Architectures for Heterogeneous Multi-Tier Networks *

Joseph R. Cavallaro (cavallar@rice.edu)

Center for Multimedia Communication Rice University, Dept. of Electrical and Computer Engineering 6100 Main Street, MS 380, Houston, TX 77005 USA +1 (713) 348-4719, Fax: +1 (713) 348-6196

December 2001

Abstract. Next-generation wireless computing platforms will contain flexible communications capabilities. At Rice University, the Rice Everywhere Network (RENÉ) project is investigating a multi-standard, multi-tier integration of W-CDMA cellular systems, high speed wireless LANs, and home wireless networks. There are many challenges in mapping these advanced communication algorithms to real-time hardware computing platforms. In this paper, we present current work on the development of a reconfigurable baseband physical layer containing DSP processors and FPGA accelerators. Our goal is the design of a multi-tier network interface card (mNIC) which is capable of exploiting efficient, low-power reconfiguration.

Keywords: W-CDMA, Wireless LAN, Rapid Prototyping, DSP and FPGA Architectures

1. Introduction

With the ever increasing use of wireless connectivity in laptop computers, personal digital assistants, and mobile phones, there is a greater need to support multiple communication standards for voice and data [1]. At Rice, we are currently developing the RENÉ (Rice Everywhere Network), as a prototype system to explore network adaptability and hardware reconfigurability. We have recently described the goals of the RENÉ concept [2], with a focus on the architectural issues in the design of a multi-tier Network Interface Card (mNIC).

Our ongoing work has been focused on third and fourth generation cellular system advanced receiver structures for Wideband Code Division Multiple Access (W-CDMA) [3]. We are developing through support from Nokia and Texas Instruments an algorithm simulation testbed using TI DSP processors and Field Programmable Gate Arrays (FPGA). This testbed infrastructure effort has the flexibility to integrate advanced prototype algorithm implementations for High Speed and Home/Desk Area Wireless LANs (WLANs). Our goal at the baseband physical layer is to enable the development of a compact, low power reconfigurable network interface card (NIC) to seamlessly support multiple wireless communication standards as shown in Figure 1.

2. Rapid Prototyping, Design and System Partitioning Tools

Wireless communication systems are evolving from voice-only cellular systems into highly configurable mobile data terminals. Our goals are to develop adaptable embedded systems for emerging multi-standard converged wireless communication systems. In order to enable the development

* This work was supported in part by Nokia Corporation, Texas Instruments, Inc., the Texas Advanced Technology Program under grant 1999-003604-080, and by NSF under grant ANI-9979465.

Figure 1. Reconfigurable Baseband Architecture.

of an advanced multi-tier wireless physical layer prototype, simulation and algorithm mapping must proceed through several design stages. There are two important realizations from the design cycle with implications for system design. First, the design takes place in two distinct locations. Because a cellular phone or PDA must be small and lightweight, its prototype by design contains minimal hardware: a power-efficient DSP, and a small display. In contrast, the system used to design the prototype is usually a powerful workstation, with a mouse, keyboard, video display, and large amounts of storage. Second, the design was specified using several languages: an equation description language, a block diagram language, and code in the C language, running on the DSP in the prototype.

In the Rice testbed environment, we integrate the two locations (host and DSP/FPGA prototype) and three languages (Simulink, Matlab, and “C”) inherent in DSP design [4]. The Wrapper, a software tool developed by at Rice, enables the designer to easily incorporate equations or “C” programs into a block diagram. The Switcher, also developed at Rice, allows the designer to freely move blocks of the design between the design system and the prototype. In contrast, traditional design methodology divides the design process into zones containing one language, and supporting one location. Moving between zones, from a simulation to an implementation on prototype hardware, requires rewriting the design in a different language appropriate for the different location. In Figure 2, an example based on a W-CDMA receiver structure is used to show a possible partitioning between the hardware resources.

