

Optimized Implementation of Serial Binary Input Compressive Sensing Decoder

Sowmya K R

PG student (VLSI Design and Embedded systems)
SIET, Tumkur,
Karnataka, India

Asha R

Assistant Prof., Dept. of E&C, SIET,
Tumkur, Karnataka, India

Abstract- Compressive sensing is a signal processing technique. It is also called as compressed sensing, compressive sampling or sparse sampling. The compressive sensing is depending upon the knowledge about a signal to obtain a compressed representation. Hence it takes advantage of the compressibility in some domain, that allowing the entire signal to be determined. Binary Input Compressive Sensing (BICS) decoder adopt smooth and continues rate but sometimes it produced very low output called as throughput and it also reduces HUE (Hardware Utilization Efficiency). To solve this problem there are 2 methods are used in this project. 1) Using the configurable aspect ratio of the block RAM in FPGA. This block RAM consists of NRAM. This makes the performance of the decoder in series or in pipelined because the implementation of parallel architecture and partial parallel architecture is very complicated and it is also reduces the throughput. Hence in order to improve the output of the decoder a serial architecture is proposed and it is implemented in FPGA. 2) Usually in normal operation each node are cannot be overlapped during computation and these nodes are computed one after the other, which results in very low HUE. So to avoid this problem the ping pang operation is used. This will increases the efficiency of BICS decoder. The serial architecture decoder of BICS is implemented on Xilinx Spartan 6 FPGA using verilog. When the clock frequency is 300MHz with the number of iteration is 10. Then the maximum throughput of the decoder is about greater then 23Mbps and 24Mbps with 13 clock cycle. This reaches the range of communication rate in modern wireless networks with less area utilization.

Keywords- *Compressive sensing; block RAM; seamless rate adaptation; ping pang operation; belief propagation.*

I. INTRODUCTION

Compressive sensing is a signal processing technique used for reconstructing a signal. This signal is efficiently acquiring by finding the solutions to underdetermined liner systems. The sampling and dimensionality reduction in signal is performed by compressive sensing under the assumption of sparsity. This sparsity has played very important role in modern signal processing and that utilizes the sparsity of signals. It depends on the sparsity of the signal but it is not depends on its highest frequency. Sparse signal means "The signal is settled at which spaced intervals." If the sparse signal consists of high frequency components then it can be highly under sampled. If the signal's highest frequency is less than half of the sampling rate then the "Nyquist Shannon" sampling thermo is proved that a signal can be perfectly reconstructed from sampling.

To obtain a compressed representation of the signal the compressive signal uses complete knowledge about a signal by adopting and acquiring a small number of nonadaptive linear measurements of the sparse signal. During the transfer of image or video some data will be missed to identify that missing data the compressive sensing are used by Shannon Nyquist theorem. This theorem states that "It is a fundamental bridge between continues signals means analog domain and discrete signals means digital domain.

The compressive sensing can also be used in orbit around the earth. The Herschel satellite, recently launched by European space agency. It is faced with the following problem. According to Bobin, large amounts of data that Herschel collects must be transmitted to earth but the transmission cost is too high. Since a small on board CPU load is required but traditional compressive techniques is cannot be used. Hence most of the computational load in compressed sensing is at the decoder side, by collecting compressed sensing measurements and it can be reduced the transmitted data volume Compressed sensing is a compressible / k-sparse signal. k- Sparse signal means f is k-sparse if only k coefficients s_i are nonzero. Compressible signals are well approximated by k- sparse representation.

Many natural and man-made signals are compressible. The important facts of compressed sensing are:

- The number of required samples or measurements are liner in sparsity and logarithmic in signal space dimension.
- It is a robust with respect to noise.
- It works for approximate sparse signals.
- In compressed sensing highly efficient reconstruction algorithms are available.
- Some signal models are supported by the CS. e.g: Group sparsity model and joint sparsity model.

First built a measurement matrix because to evaluate the performance of the decoder. Then the compressive sensing is converted all algorithms in to binary in BICS decoder. The measurement processes must not erase the information present in the k nonzero entries of signal f. BICS (Binary Input Compressive Sensing) decoder which adapts or it providing smooth and continues data rate to the input. The rate adaptation is critical to the system performance because most

existing rate adaptation schemes adapt sender rate adaptation, which is also known as adaptive modulation and coding (AMC). First the receiver observes channel condition and delivers a feedback to the sender for rate selection i.e for selecting the channel code rate and modulation constellation which means “It is process of varying one or more properties of periodic waveforms digitally”.

To solve this problem, Cui proposed Seamless Rate Adaptation (SRA) based on Rate Compatible Modulation (RCM). Rate adaptation is achieved by varying the number of modulated symbols. But it is not sensitive to the channel estimation. These modulated symbols are largely generated from a block of source bits through weighted mapping. SRA demonstrates an important application of the CS (Compressive Sensing) theory in wireless communications. The application of CS in SRA has two features.

