Abstract- Suitable techniques for High-speed Downlink Packet Access (HSDPA) services, which is using in 3G and 3G beyond, are currently being considered by the Third Generation Partnership Project (3GPP). One of the systems currently under consideration is a Space-Time Coding (STC) architecture using the Vertical- Bell-labs-Layered-Space-Time (V-BLAST) system on the High Data Rate Downlink-Shared Channel (DSCH), which will allow simultaneous transmissions over Multiple-Input-Multiple-Output (MIMO) channels destined for a single user. Although link level results for flat fading channels indicate significant performance advantages over the conventional single antenna systems, it is less successful for frequency-selective fading. Corresponding methods of design implementation are also being developed using techniques such as System-on-a-Chip (SoC), Digital Signal Processors (DSPs), and Field Program Gate Arrays (FPGA). In this paper, we present a performance comparison for both fixed and floating point DSP implementations for the proposed V-BLAST system.

1. INTRODUCTION

Multiple Input Multiple Output (MIMO) antenna processing is being considered for HSDPA services using Vertical Bell-labs Layered-Space-Time (V-BLAST) coding [1, 2, 3]. The services employ high data rates (> 10Mbps) for the 3G WCDMA air interface. HSDPA services in MIMO channels are still under development therefore the transfer to hardware must be flexible. In the paper we present our solution for mapping the design to DSP devices that’s facilitates the flexibility required.

V-BLAST utilizes the spatial and temporal diversity in to enhance spectrum efficiency using a code re-use technique. Code re-use is achieved by spreading the independent data streams using the same code set. The VBLAST system defined in this paper was originally developed using C-Code models within a Mathworks Simulink® environment. This model was used as the test bench to ascertain the scaling and range requirements of each component block to produce a effective fixed point implementation. Having defined the requirements for a fixed point implementation the design was mapped to real-time implementation on a SHARC DSP development board. The Simulink® testbench was then modified to control the implementation and provide an effective channel model to test the system against.

In the following section we first define the system model used in this paper based on the definitions prescribed by HSDPA service for 3GPP. In sections 3 and 4 we describe the techniques used to develop the fixed point system and the Section 5 presents the results of our implementation and compares the system performance for both fixed and floating point implementations. Finally we draw conclusions from our results and offer possible improvements to the system which would aid final implementation.

2. SYSTEM MODEL

The system model uses a base-band signal representation can be subdivided into three main components: transmitter, fading channel, and receiver. The block diagram of an M antenna transmitter is shown in Figure 1 where the HSDPA system based on the Downlink Shared Channel (DSCH) [4] creates a frame structure for a single user.

The DSCH high data source is de-multiplexed into QM sub-streams where Q is the number of spreading codes, devoted to the higher layer. The gth (0 ≤ g < M) group of the M sub-streams is spread by the nth (0 ≤ n < Q) spreading code . The sub-streams sharing the same spreading code are then transmitted over different antennas.

Based on the mth antenna, the base-band transmitted signal can be represented as

$$x_m = \sum_{q=0}^{Q} \left( A / \sqrt{M} \right) d_q b_{m,q} \quad (m = 1...M)$$

(1)
where \((A/\sqrt{M})\) is the signal amplitude (power), \(d_k\) is the \(q\)th spreading code (complex), and \(h_{m,q}\) is the coded (modulated) symbols for the \(q\)th code on antenna \(m\) [5].

The MIMO channel model used is a discrete-time complex base-band model, which is assumed to be stationary over every burst of data, but changes from burst to burst (i.e., quasi-static channel). Each sub-channel (between a pair of transmit-receive antennas) has 2x2 multi-paths as shown in Figure 2.

\[
r_p(t) = \sum_{m=1}^{M} \sum_{q=1}^{Q} \sum_{l=1}^{L} h_{m,q}(A/\sqrt{M})d_q(t-l)Bb_{mq} + n_p (m=1...M)
\]

where \(r_p\) is a complex received signal with \(N\) chips per symbol, \(h\) is the Rayleigh channel coefficients as shown in equation (2), and \(n_p\) is the AWGN with power spectral density (PSD) \(N_0\). Here, the effect of the Doppler components, \(K\), is neglected. Following dispersing the system estimates a set of complex data.

