

# MIPS Cost Estimation for OFDM-VBLAST systems

Haiyan Jiao, Anders Nilsson, Eric Tell, and Dake Liu  
Div. of Computer Engineering at Department of Electrical Engineering,  
Linköping University, Linköping, Sweden  
E-mail: haiji374@student.liu.se, {andni,erite,dake}@isy.liu.se

**Abstract**—The focus of this paper is to investigate the feasibility of using programmable DSP processors for MIMO based radio systems. Several detection algorithms were evaluated and MIPS costs for low complexity detection and channel estimation algorithms for OFDM-VBLAST MIMO systems were calculated. Based on the MIPS cost estimation a feasible hardware architecture was derived. The result shows the feasibility of implementing MIMO radio systems in a programmable architecture.

## I. INTRODUCTION

Multiple-Input-Multiple-Output (MIMO) technologies for radio communications give significant improvements on capacity and channel efficiency. Its implementation onto ASICs (Application Specific Integrated Circuits) has been done for some systems. An ASIC implementation gives high performance for computationally demanding algorithms specific applications. However, it does not give enough flexibility to support different requirements. Considering the various systems using MIMO technology and the convergence of multiple radio standards in a single product, it may be necessary to cover a relatively wide range of algorithms used in different standards and different systems.

According to benefits obtained, MIMO techniques can basically be split into three groups: space-time coding (STC) to maximize spatial diversity, space division multiplexing (SDM) to increase the transmit data rate, and close-loop systems. Space-time codes, (such as space-time trellis codes (STTCs), and space time block codes (STBCs)), are schemes to encode and transmit the streams jointly in several coded symbols simultaneously from each antenna, to protect transmission against errors caused by channel fading, noise and interference; while space division multiplexing (SDM) systems transmit streams of independent data over different antennas to increase the data rate.

VBLAST (Vertical Bell Labs Layered Space Time) system is a commonly used system based on the Diagonal BLAST system [1] for SDM. Here, we give an example of a VBLAST system implementation on programmable baseband DSP processors. The implementation is based on the IEEE 802.11a specification, but uses four transmit antennas and four receive antennas to reach four times of performance of IEEE 802.11a in frequency-selective fading channel conditions. We will analyze and select algorithms for detection and channel estimation of an OFDM-VBLAST system and estimate their

MIPS costs. Since the implementation uses a programmable solution, the same hardware can also be used for other radio baseband applications.

The paper is organized as follows. Section II describes the basic OFDM-VBLAST system model, with the detection and channel estimation algorithms investigated. In section III, a multiple processor architecture is presented, and the MIPS cost is estimated. Finally, conclusions are drawn in section IV.

## II. SYSTEM MODELING

### A. Scope of the MIMO transceiver

The OFDM system will be implemented as an example in this paper. The implementation of a MIMO-OFDM system with  $N_t$  transmit antennas and  $N_r$  receive antennas can be carried out as shown in Figure 1. [4]. In this paper, we only consider symbol operations, hence the system starts after the mapper in the transmitter and ends with the de-mapper in the receiver.

The processing in the transmitter includes:

- 1) Division of data into several sub-streams.
- 2) Channel coding and interleaving.
- 3) Cyclic prefix (CP) appending.

The receiver includes several processes as follows:

- 1) Time and frequency synchronization.
- 2) FFT after removing the cyclic prefixes.
- 3) Channel estimation using VBLAST detection for each OFDM sub-carrier.
- 4) Detection of each OFDM sub-carrier

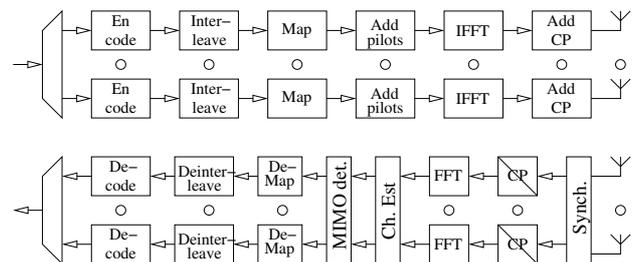


Fig. 1. OFDM-VBLAST system

### B. System modeling

The purpose of system modeling is to find the MIPS costs for channel estimation and data reception. The received signal