In order to support the flexibility required along with the variety of data rates and service conditions, heterogeneous systems composed of DSP processors and configurable co-processors (FPGA) are important candidate architectures. We are currently focusing on the flexible mapping of advanced communication algorithms onto these embedded systems. In particular, our embedded hardware/software partitioning strategies are at the design tool and algorithm level - particularly, Simulink [5] and Matlab [6] integration with Xilinx FPGA System Generator [7] via Real-Time Workshop [8]. Our existing testbed using the Lyr DSP Signal Master [9] provides Simulink hardware-in-the-loop control of both the TI DSP and the Xilinx FPGA and allows for rapid prototyping with the RF Micro Devices 2.4 GHz RF unit [10] used for laboratory experiments. Figure 3 is a diagram of the programmable transceiver hardware which is connected to a general purpose host computer for control and interfacing.

Figure 2. Rapid Prototyping and Host/Prototype Partitioning.

Figure 3. Configurable Radio Hardware Architecture.

3. Architectures for Channel Estimation

The application of signal processing algorithms and information and coding theory to the design of channel estimation and multiuser detection techniques for wireless multiuser communication systems promises to improve system capacity. During the last few years, we have successfully demonstrated the effectiveness of these algorithms in exploiting the full advantages of W-CDMA technology and their robustness to the characteristics of wireless channels [11, 12]. Ongoing work

is focused on the real-time implementation of these maximum likelihood channel estimation algorithms on state of the art programmable DSP chips and reconfigurable FPGA hardware. Once again we are exploring ultimate system capacity tradeoffs for various proposed W-CDMA implementations.

In the following sections, we describe methods that lead to efficient architectures for a given target transmission data rate for detection and decoding for W-CDMA and WLAN systems. There are tradeoffs to be made between area, time, and power complexity within the VLSI/DSP architecture and data rate and BER targets of the communication algorithm. A goal of our work is to develop new metrics that characterize the power and BER tradeoffs of these VLSI/DSP architectures.

4. Differencing Multistage Detector Implementations

For the uplink in a W-CDMA cellular system, the multistage detection algorithm has been proposed as an effective interference cancellation scheme for 3G and 4G systems. We have developed a real-time VLSI implementation of this detection algorithm where we have achieved both high performance in interference cancellation and computational efficiency [13]. Our key observation is that as the interference cancellation process converges, the difference of the detection vectors between two consecutive stages is mostly zero. Under the assumption of BPSK modulation, the differences between the bit estimates from consecutive stages are 0 and ± 2 . Bypassing the zero terms saves computations. Multiplication by ± 2 can be easily implemented in hardware as arithmetic shifts. However, the convergence of the algorithm is dependent on the number of users, the interference and the signal to noise ratio and hence, the detection has a variable execution time. By using just two stages of the differencing detector, we achieve predictable execution time with performance equivalent to at least eight stages of the regular multistage detector. A prototype detector, handling up to eight users with 12-bit fixed point precision, was fabricated using a 1.2 μm CMOS technology and can process 190 Kbps/user for 8 users. A more flexible FPGA implementation of the differencing multistage detector is also being built to demonstrate the computational savings and the real-time performance potential.

The BER for the differencing multistage detector is exactly the same as the conventional multistage detector through the simulations. This is because we do not change the framework of the iterative method, nor the convergence rate. The percentage of zeros, which in turn signifies the reduction in complexity, in the differencing vector is illustrated in Figure 4(a). In this figure, we see that the percentage of zeros in the differencing vector increases as the iterations progress, which shows that the iterations converge progressively. After the fourth stage, the number of zeros approaches 98% in a 15-user communication system. This result explicitly indicates that if we use the conventional multistage detector, almost 98% of the computation resource is unnecessary in the fourth stage. Figure 4(b) gives us a clear view of how many computations are possible to save in a real system. The dotted line represents the accumulated number of floating point operations (flops) needed after each stage in the conventional multistage detector. As we explained earlier, the number of computations remains constant for each stage, which makes the total flops increase linearly. On the contrary, the number of computations in the differencing multistage detector decreases as the iteration proceeds. Thus we can achieve a 6X speedup in an eight stage system according to Figure

Figure 4. (a) Percentage of zeros in the differencing vector. (b) Flops comparison between differencing and non-differencing method.