Let \(H\) denote the \(LP \times M\) matrix whose \((l,m)\)th component is \(h_{l,m}\). We assume we obtain the perfect channel coefficients \(\{h\}\) of \(m\)th antenna and \(l\)th multipath which are then passed to the other parts of the receiver, i.e. perfect channel estimation. Let \(\overline{H} \equiv I_q \otimes H\) denote the \(QLP \times QM\) block diagonal channel matrix (for \(L\) multipath spread) defined by the Kronecker product of \(I_q\), the \(Q \times Q\) identity matrix, and \(H\). Thus, let \(\bar{b} \equiv [b_{1,1}...b_{M,1}...b_{Q,1}...b_{M,Q}]^T\) denote the \(QM\)-dimensional data vector, and let \(\bar{n} \equiv [n_1...n_Q]^T\) denote the stacked \(NP\)-dimensional noise vector [6]. Stacking the received vectors to form an \(NP\)-dimensional vector [6], we can write:

\[
r = (A/\sqrt{M})\overline{H}\bar{b} + \bar{n}
\]

A \(QM\)-dimensional statistic vector \(x\) for the data vector \(b\) is obtained by performing a matched filter operation on \(r\). Let \(x \equiv [x_1^T...x_Q^T]\) split into \(Q\) vectors for each of dimensions \(M\), it follows from (4) with respect the channel matrix \(\overline{H}\) such that:

\[
x_q = (A/\sqrt{M})\overline{H}\bar{b}_q + \bar{n}_q
\]

where \(b_q \equiv [b_{1,1}...b_{M,q}]\) is the data vector corresponding to the spreading code \(q\), and \(n_k\) is the \(q\)th \(M\)-dimensional subvector of the noise term \(\overline{H}\bar{b}\) with covariance is \(\overline{H}\bar{H}\sigma^2\). Thus we have defined the operation of the so-called Space-Time Rake Combiner [6].

Thus \(x_k\) can then be applied V-BLAST detector to be processed independently [5]. Using the revised sufficient statistic vector, the linear transformation and ordered successive interference cancellation is repeated until all sub-streams have been detected [5]. A single V-BLAST detector is implemented for each \(y_k\) using minimum mean square error (MMSE) / Zero-Force (ZF) detector. For a frequency selective fading channel, the VBLAST algorithm requires estimation of an \(M\)-by-\(M\) code-channel correlation matrix.
\[
R_k = \sum_{p=1}^{P} \mathbf{H}_p^H \mathbf{F}_q \mathbf{H}_p
\]

where \( \mathbf{F}_q \) is the \( L \times L \) code correlation matrix for the \( q \)th code. The \((i, j)\)th element of \( \mathbf{F}_q \) is the inner product of the \( i \)th delayed Walsh sequence with the \( j \)th delayed Walsh sequence as defined in [6]. The sufficient statistic vector is:

\[
y_k = \mathbf{Rb}_q + \mathbf{n}
\]

where \( \mathbf{b}_q \) is the \( M \)-dimensional data vector spread by the \( k \)th code, and \( \mathbf{n} \) is the associated complex additive Gaussian noise vector. All \( M \) components have been detected by a repeated procedure loop.

Following the V-BLAST detectors, the sub-streams are multiplexed, de-interleaved, and decoded to yield the final decoded high-speed data stream.

### 3. FIXED-POINT PROCESSING

Portable computing is becoming pervasive in our society, particularly with the emergence of 3G mobile phone systems and wireless devices. The mobile terminals themselves universally rely on fixed-point Digital Signal Processing (DSP) devices because of their innate, higher speed, lower cost, and lower power consumption compared with their floating-point counterparts. These issues are not confined to portable appliances with the majority of high volume products requiring DSP utilising fixed-point processors.