at the  $j$ -th receive antenna at the  $k$ -th tone of the  $n$ -th block can be expressed as:

$$r_j[n, k] = \sum_{i=1}^{N_t} H_{ij}[n, k] s_i[n, k] + w_j[n, k]$$

for  $j = 1, \dots, N_r$ ,  $k = 0, \dots, N_k - 1$ , where  $s_i[n, k]$  is the symbol transmitted from the  $i$ -th transmit antenna at the  $k$ -th tone of the  $n$ -th block, and  $w_j[n, k]$  is independent and identically distributed complex zero-mean Gaussian noise with variance  $\sigma$ .

### C. Data frame structure

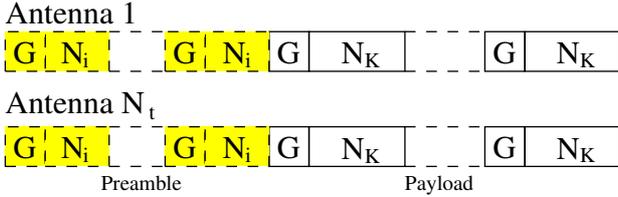


Fig. 2. Data frame structure

The data frame structure for each antenna is illustrated in figure 2, where the preamble portion has  $N_t$  training symbols of length  $N_I$ , and the user data portion has  $N$  data symbols of length  $N_k$ , [4]. All the symbols have a Cyclic Prefix (CP) of length  $G$  to avoid inter-symbol interference.  $G > K_0$ , which is the length of the discrete-time channel impulse response. At the receiver end, after removing the CPs and doing an  $N_k$ -point FFT, all the symbols become  $N_k$  long Frequency domain sequences. The preamble sequences are used for channel estimation, while the data sequences are used for detection.

### D. Channel estimation algorithm

According to the LS criterion, the temporal channel estimation  $\hat{h}_{ij}[n, l]$  can be calculated by  $\hat{h}[n] = Q^{-1}[n]P[n]$ , where:

$$Q_{ij}[n] = (q_{ij}[n, l_1 - l_2])_{l_1, l_2=0}^{K_0-1}$$

$$q_{ij}[n, l] = \sum_{k=0}^{N_k-1} s_i[n, k] s_j^*[n, k] W_{N_k}^{-kl}$$

$$P_i[n] = (p_i[n, 0], \dots, p_i[n, K_0 - 1])^T$$

$$p_i[n, l] = \sum_{k=0}^{N_k-1} r[n, k] s_i^*[n, k] W_{N_k}^{-kl}$$

$$\hat{h}[n] = (h_1[n] \dots h_{N_t}[n])^T$$

$$\hat{h}_i[n] = (h_i[n, 0] \dots h_i[n, K_0 - 1])^T$$

And the channel frequency response can be obtained by doing FFT:

$$H_{ij}[n, k] = \sum_{l=0}^{K_0-1} h_{ij}[n, k] W_{N_k}^{kl}$$

### E. Optimum Training Sequences

The  $2K_0 \cdot 2K_0$  matrix inversion for computing  $h[n]$  is computationally heavy. To simplify the equation, we can simplify the matrix  $Q$ . Assume that the modulation results is a constant-modulus signal, so that  $Q_{ij}[n] = N_k I$ , where  $I$  is a  $K_0 \cdot K_0$  identity matrix. If the matrix  $Q$  is simplified as  $Q_{ij}[n, k] = 0$  for  $i \neq j$ , then  $\hat{h}_i[n] = \frac{1}{N_k} P_i[n]$ . This means that the training sequences from different transmit antennas are desired to be orthogonal and shift-orthogonal. Then, we can get the simplified algorithm for a quasi-static fading channel, with some acceptable performance degradation [2].

$$\hat{h}_i[n] = \begin{cases} \frac{1}{N_k} P_i[n] & n = 1 \\ \frac{1}{N_k} (P_i[n] - \sum_{j \neq i}^{N_t} Q_{ij}[n] \hat{h}_j[n-1]) & n \geq 2 \end{cases}$$

With the same definition for  $Q_{ij}$ ,  $q_{ij}$ ,  $P_i$  and  $p_i$  as in the original algorithm.