4(b). With more stages in the system to increase the BER, higher speedups are obtained relative to the conventional multistage detector. This will also lead to power savings with our differencing multistage detector encouraging its flexibility and use in varying SNR and MAI situations. The differencing scheme can be easily extended to QPSK modulation, proposed in 3G systems as the real and imaginary components can be processed separately. To generalize this to other modulation schemes, the matrix transpose can be replaced with a Hermitian transpose using a generalized slicer in place of the sign function [13]. Additional improvements to detector efficiency can be achieved through the use of advanced computer arithmetic, such as on-line adders and multipliers [14].

5. Minimum Output Energy Handset Detector Implementations

For the downlink, we have investigated detector architectures for 3G wireless handsets employing DS-CDMA. The code-matched filter (MF) and minimum output energy (MOE) detectors were analyzed with respect to fixed-point arithmetic behavior. Architectures employing fixed-point arithmetic are then developed for these detectors. The maximum throughput of these architectures and the associated costs in terms of area usage and power consumption are evaluated. Results of the fixed-point analysis indicate that the MOE detector is more susceptible to quantization than the MF detector. Results of implementation indicate that the superior performance of the MOE detector is achieved at a considerably higher cost in terms of area usage and power consumption. Finally, comparison of hardware implementation with software-based DSP implementation indicates that software approaches result in considerably lower throughputs. [15]

The MF and MOE detector architectures were implemented on a Xilinx Virtex XCV800 FPGA and a TI 'C6x DSP using a length 31 Gold spreading code. The Virtex power estimator was used

to estimate the power consumption of the FPGA for both architectures. Uniform word lengths for each detector were chosen based on the quantization analysis. Results of the implementation are reported in Table I. We make two key observations with respect to these results. Firstly, the MF detector benefits from its simplicity by requiring just 1.45% of the area of the MOE detector (fixed point word-length of 16). Secondly, it achieves a maximum throughput that is twice that of the MOE detector, while its power consumption is a quarter of the MOE detector. This indicates that the higher BER performance of the MOE detector is achieved at a significantly higher cost in terms of area usage and power consumption. We also compare the performance of the detectors given a software-based DSP (fixed data path) implementation versus a custom implementation on an FPGA. The ratio of the DSP throughput to the FPGA throughput is 1/2 for the MF detector and 1/11 for the MOE detector, indicating that superior data rates are achievable in an application-specific device.

Table I. Detector Implementation Results

Detector	Fixed-Point Precision (bits)	4 Input LUT	CLB Slices	Gate Count (Gates)	Chip Rate (Mcps)	Throughput (Mbps)	Power Estimate (mW)	DSP Throughput (Mbps)
MF	16	78	43	745	98.50	3.18	310	1.61
	10	48	27	481	96.94	3.13	269	-
	8	36	21	381	101.69	3.28	259	-
MOE	16	1416	764	51257	44.35	1.47	1221	0.135
	14	1120	613	47483	46.05	1.53	1069	-
	12	734	415	42677	44.04	1.46	788	-

6. Architecture Realizations for the Viterbi Decoder

The Viterbi Algorithm is computationally demanding not because its algorithm is complex in a conceptual sense. In fact, the essence of the algorithm is a relatively simple procedure of identical add, compare, select, and traceback operations. Rather, the computational burden arises because a relatively simple set of operations must be applied to a large number of basic nodes or states at each discrete time step. The number of states grows exponentially with constraint length. With the limitations of present fabrication technology there has been a great incentive to devise algorithms that assign more than one state per processor and/or constrain interprocessor communication such that the area necessary to wire processors does not dominate the area required by the processors themselves. Locally connected processor arrays are of interest here because they satisfy the latter constraint. In fact, the Viterbi Algorithm has benefited much from research in the use of processor arrays, [16, 17, 18], for popular algorithms like sorting, polynomial multiplication, matrix transposition and Fast Fourier Transform.

