

# Design and Architecture of Spatial Multiplexing MIMO Decoders for FPGAs

Chris Dick, Xilinx, San Jose, CA, USA; chris.dick@xilinx.com;  
 Kiarash Amiri, Rice University, Houston, TX, USA; kaaa@rice.edu;  
 Joseph R. Cavallaro, Rice University, Houston, TX, USA; cavallar@rice.edu;  
 Raghu Rao, Xilinx, Austin, TX, USA; raghu.rao@xilinx.com

**Abstract**—Spatial multiplexing multiple-input-multiple-output (MIMO) communication systems have recently drawn significant attention as a means to achieve tremendous gains in wireless system capacity and link reliability. The optimal hard decision detection for MIMO wireless systems is the maximum likelihood (ML) detector. ML detection is attractive due to its superior performance (in terms of BER). However, direct implementation grows exponentially with the number of antennas and the modulation scheme, making its ASIC or FPGA implementation infeasible for all but low-density modulation schemes using a small number of antennas. Sphere decoding (SD) solves the ML detection problem in a computationally efficient manner. However, even with this complexity reduction, real-time implementation on a DSP processor is generally not feasible and high-performance parallel computing platforms such as FPGAs are increasingly being employed for this class of applications. The sphere detection problem affords many opportunities for algorithm and micro-architecture optimizations and tradeoffs. This paper provides an overview of techniques to simplify and minimize FPGA resource utilization of sphere detectors for high-performance low-latency systems.

## I. INTRODUCTION

Multiple-input multiple-output (MIMO) systems are known for their capability of achieving high data rates [1] and increasing the robustness to combat the fading in wireless channels. However, the complexity of the optimum detector, i.e. maximum-likelihood (ML) receiver, for MIMO systems grows exponentially with more antennas and higher modulation orders. In order to reduce this complexity, sphere detection [2], and its K-best variation, has been proposed [3], analyzed [4] and implemented [5], [6], [7], [8], [9].

MIMO solutions have become more popular during the recent years, and are becoming an option in several wireless standards. Therefore, it is crucial to study methods that further reduce the complexity of detection while maintaining high BER performance. Conventional K-best MIMO detectors typically require long delay cycles for sorting steps. For instance, for a multi-stage real-valued based K-best detector for a 16-QAM MIMO system, a bubble sorter needs more than 40 cycles if the detector parameter,  $K$ , is set to 10. This long list size introduces a large delay for the processing of the next stage.

In this paper, we present the FPGA implementation of a configurable MIMO detector that supports 4, 16, 64-QAM modulation schemes as well as a combination of 2, 3 and 4 antennas. The detector can switch between these parameters

on-the-fly. The breadth-first search employed in our realization presents a large opportunity to exploit the parallelism of the FPGA in order to achieve high data rates. Moreover, the extension of the detector to soft detection and its architecture implications are discussed.

The paper is organized as follows: Section II introduces the system model, section III introduces the MIMO detector. The FPGA design and implementation are discussed in section IV, and the extension to soft detection/decoding is presented in section V. Finally, the paper is concluded with section VI.

## II. SYSTEM MODEL

We consider a MIMO system with  $M_T$  transmit and  $M_R$  receive antennas. The input-output model is captured by

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\tilde{\mathbf{s}} + \tilde{\mathbf{n}} \quad (1)$$

where  $\tilde{\mathbf{H}}$  is the complex-valued  $M_R \times M_T$  channel matrix,  $\tilde{\mathbf{s}} = [\tilde{s}_1, \tilde{s}_2, \dots, \tilde{s}_{M_T}]^T$  is the  $M_T$ -dimensional transmitted vector whose elements are chosen from a complex-valued constellation  $\Omega$  of the order  $w = |\Omega|$ ,  $\tilde{\mathbf{n}}$  is the circularly symmetric complex additive white Gaussian noise vector of size  $M_R$  and  $\tilde{\mathbf{y}} = [\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_{M_R}]^T$  is the  $M_R$ -element received vector. Each modulation constellation point corresponds to  $M_c = \log w$  bits. The preceding MIMO equation can be decomposed into real-valued numbers as follows [8]:

