

Efficient DFE Architecture for MIMO Communication Systems

Wen-Chih Kan and Gerald E. Sobelman

Department of Electrical and Computer Engineering

University of Minnesota

Minneapolis, MN 55455, USA

kanx0008@umn.edu, sobelman@umn.edu

Abstract - We present a novel VLSI architecture of Multiple-input, multiple-output (MIMO) decision feedback equalization (DFE) which is based on the reduced complexity FIR filter structure. Compared with the SISO counterpart, a MIMO DFE has a formidable computation complexity. By applying the reduced complexity FIR filter techniques that reduce the use of multipliers at the expense of some additional switches, at least a 25% reduction in multipliers can be achieved. A 2×2 MIMO DFE design example is given. Complexity comparisons based on the implementations on Xilinx Virtex-4 are also discussed where a 20% reduction in slices used is reported.

Keywords: MIMO, DFE, reduced complexity, decision feedback, Xilinx, slice.

1 Introduction

A primary design goal of digital communications systems is to maximize the rate of reliable transmissions. MIMO techniques are one powerful way to achieve high throughput. On the other hand, a dispersive channel distorts the transmitted signals in the form of inter-symbol interference (ISI), which if not correctly compensated can lead to a high bit error rate. Equalization has been the solution for the ISI problem. In particular, decision feedback equalization (DFE), [7], has been shown to have good performance with moderate complexity. The MIMO system model used in this paper is based on a general linear, dispersive and noisy digital communication system [1]. MIMO DFE has been an active research topic for the past few years, [1] ~ [6] but there have not been any papers

discussing the complexity impact on the hardware implementation of the system. Based on this assumption, we will discuss the MIMO DFE in section 2 where a complexity comparison with its Single-input, Single-output (SISO) counterparts is made. Since the DFE complexity increases dramatically for the MIMO system, a low complexity technique is presented in section 3. A 2×2 MIMO DFE design example is given in section 4 with complexity comparisons from the implementation on Xilinx Virtex-4 FPGA. Section 5 is the conclusion section.

2 MIMO DFE

Figure 1 shows a generic DFE architecture where the filter is composed of a feedforward FIR (FFF) and feedback FIR (FBF). If the system is a 2×2 MIMO system, each multiply-add actually represents a 2 by 2 matrix multiplying a 2 by 1 vector and an addition of two 2 by 1 vectors. In this case, the actual multiplication complexity is 4-fold of the SISO system and actual addition is 4-fold. It's then very straightforward to verify the complexity comparisons in Table 1 where we assume that the numbers of feedforward and feedback taps are N_f and $N_b + 1$ respectively.

Table 1 Complexity comparison of SISO and MIMO

| System | Multiplier | Adder |
|--------------------------|-----------------------------|-----------------------------|
| SISO | $N_f + N_b + 1$ | $N_f + N_b$ |
| MIMO ($n \times n$) | $n^2 \cdot (N_f + N_b + 1)$ | $n \cdot (2N_f + 2N_b + 1)$ |

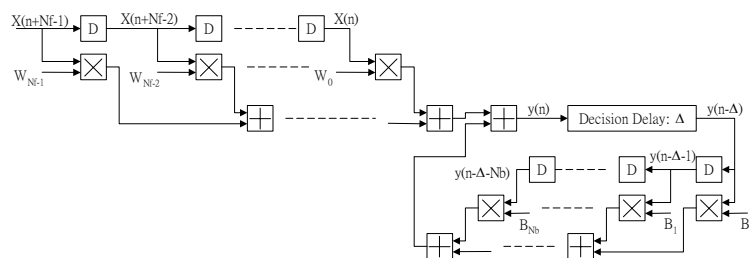


Figure 1 A generic DFE architecture

3 MIMO DFE based on Reduced complexity FIR architecture

A careful examination of Figure 1 indicates that the DFE actually is composed of two FIR filters, namely the FFF and FBF. Therefore, the reduced complexity FIR (RCFIR) architecture, as in [8], can be applied in implementing MIMO DFE to save chip area. Before the low complexity MIMO DFE is presented, the RCFIR is discussed below.