### F. VBLAST Detection

Several common methods for VBLAST system detection are listed below:

1) *Matched Filter (MF)*: MF multiplies the received signal with the conjugate transpose of the corresponding channel coefficient, and compares the result with all possible symbols. It needs the channel coefficients to be orthogonal to avoid interference from different paths.

2) *Maximum Likelihood (ML)*: ML compares the received symbol with all possible combinations of symbols after transmission resulting in a very high complexity for searching.

3) *Zero Forcing (ZF)*: ZF multiplies the inverted channel coefficient matrix with the received signal.

4) *Minimum Mean Square Error (MMSE)*: Based on ZF, MMSE considers the Signal Noise Ratio (SNR) when multiplying the inverse matrix.  $\hat{s} = (H^H H)^{-1} H^H r$ .

5) *Successive Interference Cancellation (SIC)*: SIC includes 3 processes: ordering, nulling, and cancellation [3]. First, all the substreams are ordered according to signal strength, starting with the strongest signal. Second, the receiver estimates the strongest signal, nulling out the remaining weaker signals, according to some performance criterion, like ZF or MMSE. Finally, the interference of the detected signal is subtracted from the signal vector, to reduce the complexity.

6) *Square Root Algorithm*: Computation of the  $N_t \cdot N_t$  matrix inverse in SIC is complex. One method to simplify this is by SR decomposition, where  $H$  is decomposed into a unitary matrix  $Q$  and an upper triangular matrix  $R$ , satisfying:  $H = QR$ ,  $Q^\dagger Q = I$ ,  $P^{1/2} = R^{-1}$ ,  $H^\dagger = P^{1/2} Q^*$ , and  $P^{1/2} P^{*/2} = P$ . The nulling vector for the  $i$ -th signal after permutation to the  $N_t$ -th row can be given by  $P_{N_t}^{1/2} q_{N_t}^*$ , where  $q_{N_t}^*$  is the  $N_t$ -th row of  $Q^*$  after updating. [5]

7) *SM Algorithm*: In [6] it is shown that the SR algorithm can be further simplified, and another SIC MMSE algorithm is described, using the Sherman-Morrison formula, (noted SM). The algorithm can be summarized as:

Initialization:

1)

$$R = H^H H + \frac{1}{SNR} I_{N_t \times N_t},$$

$$Q[l] = Q[l-1] - \frac{Q[l-1]H^H(l,:)H(l,:)Q[l-1]}{1 + H(l,:)Q[l-1]H^H(l,:)}$$

with

$$Q[0] = SNR \cdot I_{N_t \times N_t}$$

$$Q[N_r] = [H^H H + \frac{1}{SNR} I_{N_t \times N_t}]^{-1}$$

- 2)  $f(k) = (1, 2, \dots, N_t)$
- 3)  $l_i = k_i = \arg_m \min q_{mm}$
- 4)  $y_{k_i} = Q(k_i,:)H^H r$
- 5)  $\hat{s}_{k_i} = \text{quantify}(y_{k_i})$

Recursion for  $i = 1..N_t - 1$ :

- 1)  $r = r - s_{k_i} H(:, k_i)$
- 2) Permutate the entries  $l_m$  and  $N_t - i + 1$  of  $f$
- 3) Permutate the rows and columns  $l_m$  and  $N_t - i + 1$  of  $R[N_t - i + 1]$
- 4) Permutate the rows and columns  $l_m$  and  $N_t - i + 1$  of  $Q[N_t - i + 1]$
- 5) Determine  $R, V, \beta_{k_i}$  from  $R[N_t - i + 1]$
- 6) Determine  $T$  from  $Q[N_t - i + 1]$
- 7) Let  $a = N_t - i$

$$Q[a] = T[a] - \frac{T[a]V[a]V^H[a]T[a]}{\beta_{k_i} + V^H[a]T[a]V[a]}$$

- 8)  $l_i = \arg_m \min q_{mm}, k_i = f(l_i)$
- 9)