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n} \quad (2)$$

corresponding to

$$\begin{pmatrix} \Re(\tilde{\mathbf{y}}) \\ \Im(\tilde{\mathbf{y}}) \end{pmatrix} = \begin{pmatrix} \Re(\tilde{\mathbf{H}}) & -\Im(\tilde{\mathbf{H}}) \\ \Im(\tilde{\mathbf{H}}) & \Re(\tilde{\mathbf{H}}) \end{pmatrix} \begin{pmatrix} \Re(\tilde{\mathbf{s}}) \\ \Im(\tilde{\mathbf{s}}) \end{pmatrix} + \begin{pmatrix} \Re(\tilde{\mathbf{n}}) \\ \Im(\tilde{\mathbf{n}}) \end{pmatrix} \quad (3)$$

with  $M = 2 \cdot M_T$  and  $N = 2 \cdot M_R$  presenting the dimensions of the new system.

We call the ordering in (2), the conventional ordering. Using the conventional ordering, all the computations can be performed using only real values. Note that after real-valued decomposition, each  $s_i$  in  $\mathbf{s}$  is chosen from a set of real numbers,  $\Omega'$ , with  $w' = \sqrt{w}$  elements.

## III. MIMO DETECTION

The optimum detector for such a system is the maximum-likelihood (ML) detector. ML is essentially based on minimizing  $\|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2$  over all the possible combinations of the

s vector. The ML detection requires exhaustive exponentially growing search among all the candidates, that can become practically impossible when large number of antennas are used. In order to address this challenge, the distance metric is modified [10] as follows:

$$\begin{aligned} D(\tilde{\mathbf{s}}) &= \|\mathbf{y} - \mathbf{H}\tilde{\mathbf{s}}\|^2 \\ &= \|\mathbf{Q}^H\mathbf{y} - \mathbf{R}\tilde{\mathbf{s}}\|^2 = \sum_{i=M}^1 |y_i' - \sum_{j=i}^M R_{i,j}s_j|^2 \end{aligned} \quad (4)$$

where  $\mathbf{H} = \mathbf{Q}\mathbf{R}$  represents the channel matrix QR decomposition,  $\mathbf{Q}\mathbf{Q}^H = \mathbf{I}$ ,  $\mathbf{R}$  is an upper triangular matrix, and  $\mathbf{y}' = \mathbf{Q}^H\mathbf{y}$ .

Using the notation of [5], the norm in (4) is computed in an iterative process. Starting with  $T_{M+1}(\mathbf{s}^{(M+1)}) = 0$ , the Partial Euclidean Distance (PED) at each level is given by

$$T_i(\mathbf{s}^{(i)}) = T_{i+1}(\mathbf{s}^{(i+1)}) + |e_i(\mathbf{s}^{(i)})|^2 \quad (5)$$

$$|e_i(\mathbf{s}^{(i)})|^2 = |y_i' - R_{i,i}s_i - \sum_{j=i+1}^M R_{i,j}s_j|^2 \quad (6)$$

with  $\mathbf{s}^{(i)} = [s_i, s_{i+1}, \dots, s_M]^T$ , and  $i = M, M-1, \dots, 1$ .

This iterative algorithm can be implemented as a tree traversal with each level of the tree corresponding to one  $i$  value, and each node having  $w'$  children. The tree traversal can be performed in a breadth-first manner. At each level, only the best  $K$  nodes, i.e. the  $K$  nodes with the smallest  $T_i$ , are chosen for expansion. This type of detector is generally known as the  $K$ -best detector. Note that such a detector requires sorting a list of size  $K \times w'$  to find the best  $K$  candidates. For instance, for a 64-QAM system with  $K = 16$ , this requires sorting a list of size  $K \times w' = 16 \times 8 = 128$  at most of the tree levels. This introduces a long delay for the next processing block in the detector unless a highly parallel sorter is used. Highly parallel sorters, on the other hand, consist of a large number of compare-select blocks, and result in dramatic area increase. In order to simplify the sorting step, and significantly reduce the delay of the detector, a minimum finder can replace the sorter [6], [11], [12].