<table>
<thead>
<tr>
<th>Fixed point</th>
<th>Floating point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low number precision</td>
<td>High number precision</td>
</tr>
<tr>
<td>Small Dynamic range</td>
<td>Large Dynamic range</td>
</tr>
<tr>
<td>Low power consumption</td>
<td>High power consumption</td>
</tr>
<tr>
<td>Low device cost</td>
<td>Higher device cost</td>
</tr>
<tr>
<td>High development overhead</td>
<td>Low development overhead</td>
</tr>
</tbody>
</table>

In fixed-point processors, numbers are represented as integers (integer arithmetic) or as fractions \(-1.0\) and \(+1.0\) (fractional arithmetic). The algorithms and hardware used to implement fractional arithmetic are virtually identical to those used for integer arithmetic. In practice, most fixed-point DSP processors support fractional arithmetic and integer arithmetic. The former is most useful for signal processing algorithms, while the latter is useful for control operations, address calculations, and other operations that do not involve signals.

The format most commonly preferred by engineers is the Q notation which indicates the scaling system in use with a signed fixed-point value. A fixed-point representation is actually defined by two numbers: the word length \( N \); and the scaling exponent \( n \). However, in practice \( N \) is usually either 16 or 32 bits. The Q-format of a variable is, in effect, given by the value of \( n \) used in its representation, and the binary symbol is then read as an integer.

For example, a number might be given as 8192 (decimal) in Q15 format. In binary, this symbol is 0010000000000000 (assuming a 16-bit data word). As it is a Q15 value, we know that there are fifteen digits after the implied binary point, so the number actually being represented is \(+0.0100000000000000\), or in decimal, \(+0.25\). The conversion can be done more easily using a scaling factor, \( S \), which we set to \( 2^{15} = 32768 \). This gives us \( 8192 / S = 0.25 \). We can obtain the representation of 0.25 in Q14 format by setting \( S = 2^{14} \), and now \( S \times 0.25 = 4096 \).

An easy way to grasp the conversions is by looking at the ends of ranges. Sixteen-bit signed integer range is from \(+32767 \) (0111111111111111) to \(-32768 \) (1000000000000000). Q15 values can therefore range from \(+32767 / 2^{15} \) to \(-32768 / 2^{15} \) which is \(+0.9999694 \) to \(-1.0 \). Q14 range is from \(+1.999939 \) to \(-2.0 \), and so on.

Q-format is used throughout this implementation and specifics of the ranges used will be presented in the forthcoming sections.

### 4. SIMULATION ENVIRONMENT

An extensive simulation environment has been developed using the Simulink® toolset as the control environment for the main fixed point implementation. C code on its own does not support compilation onto a DSP platform [7] and Analog Devices VisualDSP® provides the interface to the host device in this case the ADSP-2116x family.

This combination of C-code DSP implementation and Simulink® environment is becoming an increasingly popular configuration which can drive the system design and implementation at increased speed. To quote Doug Pulley Chief Technical Officer of PICOCHIP this drive is through:

"One-third chip implementation; one-third tools; and one-third system architecture"

The block diagram of the simulation environment is shown in Figure 4 and affords full control of the different requirements including data-generation results analysis.
The Hardware platform used in the investigation is the Bittware Hammerhead-Lite-PCI (HLPC) development board. This PCI development board incorporates four low-cost 100 MHz ADSP-21161 SHARC DSPs, a 32-bit 33 MHz PCI interface, 64 MB SDRAM, 512 K SRAM, and up to 4MB flash memory [9]. The Analog Devices ADSP-21161 SHARC DSP is a member of the Super Harvard Architecture (SHARC) family of programmable digital signal processors (DSPs) providing 600 MFLOPS of processing power. The ADSP-21161 supports both fixed and floating-point data types for greater design flexibility although we concentrate on a purely fixed point implementation as we feel this is more representative of a final implementation.