3.1 RCFIR architecture

The RCFIR can only be achieved when the FIR is first converted to n -parallel. Throughout this paper, we'll assume n is 2; extension to a higher dimension is straightforward. Let's suppose the data sequence as in Eq. (1) is to be broken into 2 sequences for 2-parallel processing.

$$X(z) = x(0) + x(1)z^{-1} + x(2)z^{-2} + x(3)z^{-3} + \dots \quad (1)$$

To achieve the 2-parallel processing, one can rewrite Eq. (1) as Eq. (2) and the 2-parallel sequences are then derived as given $X_0(z^2)$, an even sequence, and $X_1(z^2)$, an odd sequence, in Eq.(3).

$$X(z) = x(0) + x(2)z^{-2} + x(4)z^{-4} + \dots + z^{-1}\{x(1) + x(3)z^{-2} + x(5)z^{-4} + \dots\} \quad (2)$$

$$X(z) = X_0(z^2) + z^{-1}X_1(z^2) \quad (3)$$

If both output sequence and impulse response are also transformed into even and odd sequences for 2-parallel processing, a 2-parallel FIR filter structure can be derived, Eq. (4) and Figure 2.

$$\begin{bmatrix} Y_0 \\ Y_1 \end{bmatrix} = \begin{bmatrix} H_0 & z^{-2}H_1 \\ H_1 & H_0 \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \end{bmatrix} \quad (4)$$

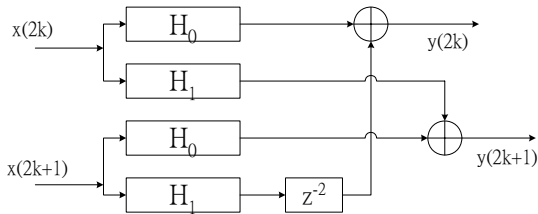


Figure 2 A 2-parallel FIR filter

Figure 3 is the reduced complexity 2-parallel FIR filter. Its derivation can be found in [8]. However, some aspects are worth noting. A 2-parallel architecture processes twice the data at the expense of twice the

hardware. The hardware utilization efficiency (HUE) can be 100% if one can multiplex 2 sets of data where both sets should be transformed into even and odd sequences. The 2-parallel RCFIR decreases the complexity by almost 25% which can be easily understood by comparing Figure 2 and Figure 3. In Figure 2, there are 4 FIRs. However there are only 3 FIRs in Figure 3. The H_0+H_1 can be pre-computed.

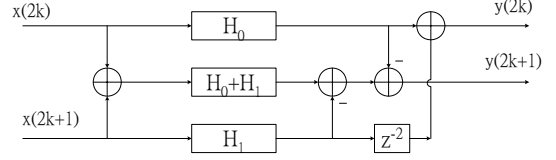


Figure 3 A Reduced Complexity 2-parallel FIR filter

3.2 Direct implementation of MIMO DFE

As in the Figure 1, a 2×2 MIMO DFE can be modeled by Eq (5).

$$y(n) = W_0x(n) + W_1x(n+1) + \dots + W_{N_f-1}x(n+N_f-1) - \{B_0y(n-\Delta) + \dots + B_{N_b}y(n-\Delta-1)\} \quad (5)$$

Since we can treat the FFF and the FBF parts as two individual FIRs, we let

$$i(n) = W_0x(n) + W_1x(n+1) + \dots + W_{N_f-1}x(n+N_f-1) \quad (6)$$

and

$$z(n) = B_0y(n-\Delta) + \dots + B_{N_b}y(n-\Delta-1) . \quad (7)$$

Multiplying out all the $W_i x(n+i)$ terms in Eq. (6) and assuming $x^0(n)$ represents data coming out of

antenna 1, $x^1(n)$ antenna 2, and $W_i = \begin{bmatrix} w_i^{11} & w_i^{12} \\ w_i^{21} & w_i^{22} \end{bmatrix}$, Eq. (6) becomes

$$i^0(n) = w_0^{11}x^0(n) + w_1^{11}x^0(n+1) + \dots + w_{N_f-1}^{11}x^0(n+N_f-1) + w_0^{12}x^1(n) + w_1^{12}x^1(n+1) + \dots + w_{N_f-1}^{12}x^1(n+N_f-1) \quad (8)$$

and

$$i^1(n) = w_0^{21}x^0(n) + w_1^{21}x^0(n+1) + \dots + w_{N_f-1}^{21}x^0(n+N_f-1) + w_0^{22}x^1(n) + w_1^{22}x^1(n+1) + \dots + w_{N_f-1}^{22}x^1(n+N_f-1) \quad (9)$$

Furthermore, $i^0(n)$ can be represented as $i^0(n) = i^{00}(n) + i^{01}(n)$ where

$$i^{00}(n) = w_0^{11}x^0(n) + w_1^{11}x^0(n+1) + \dots + w_{N_f-1}^{11}x^0(n+N_f-1) \quad (10)$$

and

$$i^{01}(n) = w_0^{12}x^1(n) + w_1^{12}x^1(n+1) + \dots + w_{N_f-1}^{12}x^1(n+N_f-1) \quad (11)$$