The soft information, typically Log-likelihood Ratio (LLR), passed from the detection block to the decoding block is obtained by

$$L_D(x_k|\mathbf{y}) = \ln \frac{P[x_k = +1|\mathbf{y}]}{P[x_k = -1|\mathbf{y}]} \quad (7)$$

where  $k = 0, \dots, M_T \cdot M_c - 1$ . This soft information is updated in the decoder and fed back into the detector. Multiple cycles of exchanging soft information between the detector and decoder would eventually lead to more reliable soft information, which will be used by the decoder, in the last iteration, to hard-decode more reliably.

Soft information can be generated using a list of possible vector candidates. Once this list is generated, LLR values of Eq. (7) are computed and passed to the decoder [4]:

$$\begin{aligned} L_E(x_k|\mathbf{y}) &\approx \frac{1}{2} \max_{\mathbf{x} \in \mathcal{L} \cap \mathbf{X}_{k,+1}} \left\{ -\frac{1}{\sigma^2} \|\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\tilde{\mathbf{s}}\|^2 + \mathbf{x}_{[k]}^T \cdot \mathbf{L}_{A,[k]} \right\} \\ &\quad - \frac{1}{2} \max_{\mathbf{x} \in \mathcal{L} \cap \mathbf{X}_{k,-1}} \left\{ -\frac{1}{\sigma^2} \|\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\tilde{\mathbf{s}}\|^2 + \mathbf{x}_{[k]}^T \cdot \mathbf{L}_{A,[k]} \right\} \end{aligned} \quad (8)$$

where  $\mathcal{L}$  is the list of possible vectors,  $\mathbf{x}_{[k]}$  is the sub-vector of  $\mathbf{x}$  obtained by omitting the  $k$ -th bit  $x_k$ ,  $\mathbf{L}_{A,[k]}$  is the vector of all *a priori* probabilities  $L_A$  for transmitted vector  $\mathbf{x}$  obtained by omitting  $L_A(x_k)$ ,  $\sigma^2$  is the noise variance,  $\mathbf{X}_{k,+1}$  is the set of  $2^{M_T \cdot M_c - 1}$  bits of vector  $\mathbf{x}$  with  $x_k = +1$ , while  $\mathbf{X}_{k,-1}$  is similarly defined.

#### IV. FPGA DESIGN OF THE MIMO DETECTOR

The detector is designed for the maximal case, i.e.  $M_T \times M_R$ , 64-QAM case, so that it can also support a smaller number of antennas and modulation orders.

Computing the norms in (4) is performed in the PED blocks. Depending on the level of the tree, three different PED blocks are used: The PED in the first real-valued level, PED<sub>1</sub>, corresponds to the root node in the tree,  $i = M = 2M_T = 8$ . The second level consists of  $\sqrt{64} = 8$  parallel PED<sub>2</sub> blocks, which compute 8 PEDs for each of the 8 PEDs generated by PED<sub>1</sub>; thus, generating 64 PEDs for the  $i = 7$  level. Followed by this level, there are 8 parallel general PED computation blocks, PED<sub>g</sub>, which compute the closest-node PED for all 8 outputs of each of the PED<sub>2</sub>s. The next levels will also use PED<sub>g</sub>. For any incoming node, PED<sub>g</sub> computes and forwards only the best children; whereas, both PED<sub>1</sub> and PED<sub>2</sub> forward all the expanded children. At the end of the very last level, the Min\_Finder unit detects the signal by finding the minimum of the 64 distances of the appropriate level. The block diagram of this design is shown in Figure 1.

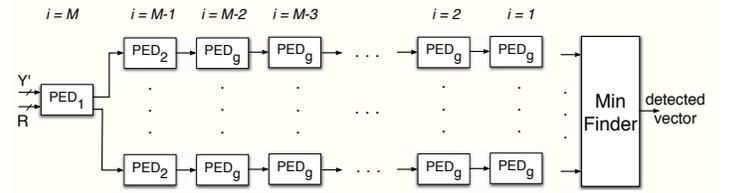


Fig. 1. The block diagram of the Flex-Sphere. Note that there are  $M$  parallel PEDs at each level. The inputs to the Min\_Finder is fed from the appropriate PED block.

