multimedia and image processing applications.

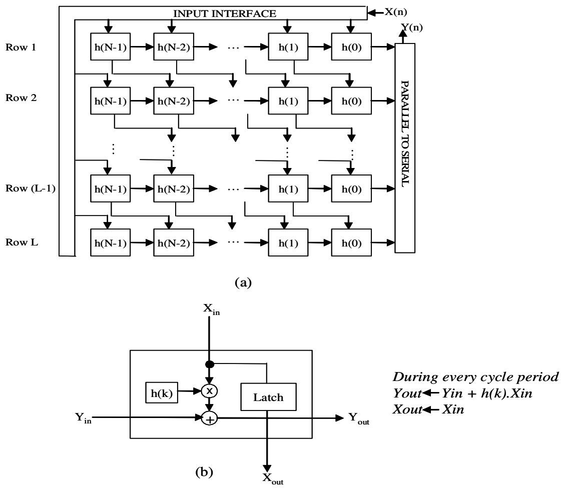
THE RECONFIGURABLE EXECUTION CORE

The finite digital convolution of a sequence {x(n)} with a convolving sequence {h(k)} is given by

$$y(n) = \sum_{k=0}^{N-1} h(k)\ x(n-k) \qquad (1)$$

Where {h(k) |, for k=0,1,…,N-1} is a finite duration sequence of length N and {x(n)} is, in general, an infinite duration sequence of input samples.

The calculation of finite digital convolution given by equation (1) involves basically N^2 number of multiply-accumulate or MAC operations, which can be implemented by a single MAC circuit for low-speed applications. But, for high-speed applications one may have to go for parallel implementation of (1) using a single array of N processing elements (PEs) which will compute N convolution output in N computational cycles where each computational cycle T = Tmult+Tadd. Tmult and Tadd are the time required to perform a multiplication and an addition, respectively[20].



During every cycle period
Yout← Yin + h(k).Xin
Xout← Xin

Fig. 1: Execution-Core Configuration for finite digital convolution (a) The execution core (b) Function of a PE

If the input sampling rate is faster enough and supposing that L samples are received by the structure in a single computational cycle it will be necessary to use L such arrays to increase the throughput rate by L times. Such an architecture consisting of L linear arrays for implementation of digital convolution given by (1) is shown in Fig. 1 (a). It consists of NL identical PEs,

Where L is number of input samples received at the input interface in each computational cycle. The PEs of the proposed structure are arranged in L rows and N columns. Function of the PEs is given in Fig. 1 (b). The proposed structure receives L input samples and yield L output samples during every cycle period. The first column of PEs of the proposed structure receives a block of L parallel samples and the last column of PEs yields a block of L convolved outputs in every cycle period.

The discrete orthogonal transforms[21] like DFT and DCT of a sequence {x(n)| for n = 0,1,…, N-1} is given by

$$y(n) = \sum_{k=0}^{N-1} C_{k,n} . x_k \qquad (2)$$

for n = 0, 1, …, N-1

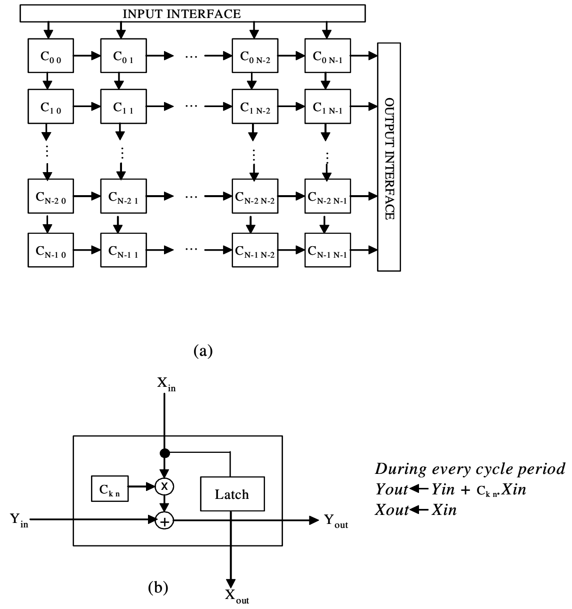


Fig. 2: Execution-Core Configuration for discrete orthogonal transform implementation (a) The execution core (b) Function of a PE