Similarly, $i^1(n)$ can be represented as

$$i^1(n) = i^{10}(n) + i^{11}(n) \text{ where}$$

$$i^{10}(n) = w_0^{21}x^0(n) + w_1^{21}x^0(n+1) + \dots + w_{N_f-1}^{21}x^0(n+N_f-1) \quad (12)$$

and

$$i^{11}(n) = w_0^{22}x^1(n) + w_1^{22}x^1(n+1) + \dots + w_{N_f-1}^{22}x^1(n+N_f-1) \quad (13)$$

The same analysis technique can be used on Eq. (7), the feedback part. Eqs. (10)~(13) clearly show that a direct implementation of a MIMO DFE actually has 4-fold complexity of its SISO counterpart.

3.3 Low complexity MIMO DFE

To reduce complexity, RCFIR can be applied. As mentioned earlier, to have a 100% HUE for a 2-parallel processor, one has to multiplex two way data. Hence we multiplex Eq. (10) with Eq. (12) and Eq. (11) with Eq. (13) that we achieve a 25% saving on the complexity over the direct implementation. Figure 4 shows this low complexity MIMO DFE implementation on one of the antennas where N_f is assumed to be 4. The same structure should be used on all other antennas. As can be seen in Figure 4, the 2-parallel architecture processes two data in one clock cycle and therefore the input data should be held for 2 clock cycles while the switches operate at the clock rate. Figure 5 is the block diagram of the complete low complexity MIMO DFE implementation. If N_f is an odd number, zero padding is used to make the filter length even. Additional savings in complexity can be achieved as the size of the system increases. For example, if the system is 3 x 3, the reduced-complexity 3-parallel FIR can be applied and the saving can be up to 33%, [8].

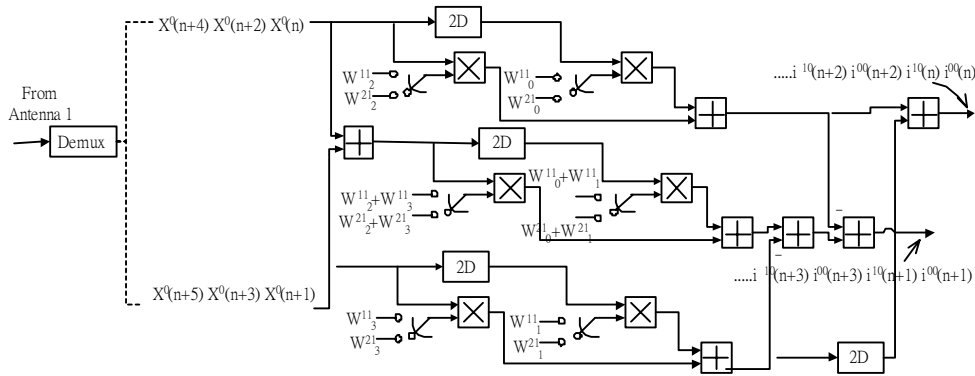


Figure 4 A Reduced Complexity 2-parallel FIR filter

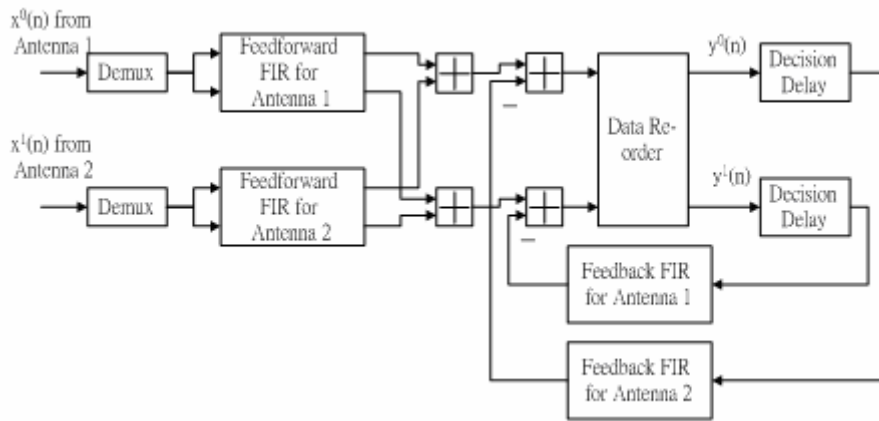


Figure 5 Block diagram of a low-complexity MIMO DFE

The saving is due to the reduction of the use of the multipliers at the cost of switches and one simple data re-order block whose functionality is explained in Figure 6. As the number of taps in the FFF and FBF increase, the saving increases too. In Sec 4, we will show this trend by a real design example implemented in Xilinx FPGA.