The most straightforward implementation of the Viterbi Algorithm is a completely sequential one where every state is evaluated, in sequence, in a single arithmetic logic unit, driven by a programmed control unit (i.e., a microprocessor). This approach, though processor poor, and speed wise slow, requires a very small area. The other extreme is a fully parallel implementation of the Viterbi algorithm where each state is assigned one processor and the interprocessor connection network is a shuffle exchange graph. In the context of a VLSI realization, this type of fully parallel layout, though dominated by large interprocessor wire area, is the architectural organization with the greatest possible throughput for a given fabrication technology. A more recent effort in this direction is a bit serial implementation of a Viterbi decoder for 3rd generation W-CDMA systems [19]. This is for a rate $1/3$ and $K = 9$ Viterbi decoder chip. Bit serial arithmetic for the ACS units is used for savings in area. Also, they use a floorplan of the ACS modules such that the 256 ACS units are clustered together in groups which reduce interconnection wiring. This bit serial implementation of the Viterbi decoder chip can operate at 2-20 Mbps. Thus, a fully parallel approach is suitable for high speed Viterbi decoding.

The various implementations of the Viterbi decoder discussed above are suitable for use in different systems depending upon the requirements of the system. For example, if the system is such that the decoding throughput is low priority and the area needs to be minimized, a uniprocessor architecture would be ideal. On the other hand, for high speed decoding, a parallel or cascade structure would be preferred. However, none of these architectures address the issue of reconfigurability of the decoder. Each of these is designed for a particular configuration of the decoder. We have tried to overcome this problem in our design of a reconfigurable decoder.

Another hardware implementation of a flexible Viterbi decoder which provides dynamic reconfigurability is that of a DSP coprocessor [20]. This decoder is geared towards a solution for 3G base station architectures i.e. a data rate of 2.5 Mbps and a constraint length of 9. It is mainly concerned with low power, small area, relatively low data rates, and a flexible system interface for co-ordination with the DSP. Under the constraints of small area and low power, the architecture used for the ACS units is the general cascade approach.

7. Real-Time Reconfigurable Viterbi Decoder Architecture

The key challenge in designing a reconfigurable decoder architecture is the realization of suitable structures for the three major blocks in the decoder i.e. the BMU, ACS, and the SMU. Since we were aiming for high decoding throughput, in the range of 2Mbps for W-CDMA to 54 Mbps for WLAN, we chose a fully parallel implementation of the decoder for all the constraint lengths we want to support. This means that we have hardware for the largest constraint length decoder and the smaller constraint length decoders use only part of the total resources available [21]. Also, this obviates the need for any state metric memory and address generation circuitry which makes the design simpler. However, now since we shall be using registers for holding the state metrics, the interconnection network between the outputs of the ACS units and the state metric registers becomes even more complex as it also needs to have the ability to configure itself into a different trellis structure for different constraint lengths.

It is interesting to compare the results of this parallel implementation with a serial implementation done on a TI TMS320C54x Digital Signal Processor [22]. The TMS320C54x incorporates a special hardware unit to accelerate the Viterbi metric update computation. This is essentially an add-compare-select-store unit with dual accumulators and a splittable ALU which is capable of performing a Viterbi butterfly in four cycles. We can see the clear dependence of the number of cycles per frame on the constraint length of the code. Table II and Table III compare this DSP implementation with our fully parallel one for the WLAN and W-CDMA standards, where L is the frame length, κ is the constraint length, and n denotes a rate $1/n$ coding rate.

However, the comparison is not completely fair because we have not taken power and cost into account. Also, a more recent TI C64x DSP has an on-chip Viterbi coprocessor which is exclusively meant to reduce the burden of Viterbi decoding from the DSP. It can decode up to 50 channels of 7.95 Kbps each giving an aggregate throughput of about 4 Mbps. It is apparent that an FPGA implementation is much closer at achieving the required data rates. For the W-CDMA case, the DSP can attain decoding rates required for indoor environments (384 Kbps), while the FPGA exceeds the required data rate requirement. For the WLAN, the FPGA is able to achieve the rates up to the 24 Mbit/s standard easily. Rates up to 54Mbps and even 100 Mbps can be achieved on the FPGA as technology improves to yield not only faster gate array devices but also software tools which efficiently map the hardware design onto the devices.

Table II. DSP Benchmarks: WLAN/W-CDMA.