The  $M_T$  determines the number of detection levels, and it is set through  $M_T$  input to the detector, which in turn, would configure the Min\_Finder appropriately. Therefore, the minimum finder can operate on the outputs of the corresponding level, and generate the minimum result. In other words, the multiplexers in each input of the Min\_Finder block, choose which one of the four streams of data should be fed into the Min\_Finder. Therefore, the inputs to final the Min\_Finder would be coming from the  $i = 5, 3$  or  $1$ , if  $M_T$  is 2, 3 or 4; respectively, see Figure 1.

The  $M_T$  input can change on-the-fly; thus, the design can shift from one mode to another mode based on the number of streams it is attempting to detect at anytime. Moreover, as will be shown later, the configurability of the minimum finder guarantees that less latency is required for detecting smaller number of streams.

In order to support different modulation orders per data stream, the Flex-Sphere uses another input control signal  $th^{(i)}$  to determine the maximum real value of the modulation order

of the  $i$ -th level. Thus,  $th^{(i)} \in \{1, 3, 7\}$ . Moreover, since the modulation order of each level is changing, a simple comparison-thresholding can not be used to find the closest candidate for Schnorr-Euchner [13] ordering. Therefore, the following conversion is used to find the closest SE candidate:

$$\tilde{s}_i = g(2[\frac{(1/R_{ii}) \cdot (y_i' - \sum_{j=i+1}^M R_{i,j}s_j) + 1}{2}] - 1) \quad (9)$$

where  $[\cdot]$  represents rounding to the nearest integer, and  $g(\cdot)$  is

$$g(x) = \begin{cases} -th^{(i)} & x \leq -th^{(i)} \\ x & -th^{(i)} \leq x \leq th^{(i)} \\ th^{(i)} & x \geq th^{(i)} \end{cases} \quad (10)$$

The above procedure is performed in  $PED_g$  to ensure selecting candidates within the proper range. In  $PED_1$  and  $PED_2$ , i.e. the first two levels, the PED of the out-of-range candidates are simply overwritten with the maximum value; thus, they will be automatically discarded during the minimum-finding procedure.

As for the real-valued decomposition, we use the modified real-valued decomposition (M-RVD) ordering of [11], [12]. In M-RVD, unlike the conventional ordering, each quadrature component is followed by the in-phase component of the same antenna. In other words, with the modified real-valued decomposition (M-RVD), every antenna is isolated from other antennas in two consecutive levels of the tree. Therefore, if we use conventional real-valued decomposition, the results for a  $2 \times 2$  system would be ready only after going through all the in-phase tree levels and the first two quadrature levels, while, using M-RVD, there is no need to go through the latency of the unnecessary levels. Thus, using the M-RVD technique offers a latency reduction compared to the conventional real-valued decomposition.

#### A. FPGA Synthesis Results

The System Generator FPGA implementation results of the MIMO detector on a Xilinx Virtex-5 FPGA, xc5vsx95t-3ff1136 for 16-bits precision and  $M_T = 4$  are presented in Table I. The maximum achievable clock frequency is 285.71 MHz. The folding factor of the design is  $F = 8$ , thus, the maximum achievable data rate is

$$D = \frac{M_T \cdot \log w}{F} \cdot f_{max} = 857.1 \text{ [Mbps]} \quad (11)$$

for  $M_T = 4$  and  $w_i = 64$ .

TABLE I  
FPGA RESOURCE UTILIZATION SUMMARY OF THE PROPOSED FLEX-SPHERE FOR THE XILINX VIRTEx-5, XC5VSX95T-2FF1136, DEVICE.

Number of Slices	11,604/14,720 (78 %)
Number of Slice Registers	27,115/58,880 (46 %)
Number of Slice LUTs	33,427/58,880 (56 %)
Number of DSP48E	321/640 (50 %)
Max. Freq.	285.71 MHz

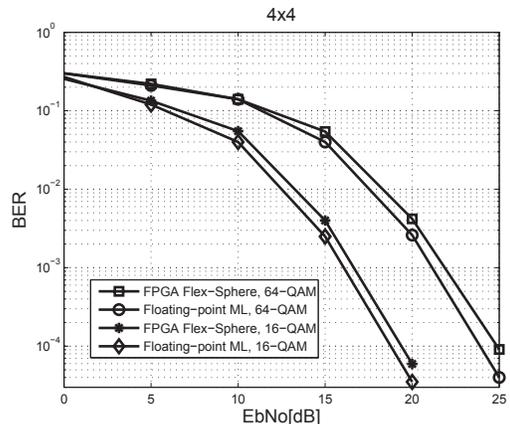


Fig. 2. BER plots comparing the performance of the floating-point maximum likelihood (ML) with the FPGA implementation.