5. RESULT

The following results are based on the system model previously defined in section 2. We have assumed a spreading factor of Q=32 and the modulation to be 64 QAM. The signal is transmitted at 120K symbols/second in the frequency-selective fading channel with two paths and the ratio between the paths (multipath gain) is a small value.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise and Interference</td>
<td>frequency-selective fading (2 paths) + AWGN</td>
</tr>
<tr>
<td>Channel code</td>
<td>Ideal channel code</td>
</tr>
<tr>
<td>Chip rate</td>
<td>3.84Mcps</td>
</tr>
<tr>
<td>Packet size</td>
<td>8000bits</td>
</tr>
<tr>
<td>Spreading factor</td>
<td>32</td>
</tr>
<tr>
<td>Spreading code available</td>
<td>32</td>
</tr>
<tr>
<td>Multipath gain</td>
<td>-30dB</td>
</tr>
</tbody>
</table>

Table 2: Simulation parameters

Our principle aim is to implement the VBLAST receiver algorithm using fixed-point arithmetic. Although Simulink® can be used as development tool for direct operation on the HLPC, in this case VisualDSP® was used to port C-code into Simulink® S-functions which was then evaluated in the original simulation before being tested and debugged on the DSP board.

The DSCH data is generated in the test-bed, and transmitted using the HSDPA service scheme. We assume that the received data range is not greater than ±10 from combiner (see section 2). We therefore scale these values to Q12 format fixed-point value. The received signals do not require manual scaling within the software as the signals are quantised to 16-bit via the PCI link. The synchronized channel coefficients also employ a Q-format fixed-point arithmetic, and are stored into a corresponding memory address after being pre-scaled. Figure 5 illustrates the overall system process.

The same receiver functionality was implemented using floating-point arithmetic to produce a ‘true’ performance comparison with the fixed-point system rather than relying upon the original simulation results.

Table 3 shows the processing time of the different VBLAST receiver implementations, where $T$ is processing time. As expected the fixed-point processing is faster than floating-point processing by up to 25, although this figure can be improved using further code optimization which was outside the scope of this particular implementation.

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Processing Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Floating-point</td>
<td>100%$T$</td>
</tr>
<tr>
<td>Fixed-point</td>
<td>88.6%$T$</td>
</tr>
</tbody>
</table>

Table 3: Processing time of each VBLAST unit
In figure 6, we consider the system Bit Error Rate performance for both fixed- and floating-point VBLAST receiver implementations using identical MIMO channel models.

The figure shows an expected drop in performance for fixed-point arithmetic system compared with the system using floating-point arithmetic. The worst case reduction is approximately 2 dB which although is by no means insignificant could be reduced further by more intensive use of code optimisation.

![Figure 6: System performance](image)

6. CONCLUSION

This paper has shown the performance of a fixed-point implementation of the proposed scheme for HSDPA service currently being considered by 3GPP. There is no doubt that fixed-point implementation represents the most efficient and cost effective implementation for systems related to portable and mass produced systems and therefore represents the most realistic system implementation. Through simulating this system in a combined software/hardware environment we could move quickly from the theoretical simulation to a realistic system implementation.

It should be noted that the scheme of HSDPA service is still developing and the final system is likely to be somewhat different due to intrinsic problems associated with this system in a frequency selective fading environment. However, the techniques used in the production of these results shows how modern DSP tools can be combined to produce a flexible simulation/implementation environment.

REFERENCES


[3] Lucent Technologies, “Enhancements for HSDPA Using Multiple Antennas”, 3GPP, TSG_R WG1 document TSGR1#15(00) 1096, August 2000, Berlin, Germany

[4] Technician specification group radio access network, “Physical channels and mapping of transport channels onto physical channels (FDD)”, 3GPP, TS 25.211, V4.2. 2001