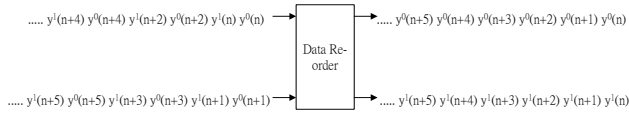


Figure 6 Data re-order block

4 A 2 x 2 Design Example

4.1 Simulation parameters

Three design cases, listed in Table 2, were considered in the simulations to show the complexity reduction.

Table 2 Design Cases

| Parameter | Case 1 | Case 2 | Case 3 |
|--------------------|--------|--------|--------|
| Number of FFF taps | 10 | 20 | 30 |
| Decision Delay | 9 | 19 | 29 |
| Number of FBF taps | 8 | 20 | 30 |

4.2 Xilinx Virtex-4 implementation results

For both cases, we implemented two designs, direct implementation and low-complexity. For illustration purpose, we also include the simulation results of their SISO counterparts. The results from Xilinx Map Report are shown in Table 3. For case 1, the slices used for both direct and low-complexity implementations are about the same. However, as the number of the taps increases, the low-complexity implementation consumes a lot less area than direct one. For case 2, the actual slices saving for low-complexity implementation is about 18.5% over the direct one. When the number of the taps increases to 30, as in case 3, the saving increases to 20% too.

5 Conclusions

In this paper, we have presented the low complexity MIMO DFE that uses the RCFIR technique. The results of actual implementations in Xilinx FPGA were reported in Table 3. One of the design cases provided a 20% saving in slices which is close to theoretical upper bound of 25%. This low complexity technique works even better as the system dimension grows larger. For example, if the system dimension is 3×3 , the maximum saving can be up to 33%. For a 4×4

system, a maximum of 43.75% chip area saving can be achieved.

Table 3 Xilinx Virtex-4 Implementation Results

| System | Slices | FFs | LUTs |
|-----------------------|--------|-------|--------|
| SISO for case 1 | 4165 | 7048 | 7740 |
| Direct case 1 | 16596 | 28111 | 30835 |
| Low complexity case 1 | 16325 | 27784 | 29602 |
| SISO for case 2 | 9520 | 16170 | 17645 |
| Direct case 2 | 38028 | 64599 | 70483 |
| Low complexity case 2 | 30991 | 53086 | 58026 |
| SISO for case 3 | 14296 | 24292 | 26477 |
| Direct case 3 | 57100 | 97023 | 105745 |
| Low complexity case 3 | 45673 | 78420 | 86450 |

References

- [1] N. Al-Dhahir and A. H. Sayed, "The finite-length multi-input multi-output MMSE-DFE", *IEEE Trans. On Signal Processing*, Vol. 48, No. 10, pp. 2921-2936, Oct. 2000.
- [2] N. Al-Dhahir and A. H. Sayed, "A computationally-efficient FIR MMSE-DFE for multiuser communications", *Asilomar Conference on Signal, System and Computer*, Pacific Grove, CA, Vol. 2, pp. 1867-1871, Oct. 2000.
- [3] A. Chevreuil and L. Vendendorpe, "MIMO MMSE-DFE: a general framewrok", *IEEE SP workshop on Statistical Signal and Array Processing*, pp. 368-371, Sept. 1998.
- [4] A. Lozano and C. Papadias, "Space-time receiver for wideband BLAST in rich-scattering wireless channels", *Proc. IEEE VTC*, Tokyo, Japan, Vol. 1, pp. 186-190, May 2000.
- [5] A. Lozano and C. Papadias, "Layer space-time receivers for frequency-selective wireless channels", *IEEE Trans. On Comm.*, Vol. 50, No. 1, pp. 65-73, Jan. 2002.
- [6] X. Zhu and R. D. Murch, "MIMO-DFE based BLAST over frequency selective channels", *IEEE GLOBECOM '01*, San Antonio, TX, Vol.1, pp. 499-503, Nov. 2001.
- [7] J. G. Proakis, *Digital Communications, 4th ed.* McGraw-Hill, New York, 2001.
- [8] K. K. Parhi, *VLSI Digital Signal Processing Systems*, John Wiley & Sons New York, 1999.