$$y_{k_i} = \sum_{j=0}^{N_t-i} Q(l_i, j)h^H(:, f(j))r$$

- 10)  $\hat{s}_{k_i} = \text{quantify}(y_{k_i})$

### G. Detection Algorithms Comparison

In general the performance of the MF algorithm is to low, while ML, ZF and MMSE are complex to implement, so SIC is the most comminly used algorithm. In [6], the computation cost for three decomposition methods for implementation of SIC, (Singular Value Decomposition (SVD), SR, and SM) can be found. The result is summarized in table I, where the transmitter and the receiver have the same number of antennas  $M = N_t = N_r$ .

TABLE I  
ALGORITHM COMPARISON

	SVD	SR	SM
MUL	$\frac{43}{12}M^4 + \frac{22}{3}M^3 + O(M^2)$	$\frac{29}{3}M^3 + O(M^2)$	$\frac{11}{3}M^3 + O(M^2)$
ADD	$\frac{43}{12}M^4 + \frac{20}{3}M^3 + O(M^2)$	$\frac{29}{3}M^3 + O(M^2)$	$3M^3 + O(M^2)$

It is apparent that SM is more efficient that the other two, so we focus on the SM algorithm.

### H. Flow chart

The SM algorithm as pllied after synchronization can be described as follows: The channel coefficient  $H$  is calculated by the process shown in Flow 1, then the weighting vector  $w$  is calculated, based on  $H$ , as shown in Flow 2. Finally, the original signal  $s$  is recovered using  $w$ , following the process in Flow 3.

#### Flow 1: Channel estimation

- 1) For  $m = 0..N_r$  execute step 2
- 2) For  $n = 0..N_t$  execute step 3,4,8
- 3) Remove CP, FFT
- 4) For  $i = 0..N_t$  execute step 5,7
- 5) For  $k = 0..N_k$  execute step 6
- 6)  $R_{xy}(k, i) = r(k)s_i^*(k)$
- 7)  $P_i(\cdot) = IFFT(R_{xy}(\cdot, i))$
- 8) For  $i = 0..N_t$  execute step 9,12
- 9) For  $k_1 = 0..K_0$  execute step 10-11
- 10)

$$y_i(k_1) = \sum_{i \neq j} \sum_{k_2=1}^{K_0} Q_{j,i}(k_1, k_2) \hat{h}(k_2)$$

- 11)  $h_i(k_1) = [P_i(k_1) - y_i(k_1)]/N_k$
- 12)  $H_{m,i}(\cdot) = FFT(j_i(\cdot))$

#### Flow 2: Detection

- 1) For  $n = 0..N$  execute step 2,4
- 2) For  $m = 0..N_r$  execute step 3
- 3) Remove CP, FFT
- 4) For  $k = 0..N_k$  execute step 5
- 5) For  $i = 0..N_r - 1$  execute step 6-7
- 6)

$$y_{k_i} = \sum_{m=1}^{N_r} w_{m,i}(k) r_m(k)$$

- 7)  $s_{k_i}(k) = \text{quantify}(y_{k_i})$
- 8) For  $m = 0..N_r$  execute step 9
- 9)  $r_m(k) = r_m(k) - s_{k_i}(k)H_{m,k_i}(k)$

#### Flow 3: Computing nulling vector

- 1) For  $k = 0..N_k$  execute step 2,3,7-8,18-19,28,31-32
- 2)  $R = \frac{1}{SNR} I_{N_t \times N_t}$
- 3) For  $i = 0..N_t$  execute step 4
- 4) For  $j = 0..i$  execute step 5-6
- 5)

$$R_{i,j} = \sum_{n=1}^{N_r} H_{n,i}^*(k) H_{n,j}(k)$$

- 6)  $R_{j,i} = R_{i,j}^*$
- 7)  $T[0] = SNR \cdot I_{N_t \times N_t}$
- 8) For  $l = 0..N_r$  execute step 9,13-14
- 9) For  $i = 0..N_t$  execute step 10-11
- 10)