Standard	R	K	L	Cycles/ L	F_{max} (MHz)	Rate(Mbps)	Desired
WLAN	1/2	7	192	40752	200	0.94	6-54
W-CDMA	1/2	9	192	129840	200	.29	2.5
W-CDMA	1/3	9	192	131376	200	.29	2.5

Table III. FPGA Benchmarks: WLAN/W-CDMA.

Standard	R	K	L	Cycles/ L	F_{max} (MHz)	Rate(Mbps)	Desired
WLAN	1/2	7	192	588	80	26.1	6-54
W-CDMA	1/2	9	192	592	80	26.1	2.5
W-CDMA	1/3	9	192	592	80	25.9	2.5

8. Maximum Weight Basis Decoding

We have recently developed a new suboptimal decoding technique for linear codes based on the calculation of maximum weight basis of the code. [23]. The idea is based on estimating the maximum number of locations in a codeword which have least probability of estimation error without violating the codeword structure. For example, the error correcting capability of the convolutional code increases with the constraint length of the code. Unfortunately the decoding complexity of the Viterbi algorithm grows exponentially with the constraint length. We also augment the maximal weight basis algorithm by incorporating the ideas of list decoding technique. The complexity of the algorithm grows only quadratically with the constraint length and the performance of the algorithm is comparable to the optimal Viterbi decoding method. The reduction in complexity is achieved without significantly affecting the performance of the system. It should be mentioned that our algorithm is equally applicable to any block-codes. The decoding idea is similar to the generalized Dijkstra's algorithm [24] used for block codes.

Our analysis of the computational complexity shows that it requires significantly fewer operations than the Viterbi algorithm. Since the algorithm is not necessarily an optimal algorithm we need to study the performance loss due to this suboptimal decoding technique. We first compare the performance of the suboptimal channel decoding algorithm with the Viterbi algorithm in an AWGN channel. Figure 5 shows the performance of a systematic convolutional code of rate 1/2 and constraint length 7. We decode 100 information bits at a time. For our simulation we have weight bases, $M = 6$, that is we only have kept a codeword list of size 6 in the suboptimal algorithm. The simulation results show there is very little performance loss using the suboptimal algorithm over the optimal decoding algorithm.

Figure 5. (a) Suboptimal decoding algorithm versus Viterbi decoding for rate 1/2 convolutional code with $\kappa = 7$. (b) Performance of Viterbi algorithm on codes of $\kappa = 11$ and MWB algorithm on codes with $\kappa = 18$.

But the performance gain is more pronounced for codes with larger constraint length. In fact even though the performance of our algorithm is suboptimal compared to the Viterbi algorithm, since the computational complexity of our algorithm is much lower, we can afford to decode a stronger code with larger constraint length within the same time in which the Viterbi algorithm can decode a codeword with smaller constraint length. We study such a situation in Figure 5. In this example we decode a convolutional code of rate $1/2$ and constraint length 11 which is decoded by an optimal Viterbi decoder and another codeword of same rate but constraint length 18 which is decoded by our algorithm and compare their performance. We use $M = 10$. For the Viterbi algorithm we need to perform $(2^{12}N)$ operations while our algorithm requires only $(18^2 + 100)N$ operations. Yet the performance is almost indistinguishable. This is because of the better error correcting capability of the code with larger constraint length. It should however be noted we haven't counted the exact number of operations but merely the order, and such a study of implementation aspects is planned for the future, but the potential of our system is apparent.

9. Summary and Future Directions

Higher data rates in wireless communication systems are typically achieved at the expense of increased power consumption in both the digital baseband processing and in the RF transmitter. In order to implement feasible battery-operated wireless handheld systems, a careful balance must be achieved between RF and baseband power consumption [25, 26, 27, 28]. Current research can be broadly divided in terms of schemes at the circuit level or at the signal processing algorithm level. At the circuit level, efficient power amplifier designs promise RF power reduction while efficient architectures employing dynamic frequency and voltage scaling will reduce digital baseband power consumption. Furthermore, at the signal processing algorithm level, RF power efficiency can be improved through schemes for the reduction of transmitter Peak to Average Power Ratio (PAPR). Similarly, digital baseband processor efficiency can be improved through dynamic modulator, detector, and decoder algorithm adaptation. At the system level, direct conversion receivers seek to save power by converting from RF carrier frequencies to baseband resulting in the elimination of the intermediate frequency stages. Also at the system level, software defined radios may provide reconfiguration capabilities to different standards and can be enhanced by selected co-processor modules. In this paper, we have overviewed some of our current research efforts at Rice University on high performance wireless communication systems. Our goal is a low-cost, low-power multi-standard cellular and WLAN terminal and the associated basestation infrastructure.