#### B. Simulation Results

In this section, we present the simulation results for the Flex-Sphere, and compare the performance of the FPGA fixed-point implementation with that of the optimum floating-point maximum-likelihood (ML) results. Prior to the M-RVD, we employ the channel ordering of [14] to further close the gap to ML. Also, we make the assumption that all the streams are using the same modulation scheme. We assume complex-valued channel matrices, with the real and imaginary parts of each element drawn from the normal distribution.

In order to ensure that all the antennas in the receiver have similar average received SNR, and none of the users messages are suppressed with other messages, a power control scheme is employed. Figure 2 shows the simulation results for the maximal  $4 \times 4$  configuration. As can be seen, the proposed hardware architecture implementation performs within, at most, 1 dB of the optimum maximum-likelihood detection.

#### V. SOFT DETECTION/DECODING

The list of candidates generated at the last level of the MIMO detector can be used to generate soft values, i.e. LLRs, using Eq. (8). Those LLRs will be, then, used by the channel decoder to decode the information bits. Figure 3 provides a schematic representation of Eq. (8). The inputs to the computation is the length  $M_T M_c$  vector of bit-level APP probabilities computed by the outer channel decoder, a list of  $P$  candidate output vectors from the MIMO sphere detector, each bit vector is of length  $M_T M_c$  and finally a  $P$ -vector of distance metrics, or costs, for each of the  $P$  candidates in the sphere detectors output symbol list.

To determine the cost, in terms of time initially, for computing the soft outputs from the list of candidates generated by the Sphere Detector, first consider the number of clock cycles required to compute Eq. (12) for a single candidate using a sequential approach.

$$\left\{ -\frac{1}{\sigma^2} \|\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\tilde{\mathbf{s}}\|^2 + \mathbf{x}_{[k]}^T \cdot \mathbf{L}_{A,[k]} \right\} \quad (12)$$

TABLE II  
CYCLE COUNT COST  $T_{soft}$  FOR VARIOUS MIMO CONFIGURATIONS.

Modulation	$M_T$	$K$	Cycles
QPSK	2	5	3132
QPSK	2	10	6252
16-QAM	2	5	12492
16-QAM	2	10	24972
64-QAM	2	5	49932
64-QAM	2	10	99852
QPSK	4	5	12024
QPSK	4	10	24024
16-QAM	4	5	48024
16-QAM	4	10	96024
64-QAM	4	5	192024
64-QAM	4	10	384024

Since both  $\mathbf{x}_{[k]}$  and  $\mathbf{L}_{\mathbf{A},[k]}$  exclude the  $k$ 'th bit of the hard-decision bit-vectors in the list of candidates generated by the sphere detector, and further that each entry of  $\mathbf{x}_{[k]}$  takes on the values of only  $\pm 1$ , the inner product  $\mathbf{x}_{[k]} \cdot \mathbf{L}_{\mathbf{A},[k]}$  is computed in  $M_T \cdot M_c - 1$  clock cycles using only a single adder. One further addition is required to form the sum  $\|\mathbf{y} - \mathbf{H} \cdot \mathbf{s}\|^2 + \mathbf{x}_{[k]} \cdot \mathbf{L}_{\mathbf{A},[k]}$ . This component of the calculation is completed by taking a Jacobian logarithm. All of the candidates in the list need to be processed, and assuming that there are  $K \cdot |\Lambda|$  such candidates, where  $|\Lambda|$  denotes the cardinality of the constellation, results in the time required to compute the soft value for a single bit in the length  $M_T \cdot M_c$  output bit vector is  $K \cdot |\Lambda| \cdot (M_T \cdot M_c + T_{jactn})$  where  $T_{jactn}$  is the time to compute a Jacobian logarithm. The difference between the two primary terms in Eq. (8) corresponding to  $\mathbf{x} \in \mathbf{X}_{k,+1}$  and  $\mathbf{x} \in \mathbf{X}_{k,-1}$  requires one subtraction, and there are  $M_T \cdot M_c$  such calculations. Combining this cost gives the final workload  $T_1$  for computing the soft value for a single bit as  $M_T \cdot M_c$  bits