$$ht_i = \sum_{n=1}^{N_t} H_{l,n}(k) T_{n,i}$$

- 11) For  $j = 0..i$  execute step 12
- 12)  $up_{m,j} = vt_m^* vt_j$
- 13)

$$dn = \sum_{n=1}^{N_t} H_{l,n}^*(k) ht_n$$

- 14) For  $i = 0..N_t$  execute step 15
- 15) For  $j = 0..i$  execute step 16-17
- 16)  $T_{i,j} = T_{i,j} - up_{i,j}/dn$
- 17)  $T_{j,i} = T_{i,j}^*$  ( $j \neq i$ )
- 18)  $l_i = k_i = \text{argmin}(T_{i,i})$
- 19) For  $i = 0..N_r$  execute step 20
- 20)

$$w_{i,1}(k) = \sum_{n=1}^{N_t} T_{k_i,n} H_{i,n}^*(k)$$

- 21) For  $i = 0..N_r - 1$  execute step 22-23,27-28,32-33
- 22) Permutate rows and columns  $l_i, N_t - i + 1$  of  $R, T$
- 23) For  $m = 0..N_t - i$  execute step 24-25
- 24)

$$vt_m = \sum_{n=1}^{N_t-i} R_{n,N_t-i+1}^* T_{n,m}$$

- 25) For  $i = 0..m$  execute step 26
- 26)  $up_{m,j} = vt_m^* vt_j$
- 27)

$$dn = R_{N_t-i+1,N_t-i+1} + \sum_{n=1}^{N_t} vt_m R_{m,N_t-i+1}$$

- 28) For  $m = 0..N_t - i$  execute step 29
- 29) For  $j = 0..m$  execute step 30-31
- 30)  $T_{m,j} = T_{m,j} - up_{m,j}/dn$
- 31)  $T_{m,i} = T_{m,j}^*$  ( $j \neq i$ )
- 32)  $l_i = \text{argmin}(T_{m,m})(m < N_t - i), k_i = f(l_i)$
- 33) For  $m = 0..N_r$  execute step 34
- 34)

$$w_{m,i+1}(k) = \sum_{n=1}^{N_t-i} T_{k_i,n} H_{m,f(n)}^*(k)$$

### III. COMPUTATION PROFILING AND COMPLEXITY ANALYSIS

Based on the previous section, the most common operations used can be summarized as:

- 1)  $N_k$  point FFT
- 2) Complex multiplication and accumulation
- 3) Complex vector multiplication
- 4) Complex vector subtraction

#### A. Definition of Benchmarking and MIPS

MIPS cost computations have been made for a programmable DSP processor architecture optimized for baseband processing [7]. The numbers are based on the following assumptions:

- 1) One Complex valued MAC per cycle

- 2) One radix-2 FFT butterfly per cycle;
- 3) Memory operations run in parallel behind computing;
- 4) Prolog and epilog costs and other overhead is around 10% of the computing cost.

#### B. Architecture Proposal

In this section, we propose a processor architecture suitable for implementing OFDM-VBLAST systems onto. We consider a homogenous multi-processor system where each processor node is a DSP-processor containing:

- Native complex data-paths
- A double CMAC-unit capable of performing a Radix-2 butterfly per cycle
- Local memories
- A RISC control unit
- Network interface for fast data-transfer

An overview of the processor architecture is presented in Figure 3.

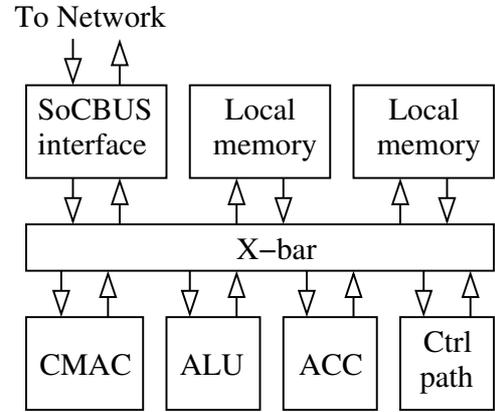


Fig. 3. Overview of processor node

Several processor nodes are connected to each other an interface blocks trough an on-chip network, SoCBUS [8]. SoCBUS is a low-latency on-chip network which allows the communicating cores to set up dedicated data channels between each other.