Acknowledgements

The Rice Everywhere Network and the multi-tier Network Interface Card is a multi-year research activity of the Rice Center for Multimedia Communication (<http://www.cmc.rice.edu>) and is a collaboration among Behnaam Aazhang, Richard Baraniuk, Joseph Cavallaro, Patrick Frantz, Edward Knightly, Ashutosh Sabharwal, and Dan Wallach in Electrical and Computer Engineering and Computer Science.

References

1. J. Kalliokulju, P. Meche, J. Rinner, J. Vallström, P. Varshney, and S.-G. Häggman, "Radio access selection for multistandard terminals," *IEEE Communications Magazine*, vol. 39, pp. 116–124, October 2001.
2. B. Aazhang and J. R. Cavallaro, "Multi-tier Wireless Communications," *Kluwer Wireless Personal Communications, Special Issue on Future Strategy for the New Millennium Wireless World*, vol. 17, pp. 323–330, June 2001.
3. T. Ojanpera and R. Prasad, eds., *Wideband CDMA for Third Generation Mobile Communications*. Artech House Publishers, 1998.
4. B. A. Jones, "Rapid Prototyping of Wireless Communications Systems," Master's thesis, Department of Electrical and Computer Engineering, Rice University, Houston, TX, January 2002.
5. The MathWorks, Inc., Natick, MA, *Using Simulink*, June 2001. Revised for Simulink 4.1 (Release 12.1).
6. The MathWorks, Inc., Natick, MA, *Using Matlab*, June 2001. Revised for MATLAB 6.1 (Release 12.1).
7. The MathWorks, Inc., U. S. A, *Xilinx System Generator v2.1 for Simulink: User's Guide and Blockset Reference Manual*, 2001.
8. The MathWorks, Inc., Natick, MA, *The Real-Time Workshop User's Guide*, June 2001. Revised for Simulink 4.1 (Release 12.1).
9. L. Belanger, J. Ahern, and P. Fortier, "Prototyping wireless base stations or edge devices on a dsp/fpga architecture using high-level tools," in *International Conference on Signal Processing, Applications, and Technology (ICSPAT)*, (Dallas, TX), October 2000.
10. "RF2938 2.4GHz Spread-Spectrum Radio." RF Micro Devices, Reference Design RD0310, 2000.
11. C. Sengupta, J. R. Cavallaro, and B. Aazhang, "On multipath channel estimation for CDMA using multiple sensors," *IEEE Transactions on Communications*, vol. 49, pp. 543–553, March 2001.
12. S. Rajagopal, S. Bhashyam, J. R. Cavallaro, and B. Aazhang, "Efficient VLSI architectures for baseband signal processing in wireless base-station receivers," in *12th IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, (Boston, MA), pp. 173–184, July 2000.
13. G. Xu, S. Rajagopal, J. R. Cavallaro, and B. Aazhang, "VLSI Implementation of the Multistage Detector for Next Generation Wideband CDMA Receivers," *to appear in Journal of VLSI Signal Processing*, 2002.
14. S. Rajagopal and J. R. Cavallaro, "On-line arithmetic for detection in digital communication receivers," in *15th IEEE International Symposium on Computer Arithmetic (ARITH-15)*, (Vail, CO), pp. 257–265, June 2001.
15. F. Livingston, V. Chandrasekhar, M. Vaya, and J. R. Cavallaro, "Handset Detector Architectures for 3G Wireless Systems," in *Proc. IEEE Intl. Symposium on Circuits and Systems (ISCAS)*, (Phoenix, AZ), 2002. Submitted.
16. P. Gulak and T. Kailath, "Locally Connected VLSI Architectures for the Viterbi Algorithm," *IEEE Journal on Selected Areas of Communications*, April 1988.
17. G. Fettweis and H. Meyr, "Parallel Viterbi Algorithm Implementation: Breaking the ACS Bottleneck," *IEEE Transactions on Communications*, vol. 37, August 1989.
18. G. Fettweis and H. Meyr, "High Rate Viterbi Decoders: A Systolic Array Solution," *IEEE Journal on Selected Areas in Communications*, vol. 8, pp. 1520–1534, October 1990.
19. H. Suzuki, Y.-N. Chang, and K. K. Parhi, "Low-power bit-serial Viterbi decoder for 3rd generation W-CDMA systems," in *Proc. IEEE Custom Integrated Circuits Conference*, (San Diego, CA), pp. 589 – 592, May 1999.
20. D. Hocevar and A. Gatherer, "Architecture Selection for a Low Power Flexible Viterbi Decoder," in *IEEE Int. Conf. on Third Generation Wireless Communications*, (San Francisco, CA), June 2000.
21. K. Chadha and J. R. Cavallaro, "A Dynamically Reconfigurable Viterbi Decoder Architecture," in *Proc. Asilomar Conference on Signals, Systems and Computers*, (Pacific Grove, CA), November 2001.
22. H. Hendrix, "Viterbi Decoding Techniques in the TMS320C54x Family," June 1996.
23. S. Das, E. Erkip, J. R. Cavallaro, and B. Aazhang, "Maximum Weight Basis Decoding of Convolutional Codes," in *Proc. IEEE Global Telecommunications Conference (Globecom)*, vol. 2, (San Francisco, CA), pp. 835–841, November 2000.
24. T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. MIT Press, 1995.