$$T_1 = K \cdot |\Lambda| \cdot (M_T \cdot M_c + T_{jactn}) + 1 \quad (13)$$

The hard decision bit vector contains  $M_T \cdot M_c$  entries, for each of which a soft value needs to be computed, giving the total time  $T_{soft}$  for computing the soft output for all of the bits as

$$T_{soft} = M_T \cdot M_c \cdot (K \cdot |\Lambda| \cdot (M_T \cdot M_c + T_{jactn}) + 1) \quad (14)$$

Scaling by the noise variance term  $-1/\sigma^2$  in Eq. (8) can be handled as a pre-processing phase to computing the soft-outputs. That is, prior to engaging the soft-output generation circuit the  $K \cdot |\Lambda|$  length list of cost metrics is scaled by  $-1/\sigma^2$ . The cost of the scaling by  $1/2$  in Eq. (8) is also not included in the calculations as this is realized in hardware as a simple bit shift that is accommodated in the circuit wiring and does not incur any compute fabric cost in an FPGA.

Table II provides a tabulation of the cost for computing soft output values, as defined by Eq. (14), for several MIMO configurations.

## VI. CONCLUSION

In this paper, we presented a configurable architecture for MIMO detection. The proposed architecture enhances the performance of MIMO systems for next generation wireless standards, and can support a wide range of different scenarios.

Moreover, the FPGA synthesis results demonstrated achieving high data rates. Finally, we presented a scalable architecture to generate soft values using the list of the candidates generated at the last level of the MIMO detector.

## VII. ACKNOWLEDGEMENT

This work was supported in part by Xilinx Inc., and by NSF under grants EIA-0321266, CCF-0541363, CNS-0551692, and CNS-0619767.

## REFERENCES

- [1] G. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multiple antennas," *Bell Labs. Tech. Journal*, vol. 2, 1996.
- [2] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. Comput.*, vol. 44, no. 170, pp. 463–471, Apr. 1985.
- [3] E. Viterbo and J. Boutros, "A universal lattice decoder for fading channels," *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1639–1642, Jul. 1999.
- [4] B. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. on Comm.*, vol. 51, pp. 389–399, Mar. 2003.
- [5] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bolcskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1577, July 2005.
- [6] L. G. Barbero and J. S. Thompson, "Performance analysis of a fixed-complexity sphere decoder in high-dimensional MIMO systems," *IEEE Conference on Acoustics, Speech and Signal Processing*, vol. 4, May 2006.
- [7] K. Amiri and J. R. Cavallaro, "FPGA implementation of dynamic threshold sphere detection for MIMO systems," *40th Asilomar Conf on Signals, Systems and Computers*, Nov 2006.
- [8] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-Best sphere decoding for MIMO detection," *IEEE JSAC*, vol. 24, no. 3, pp. 491–503, Mar. 2006.
- [9] K. Wong, C. Tsui, R. S. Cheng and W. Mow, "A VLSI architecture of a K-best lattice decoding algorithm for MIMO channels," *IEEE Int. Symp. Circuits Syst.*, vol. 3, pp. 273–276, May 2002.
- [10] M. O. Damen, H. E. Gamal and G. Caire, "On maximum likelihood detection and the search for the closest lattice point," *IEEE Trans. on Inf. Theory*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.
- [11] K. Amiri, C. Dick, R. Rao and J. R. Cavallaro, "Novel sort-free detector with modified real-valued decomposition (M-RVD) ordering in MIMO systems," *Proc. of IEEE Globecom*, Dec. 2008.
- [12] —, "Flex-Sphere: An FPGA Configurable Sort-Free Sphere Detector for Multi-user MIMO Wireless Systems," *Proc. of SDR Forum*, Oct. 2008.
- [13] C. P. Schnorr and M. Euchner, "Lattice basis reduction: improved practical algorithms and solving subset sum problems," *Math. Programming*, vol. 66, no. 2, pp. 181–191, Sep. 1994.
- [14] L. G. Barbero and J. S. Thompson, "A fixed-complexity MIMO detector based on the complex sphere decoder," *IEEE 7th Workshop on Signal Processing Advances in Wireless Communications, 2006. SPAWC '06*, Jul. 2006.