In addition to processor nodes, the system also contain input and output nodes as well as dedicated memory nodes to complement the local memory in each processor node. Input and output nodes are responsible for the interface towards the ADC/DAC for each radio as well as the interface towards the bit manipulation part of the system.

An overview of the complete system is presented in Figure 4.

#### C. MIPS costs for channel estimation

The computational complexity is concentrated on two steps: computing the  $P$  matrix based on the signal received, and computing the channel matrix based on the  $P$  and  $Q$  matrices.

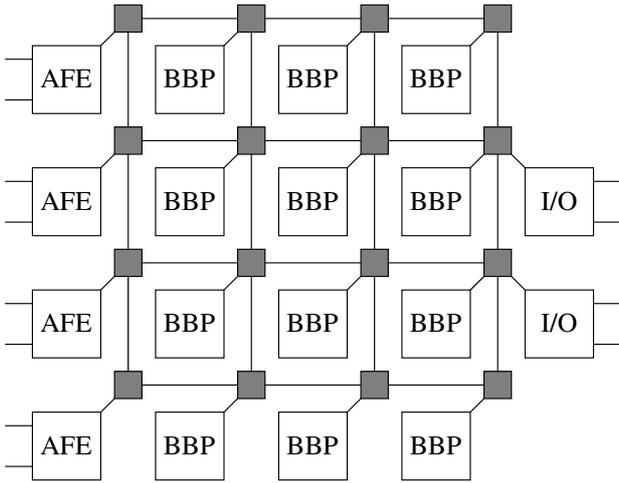


Fig. 4. Complete system

1) *Computing the P matrix:* The  $P$  matrix can be expressed as:  $P = IFFT(FFT(r) \cdot s)$ , where  $r$  is a received symbol at one receive antenna after synchronization in time domain, and  $S$  is the  $N_t \times N_t$  preamble symbol matrix for the corresponding sub-carrier.

The cycle costs of the (I)FFT for one symbol is  $\frac{N_k}{2} \log_2 N_k$ . The cost of multiplication with  $N_t$  transmitted preamble symbols stored at the receiver end, is  $N_t N_k$ . For  $N_r$  receive antennas, the total MIPS cost for processing a received symbol is  $N_r N_k [N_t (\frac{1}{2} \log_2 N_k + 1) + \frac{1}{2} \log_2 N_k]$ .

2) *Channel matrix computation:*  $H(n) = FFT(P - Q \cdot h(n-1))$  is the channel matrix for channel compensation. The multiplication of an  $N_t K_0 \times N_t K_0$  matrix  $Q$  and  $N_t K_0$  size vector  $h$  of  $H(n)$  costs  $(N_t K_0)^2$  cycles. After adding the cost of subtraction and FFT, the total cost for one received symbol at  $N_r$  receive antennas is  $N_r N_t (\frac{N_k}{2} \log_2 N_k + K_0 + K_0^2 N_t)$ . Therefore, the total cycle cost for one received symbol at  $N_r$  receive antennas is  $N_r \{N_k [(N_t + \frac{1}{2}) \log_2 N_k + N_t] + K_0 N_t (K_0 N_t + 1)\}$ .

Given our example with four transmit antennas and four receive antennas, that is,  $N_t = 4$ ,  $N_r = 4$ ,  $N_k = 64$ ,  $K_0 = 10$ , where  $N_k$  and  $K_0$  are based on the experience from IEEE 802.11a [9], the total cost for channel estimation is  $4\{64[(4 + \frac{1}{2}) \log_2 64 + 4] + 10 \times 4(10 \times 4 + 1)\} = 14496$  complex valued instructions. If these computations must be completed in one symbol time ( $4\mu s$ ) the corresponding MIPS cost is 3624 MIPS, for reaching 4 times the data rate of IEEE 802.11a. Assuming the previously mentioned processor core could perform four complex operations per clock cycle, DEBUG MIPS could be mapped to each processor DEBUG such processors are needed to handle the computational load in this example.