$C_{k,n}$ for k, n = 0, 1, …, N-1 form the transform kernel matrix of size (N × N) for the desired orthogonal transform. The transform output of (2) may be computed by the execution-core configuration depicted in Fig. 2. For calculation of length-N, transforms the structure consists of an N^2 PEs arranged in a square array of size (N × N), where each PE performs on MAC operation in every computational cycle T. During every cycle the structure accepts N input samples and delivers a throughput at the same rate once the pipeline is filled in. From Figs. 1 and 2 it is easy to see close similarities between the execution cores for the convolution and the orthogonal transform. The function of the PEs is identical in both the cases. The PEs are also arranged in a regular two dimensional array in both the cases. The structures, however, differ in terms of the data-flow pattern and number of PEs to be used, which the reconfiguration scheme has to take care of. Each PE has a local buffer to store the elements of transform kernel $C_{k,n}$ for computing the transforms or the coefficients {h(n)} for convolution operation. At the time of reconfiguration the local buffers of all those PEs involved in the computation are to be initialized by writing appropriate values, while the local buffer of all those PEs which do not participate during a particular run-time configuration are reset. The local buffer initialization is shown in Fig 3.

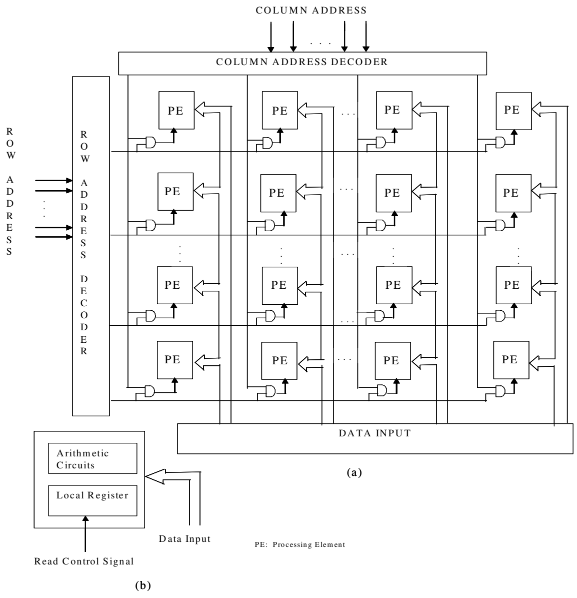


Fig. 3: Initialization of the local buffer of the PEs (a) The Initialization scheme (b) Structure of a PE

As shown in Fig 3, the execution core is provided with a row address decoder and a column address decoder for selecting the participating PEs one after another and appropriate coefficients from a data input buffer is written in to the local registers of the PEs. The scheme for facilitating necessary change in the data flow pattern for each configuration is shown in Fig 4. By a single bit control 'C' the structure can change its data-flow pattern from that for convolution to the pattern for the transform. For C = 1, it makes data-flow for convolution while for C = 0 it transfers data for transform computation[22].

MERGED ARCHITECTURE USING RECONFIGURABLE CORE

We have shown here that the run-time execution core presented in previous section for calculation of DOT and convolution can be used to realize a merged DSP-microcontroller architecture. It uses 8051 microcontroller. 8051 is an 8-bit microcontroller developed by Intel in 1981. It has 128 bytes of RAM, 4K bytes of on-chip ROM, two timers, one serial port, and four ports (each 8-bit wide) all on a single chip. The CPU can work only 8-bit of data at a time. Data larger than 8-bits has to be broken in to 8-bit pieces to be processed by the CPU. The 8051 has a total of four I/O ports, each 8-bits wide.