1. The input signals are binary, known as Binary Input Compressive Sensing (BICS) decoder.
2. The measurement matrix is sparse means it settled at widely spaced intervals with eight nonzero elements in each row.

II. PROPOSED SYSTEM

Fig 3.1 shows the block diagram of the binary input compressive sensing decoder. This architecture has the following blocks as shown in the fig:

- SNFU (Symbol Node Functional Unit)
 - Subset decode
 - Path in
- BNFU (Bit Node Functional Unit)
 - Metric
 - Compute metric
- RAM (Random Accesses Memory)
 - NRAM

This BICS decoder architecture consists of one SNFU and one BNFU for each node processing. SNFU means Symbol Node Functional Unit consists of subset decode and path in blocks. BNFU means Bit Node Functional Unit consists of metric and computes metric blocks. In this architecture first the Non volatile RAM (NRAM) selects the module either it select SNFU or BNFU because this unit is always computed one after the other. The NRAM consists of SNRAM and BNRAM. SNRAM means Symbol Node RAM and BNRAM means Bit Node RAM. In the first step, if the NRAM selects SNFU then the BNRAM send bit-to-symbol messages to the SNFU and received symbol register send received symbols to the SNFU. Then it calculates symbol-to-bit messages and that result will be written in to the SNRAM. Similarly in the next step the reduced matrices get received symbols and this matrix reduces the number of iteration and the clock cycle. Then SNRAM send symbol-to-bit messages to the BNFU then it calculates bit-to-symbol messages and that result will be written in to the BNRAM. This process will be continued.

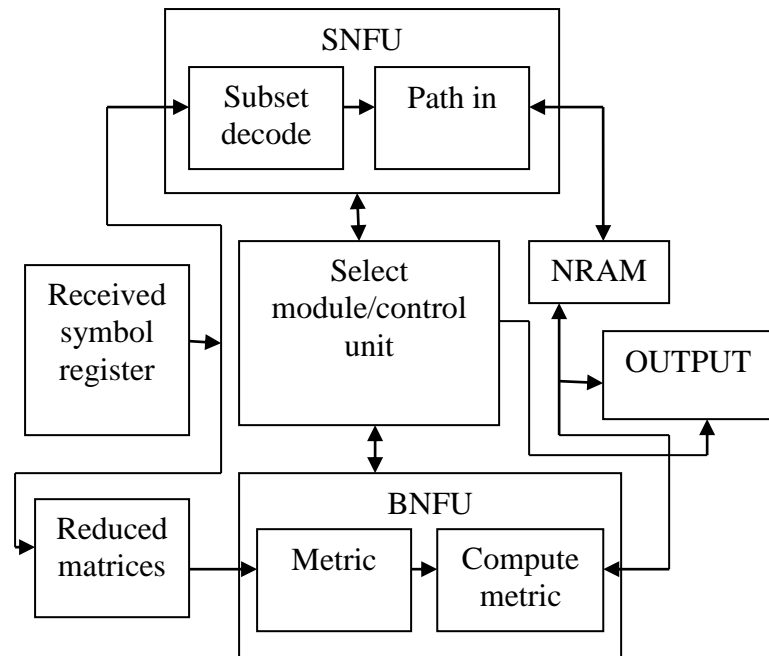


Fig 1: Serial architecture of BICS decoder

Here the select module or control unit is situated between SNFU and BNFU. The NRAM is initialized by using this control unit and also number of iteration process is controlled by this unit. The other part of the BNFU is connected to the output and by making a hard decision it will produces the output result after certain number of iterations. The BICS which adopt smooth and continues data rate to the input but BNFU exhibit very low throughput because of this reason it reduces HUE. To solve this problem there are 2 methods are exists. First, using the configurable aspect ratio of the NRAM in FPGA then the decoder can perform in series. Second, using the ping pang operation the Hardware Utilization Efficiency (HUE) is greatly increased.

If both SNFU and BNFU can work in series, then all computation in each node processing can be executed with in the single clock cycle. The throughput of the decoder can be calculated by using this formula $fclk \cdot N / ((M+N) \cdot T)$. fclk - Clock frequency, M - Symbol node, N - Bit node. By applying the clock frequency with the number of iteration the symbol nodes are denoted by M are used to decode bit nodes denoted by N.

III. DESCRIPTION

1. Serial implementation for BICS

Here the main intention is to improve the throughput of the decoder and HUE (Hard Utilization Efficiency). In order to improve the performance of the decoder it needs measurement matrix. It can evaluate the performance of the BICS decoder. This measurement matrix is in the size of 8192x4096. 8192 which indicates the number of rows and 4096 indicate the number of columns. But it is too complicated to be implemented hence in this project measurement matrix is generated instead of implementation. In measurement matrix it must be consists of 8 nonzero

elements in each rows and that should be in the ascending order. These 8 nonzero elements can be located in the different position in the each row. Ex: (-4,-4,-3,-2,+2,+3,+4,+4). So the each column weight is 1 in every 512 rows, then the total column weight of this matrix is about 16.