#### D. MIPS costs for detection

For initialization, the cycle cost for calculating  $R$  is  $N_t N_r (N_t + 1)/2 + N_t$ , for  $T$  the cost is  $\frac{5}{2} N_t N_r (N_t + 1)$ , for  $w_{k_i}$  it is  $N_t N_r$ , and for  $y_{k_i} = q_{k_i} r$  it is  $N_r$ . For each loop  $i$  in the  $(N_t - 1)$  recursions, the cost for cancellation is

$2N_r$ , for computing  $T$  the cost is  $\frac{5}{2}(N_t - i)(N_t - i + 1)$ , for computing  $w_{k_i}$  it is  $N_r(N_t - i)$ , and for  $y_{k_i}$  it is  $N_r$ .

Considering  $N_k$  sub-carriers, the total cost for one received symbol at  $N_r$  receive antennas for channel compensation can be summarized as:  $3N_r N_k N_t - N_r N_k$  for each data symbol plus:

$$N_k \left\{ \sum_{i=1}^{N_t-1} \left[ \frac{5}{2} (N_t - i)(N_t - i + 1) + N_r (N_t - i) \right] + N_t N_r (3N_t + 4) + N_t \right\}$$

for each data frame.

Continuing on the example, with  $N_t = 4$ ,  $N_r = 4$ ,  $N_k = 64$ ,  $K_0 = 10$ , the total cost for channel compensation is:

$$64 \left\{ \sum_{i=1}^{4-1} \left[ \frac{5}{2} (4 - i)(4 - i + 1) + 4(4 - i) \right] + 4 \times 4 (3 \times 4 + 4) + 4 \right\} = 21376$$

instructions for one data frame, plus  $3 \times 4 \times 64 \times 4 - 4 \times 64 = 2186$  instructions for one data symbol. Again these calculations must be completed in one symbol time of  $4\mu s$ .

The channel compensator can be implemented with four additional cores.

## IV. CONCLUSION

This paper has presented estimations of MIPS costs for channel estimation and detection algorithms in an OFDM-VBLAST system, for MIMO transceiver implementations using programmable processors instead of ASIC. The final MIPS cost for each algorithm have been estimated. With the example of using four transmit and four receive antennas, to achieve 4 times bandwidth of IEEE 802.11a standard, the MIPS cost for the algorithms shows the feasibility of such an implementation.

## REFERENCES

- [1] Gerard J. Foschini. Layered Space-Time Architecture for Wireless Communication in a Fading Environment When Using Multi-Element Antennas. Bell Labs Technical Journal, pages 41-59, autumn 1996.
- [2] Ye Li. Simplified Channel Estimation for OFDM Systems With Multiple Transmit Antennas. IEEE Transactions on Wireless Communications, volume 1, NO. 1, pages 67-75, January 2002.
- [3] Ye Li, J. H. Winters and N. R. Sollenberger. Signal Detection for MIMO-OFDM Wireless Communications. IEEE International Conference on Communications, pages 3077-3081, May 2001.
- [4] Gordon L. Stuber, John R. Barry, Steve W. McLaughlin, Ye Li, Mary Ann Ingram, and Thomas G. Pratt. Broadband MIMO-OFDM Wireless Communications. Proceedings of the IEEE, volume 92, NO. 2, pages 271-294, February 2004.
- [5] Babak Hassibi. An Efficient Square-Root Algorithm for BLAST. Mathematics of Communications Research, January 27, 2000.
- [6] Jacob Benesty, Yiteng Huang, and Jingdong Chen. A Fast Recursive Algorithm for Optimum Sequential Signal Detection in a BLAST System. IEEE Transactions on Signal Processing, Volume 51, NO. 7, pages 1722-1730, July 2003.
- [7] Dake Liu, Eric Tell, Low-Power Baseband processors for Communications, ch. 23 in Low power Electronics Design, edited by C. Piguet, CRC 2004, ISBN 084931941 2.
- [8] Daniel Wiklund och Dake Liu SoCBUS: Switched Network on Chip for Hard Real Time Systems Proc of the International Parallel and Distributed Processing Symposium (IPDPS), Nice, France, Apr 2003
- [9] IEEE 802.11a standard, 1999.