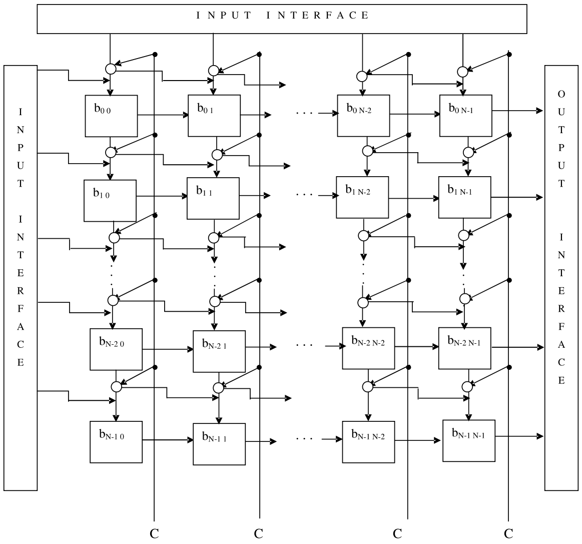


Fig. 4: (a) Reconfiguration Scheme for Execution

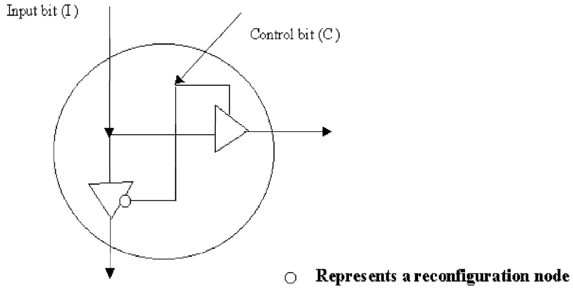


Fig. 4: (b) Structure of Reconfiguration Node for single-bit redirection

The proposed execution core (discussed in the last Section) assumes two configurations-one for calculation of the DOT and the other for convolution operation. For switching over from one configuration to the other it requires a single bit control signal. The core can be merged with a microcontroller to realize both DOT and convolution operations. One may consider using the core with Intel 8051. The necessary control signal for reconfiguration can be derived from one of the pins of port 1 of Intel 8051 as shown in Fig 5. If the pin is SET to 1 it may assume the configuration of convolution and if it is RESET to 0 it may correspond to DOT configuration. Loading the values of coefficient from the external buffer, one can initialize the local buffers of execution core. The control signal along with an address generator can be used for coefficient initialization.

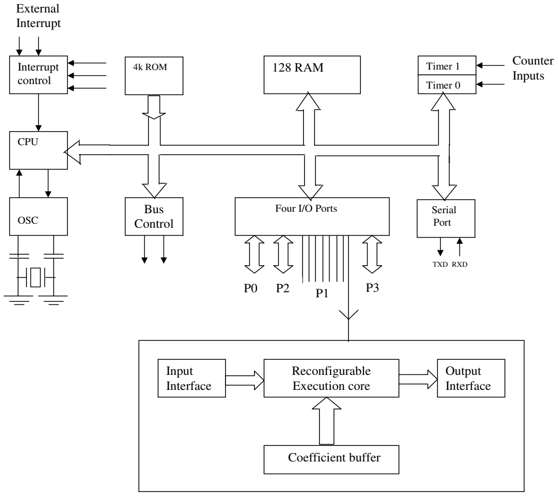


Fig.5: Proposed merged architecture

Merged architecture to realize digital signal processing and microcontroller functionalities have gained considerable popularity in the past few years in the embedded system arena due to various commonalities in their structure and common presence in several domain of applications. In this paper we have presented a merged DSP microcontroller architecture, where math-intensive functions of algorithms are relegated to a DSP component comprised of a transform modules, a multiplier array storage modules and a data interface unit. The DSP components can be integrated with microcontroller components to form a system-on-a-chip. The use of such transform modules will facilitate scalability, reusability and flexibilities for wide varieties of DSP functionalities. Desired speed performance can be achieved by exploiting the parallelism inherent with the computation of orthogonal transforms in pipelined arrays so as to cater to the need of real-time performance. Additional data storage and dedicated buses for DSP functionalities have been suggested to avoid possible conflict in resource sharing. The proposed architecture makes only incremental modification to the instruction set of conventional microcontroller. Therefore, the DSP hardware of the proposed structure may also be used as pluggable core to be used with a microcontroller when DSP algorithms are required to be implemented. The proposed merged architecture will be simple to design so as to take care of short-time-to market of the evolving embedded products. Apart from that using FPGA based transform modules it can be programmable for flexible custom solutions to domain specific applications[23-26].

CONCLUSION

A run-time reconfigurable execution core for calculation of discrete orthogonal transform and linear convolution is presented. Using the proposed execution core math-intensive DSP functions can be implemented by a simple microcontroller using a single-bit control signal through an I/O port. The execution core can be used with a microcontroller like Intel 8051 for several applications which involve supervisory function of microcontroller as well as DSP functions. The proposed core can be used as a pluggable module with existing microcontroller when DSP algorithms are required to be implemented. Using such execution core the computational load of the processor can be significantly reduced as the math-intensive components of the DSP algorithm is relegated to the execution core. The use of such pipelined core will not only caters to the need of real-time performance, but also it will facilitate scalability, reusability and flexibility for wide varities of DSP functionalities.