The construction of the measurement matrix is used to create the symbols by taking a part of rows in the matrix, Whose number is varied from 1536 to 8192 and 10 decoder iterations were sufficient for good decoding performance. The serial implementation for BICS consists of following blocks

- SNFU Architecture
- BNFU Architecture
- Implementation of RAM

A. SNFU Architecture

SNFU means Symbol Node Functional Unit. It calculates the 8 symbol-to-bit messages for the 8 bit nodes. That should be connected to the symbol node as shown in the fig 2. Here the SNFU calculates bit node by using symbol node. This 8 bit symbol node consists of 8 nonzero entries that should be in the ascending order. The probability distribution from the bit nodes are assigned to P [0] to P [7] with the weight value. Ex:{-5,-4,-2,-1,+1,+2,+4,+5}. But each probability distribution vector has only two nonzero elements. It reduces the number of multiplications of the convolution of two vectors.

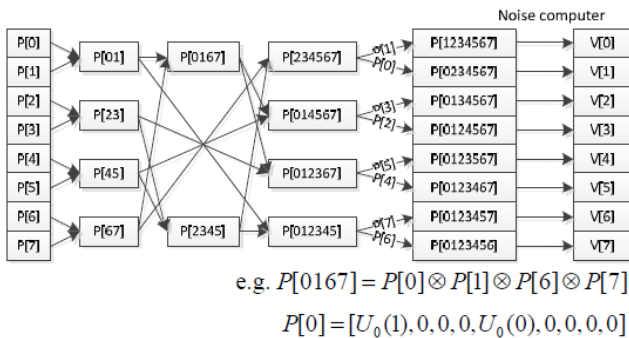


Fig 2: SNFU Architecture

As shown in this figure, only four multiplications and one addition are needed in the calculation of the convolution of the vector P [0] and P [1] the length of the vectors is about is 9. In case of addition, the intermediate results can be used repeatedly for next step calculations during the convolution computation. It saves the large number of calculation. Then the total number of multiplication in SNFU can be reduced to 750. In case of multiplication some entries will be missed. By assigning that missing values to the output the bit nodes will be produced with noise. These noises will arise by the noise computer and used to control the number of iteration process and multiplication.

B. BNFU Architecture

The figure 3 shows BNFU architecture. It calculates 8 bit-to-symbol messages for the 8 symbol nodes. This symbol

node is connected to the bit node. The BNFU consists of 16 inputs namely V [0]...V [15]. Here the BNFU perform 8 bit multiplication. That will explain with figure 3. In this figure V [0] to V [15] is taken as input. Here it needs to perform 8 bit multiplication by multiplying 2 inputs respectively as shown below. After the 8 bit multiplication result will be assigned to W0...W7. After the BNFU perform 16 bit multiplication by multiplying 2 values as shown in the figure then, that multiplied value will be stored on Y0, Y1, Y2, and Y3. But here the BNFU needs only one output during multiplication. Hence by performing 32 bit multiplication the values produced from this multiplication are assigned to Z0, Z1. After 64 bit multiplication is performed then the value produced from this multiplication will be assigned to X.

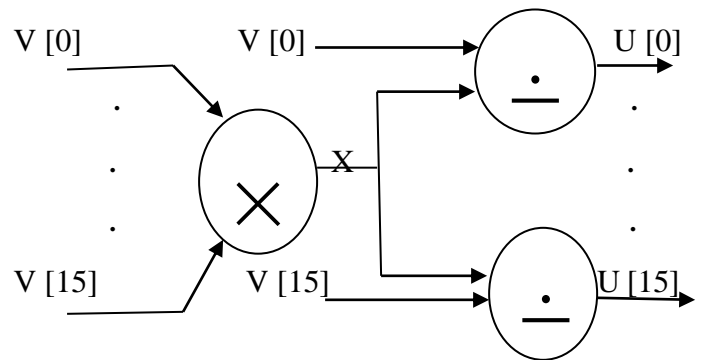


Fig 3: BNFU Architecture

In second stage the BNFU needs to be performing division by dividing the each input values of 1st stage with output of 1st stage i.e X value as shown in figure 3.4. The BNFU produced output value from multiplication and division. The output is taken as U [0].....U [15] as shown in the figure.

C. Implementation of RAM

The BICS which adopt smooth and continues rate but sometimes the BNFU exhibit very low throughput and it also reduces HUE. To solve this problem there are 2 methods are present. First, using the configurable aspect ratio of the block RAM in FPGA so it makes the decoder can perform in series. This RAM is usually situated in the control unit. Second, using the ping pang operation the Hardware Utilization Efficiency (HUE) is greatly increased.