**key**

Mn: memory element n  
 Fn: functional unit n  
 MPVn: multiplier n  
 An: adder n  
 Cn: comparator n  
 Rn: register n

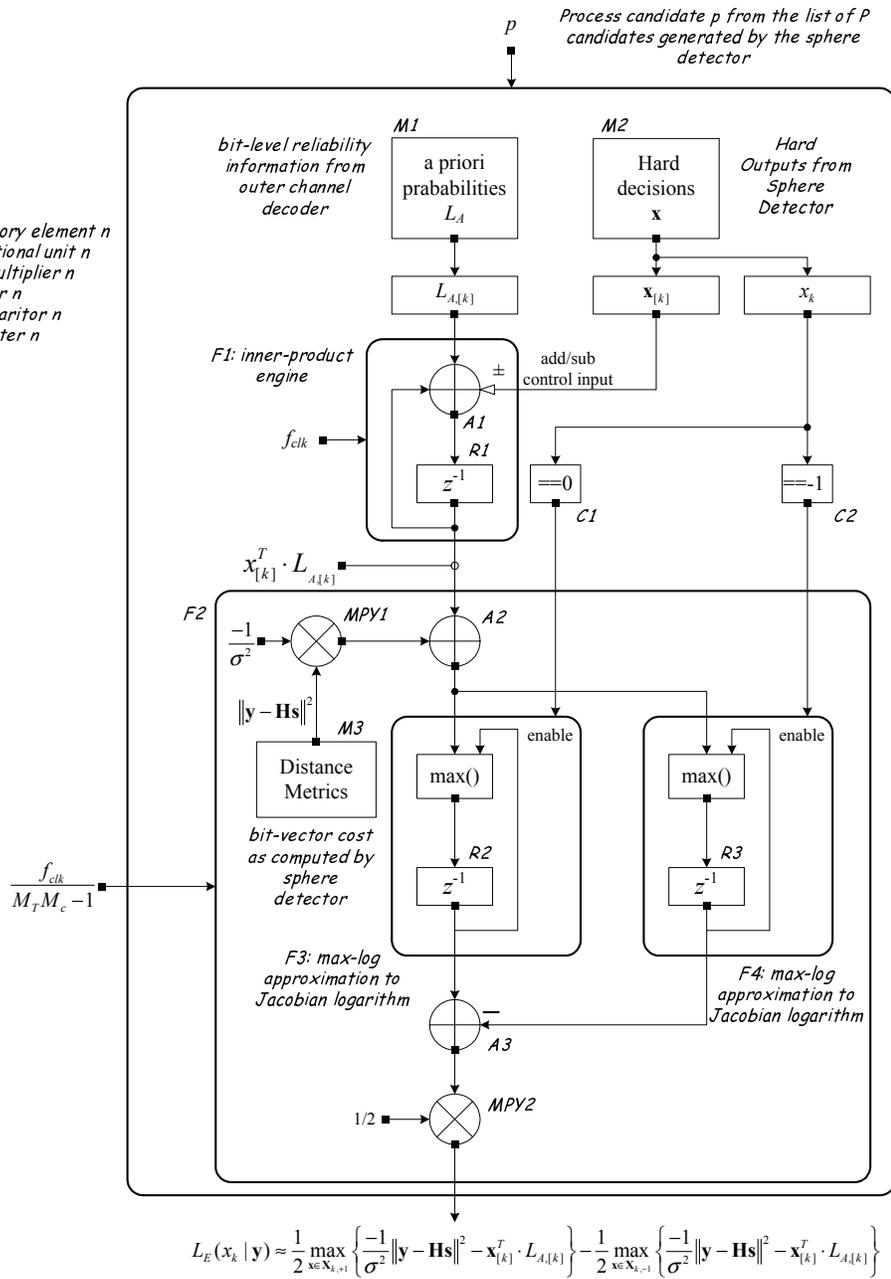


Fig. 3. Soft-output generation for sphere detector.