REFERENCES

1. Fettweis, G., 1997. DSP Cores for Mobile Communications: Where are we going? Proc. of ICASSP, pp: 279-282.
2. Verbauwhede, I. *et al.*, 1996 A low-power DSP engine for wireless communications. VLSI Signal Processing IX, IEEE, Eds. W. Burleson *et al.*, pp: 469-478.
3. Lee, E.A., 1988-89 Programmable DSP architectures: Part I & II. IEEE ASSP Magazine.
4. Lapsley, P., J. Bier, A. Shoham and E.A. Lee, 1996. DSP processor Fundamentals: Architectures and Features. IEEE Press.
5. Garreau, O. and R.E. Owen, 1998. Merged architecture approach embeds digital signal processing and improves real-time performance of microcontrollers. Proc. Paper #407 Embedded Systems Conf.
6. Walsh, D., 1996. Piccolo - The ARM architecture for signal processing: An innovative architecture for unified DSP and microcontroller processing. Proc. Intl. Conf. Signal Process. Applications and Technology (ICSPA96), 1 : 658-663.
7. Martin, D. and R. Owen, 1998. A RISC architecture with uncompromised digital signal processing and microcontroller operation. IEEE Intl. Conf. Acoustic Speech and Signal Processing (ICASSP98), pp: 3097-3100, Seattle, WA.
8. Rath, A.K., 2004. Core-based design of embedded DSP system. Ph.D. Thesis. Utkal University, Bhubaneswar.
9. Karthikeyan, M. *et a*L., 2000. A framework for cost vs. performance tradeoffs in the design of digital signal processor cores. 13th Intl. Conf. VLSI Design.
10. John, G.P. and D.G. Manolakis. Digital Signal Processing: Principles, Algorithms, and Applications. 3rd Edn.
11. Clark, G.A, S.R. Parker and S.K. Mitra, 1983. A unified approach to time- and frequency- domain realization of FIR adaptive digital filters: IEEE Trans. On Acoustics, Speech, & Signal Processing, ASSP-31 : 1073-1083.
12. Meher, P.K. and G. Panda, 1995. Fast computation of circular convolution of real-valued data using prime factor fast hartley transform algorithm. J. IETE, 41 : 261-264.
13. Nayak, S.S. and P.K. Meher, 1999. High-throughput vlsi implementation of discrete orthogonal transform using bit-level vector-matrix multiplier. IEEE Trans., Circuits and Systems-II: Analog and Digital Signal Processing, 46: 655-658.
14. Maharana, G. and P.K. Meher, 2000. Parallel algorithms and systolic architectures for 1- and 2-D interpolation using discrete transform. Intl. J. Computers & Applications, 22 : 1-7.
15. Agarwal, R.C. and J.W. Cooley, 1997. New algorithms for digital convolution. IEEE Trans. on Acoustics, Speech, and Signal Processing, 25 : 392-410.
16. Oppenheim, A.V. and R.W. Schafer, 1975. Digital Signal Processing. Englewood Cliffs, New Jersey, Prentice-Hall.
17. Cooley, J.W. and J.W. Tukey, 1965. An algorithm for the machine calculation of complex Fourier series. Math. Comp., 19 : 297-301.
18. Cowan, C.F.N. and P.M. Grant (Eds.), 1985. Adaptive Filters: Englewood Cliffs, New Jersey, Prentice-Hall.
19. Widrow, B. and S.D. Stearn, 1985. Adaptive Signal Processing. Englewood Cliffs, New Jersey, Prentice-Hall.
20. Clark, G.A., S.K. Mitra and S.R. Parker, 1981. Block implementation of adaptive digital filters. IEEE Trans. On Circuits and Syst., CAS-28 : 584-592.
21. Clark, G.A., S.R. Parker and S.K. Mitra, 1983. A unified approach to time- and frequency-domain realization of FIR adaptive digital filters. IEEE Trans. on Acoustics, Speech, & Signal Processing, ASSP : 1073-1083.

22. Schafer, R.W. and L.R. Rabiner, 1973. A digital signal processing approach to interpolation. Proc. IEEE, 61: 692-702.
23. Rath, A.K. and P.K. Meher, 2001. Embedded DSP microcontroller using discrete orthogonal transform. Proc. XXXVI Ann. Convention, CSI 2001, Kolkata, pp: c-7-c-11.
24. Rath, A.K. and P.K. Meher, 2002. Merged DSP Microcontroller Using Reconfigurable Execution Core Presented in 4th Asian Control Conference SICEC, Singapore. pp. 1910-1915.
25. Meher, P.K. , T.Srikanthan and A.K. Rath, 2003. Design of Efficient Embedded Merged DSP Microcontroller Using Configurable Cores" Presented in IEEE Symposium on Consumer Electronics-2003, Sydney Australia.
26. Rath A.K. and P.K. Meher, 2002. Reconfigurable Execution Core for High Performance DSP Applications, IEEE Asia-Pacific Conference on Circuits and Systems Kartika Plaza Beach Hotel, Bali, Indonesia. Vol 2, pp. 509-514.