25. B. A. Myers, J. B. Willingham, P. Landy, M. A. Webster, P. Frogge, and M. Fischer, "Design considerations for minimal-power wireless spread spectrum circuits and systems," *Proc. IEEE*, vol. 88, pp. 1598–1612, October 2000.
26. C. Luschi, M. Sandell, P. Strauch, J.-J. Wu, C. Ilas, P.-W. Ong, R. Baerwyl, F. Battaglia, S. Karageorgis, and R.-H. Yan, "Advanced signal processing algorithms for energy-efficient wireless communications," *Proc. IEEE*, vol. 88, pp. 1633–1650, October 2000.
27. J. M. Rabaey, M. Potkonjak, F. Koushanfar, S.-F. Li, and T. Tuan, "Challenges and opportunities in broadband and wireless communication designs," in *Proc. IEEE/ACM International Conference on Computer Aided Design*, (San Jose, CA), pp. 76–82, November 2000.
28. C. Teuscher, D. Lee, N. Zhang, and R. Brodersen, "Design of a wideband spread spectrum radio using adaptive multiuser detection," in *IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 593–599, May 1998.

Author's Vitae

Joseph R. Cavallaro

Joseph R. Cavallaro was born in Philadelphia, Pennsylvania on August 22, 1959. He received the B.S. degree from the University of Pennsylvania, Philadelphia, PA, in 1981, the M.S. degree from Princeton University, Princeton, NJ, in 1982, and the Ph.D. degree from Cornell University, Ithaca, NY, in 1988, all in electrical engineering.

From 1981 to 1983, he was with AT&T Bell Laboratories, Holmdel, NJ. In 1988, he joined the faculty of Rice University, Houston, TX, where he is currently an Associate Professor of Electrical and Computer Engineering, having received tenure in 1994. His research interests include computer arithmetic, fault tolerance, VLSI design and microlithography, and DSP and VLSI architectures and algorithms for applications in wireless communications and robotics.

Dr. Cavallaro is a recipient of the NSF Research Initiation Award 1989-1992 and the IBM Graduate Fellowship 1987-1988, and is a member of IEEE, Tau Beta Pi and Eta Kappa Nu. During the 1996-1997 academic year, he served at the U.S. National Science Foundation as director of the Prototyping Tools and Methodology Program in the Computer (CISE) Directorate. He is currently the Associate Director of the Center for Multimedia Communication at Rice University.