During implementation of the RAM both the data width and depth of the memory can be selected differently for input port and output port when block RAM is taken as dual port RAM The implementation of RAM is as shown in the figure 4. Usually the block RAM is configured as dual port RAM. This block RAM consists of both SNRAM and BNRAM. This both RAMs consists 16 dual port sub RAMs and each sub RAMs occupy 8 18kb block RAMs. In block RAM the input port of each sub BNRAM is configured as 8K*18 and output port of each sub BNRAM is configured as 1K*144. Similarly in SNRAM the input port of each sub SNRAM is configured as 8K*18 and output port of each sub

SNRAM is configured as 1K*144 respectively. SNFU receives Bit-to-Symbol messages from BNRAM which means it reads the input messages of 1-512 rows of the matrix from BNRAM_1 and 512-1024 rows from BNRAM_2. It reads the input messages up to 7680-8192 rows from BNRAM_16. Each word is 144 bit. This represents 8 Bit-to-Symbol messages. When processing of SNFU is over it calculates Symbol-to-Bit messages and that result will be written on SNRAM. At the same time the SNRAM_1 is used to store the messages of 1-512 rows, and SNRAM_2 is used to store the messages of 513-1024 rows. The BNFU stored 16 input messages in 16 sub RAM and with in the single clock cycle it can be read the input messages. Similarly with in the single clock cycle the BNFU stored 16 outgoing messages and these messages can be written to 16 sub BNRAMs.

18.....18

18.....18

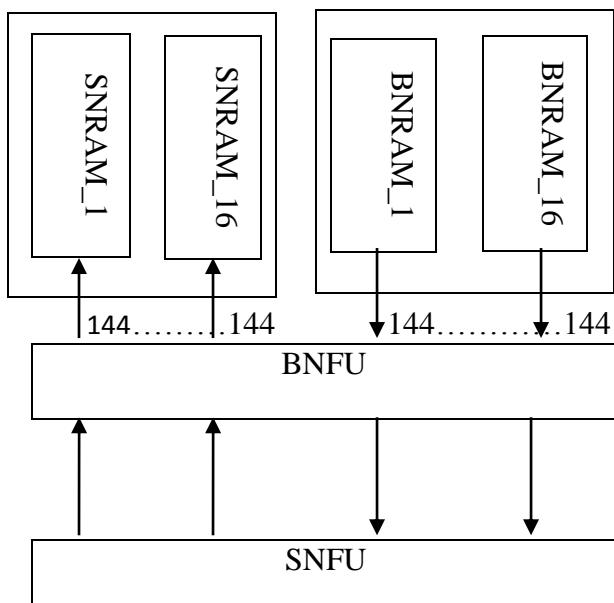


Fig 4: Implementation of RAM

This sub RAM mainly depends on the entry of each column for access the address of each RAM and this address will be increased by 1 for the next symbol node updating.

2. Ping pang operation

This operation can be used to improve the HUE. Usually the functional units like SNFU (Symbol Node Functional Unit) and BNFU (Bit Node Functional Unit) cannot be overlapped during each computation. Hence the both functional units are computed one after the other. Because this unit requires the updated messages from each other. This produces a very low HUE. To improve this efficiency the ping pang operation is used.

In this operation the received symbols are prepared two blocks for the decoder in which the symbol-to-bit messages are updating from SNFU for the one block symbols and bit-to-symbol messages are updating from BNFU for the other block symbols. This operation is done due to two reasons.

- 1) Simultaneously the read and write operations can be performed by SNFU and BNFU when all the RAMs are arranged as the dual port RAMs in a particular way.
- 2) Each sub RAMs is utilized about 1/2 because only 512 node messages are stored during normal operation. Hence the no more block RAMs are needed by this operation. The front half of the RAMs are used by one block symbols and remaining half of the RAMs are used by another block symbols.

By using this ping pang operation the HUE is largely increased. The HUE can be calculated by using this formula: $fclk * 2N / (\max(M, N) * (T-1) + M + N)$.

IV. BICS ENCODING AND DECODING

A. Matrix representation of BICS

Let $X=(X_1, X_2, \dots, X_N)$ be a block of input of length N. The M symbols $S=(S_1, S_2, \dots, S_N)$ are generated by random mapping.

$$S = WX^T \tag{1}$$

Where W is a sparse random measurement matrix with dimension M*N consists of only L nonzero entries in each row.

B. The BP decoding algorithm for BICS.

The standard message passing algorithm for BICS decoding is described as follows.

1) Initialization

The message sent by x_j to s_i is initialized as below. Where P_0 is the prior probability of $x_j=0$.

$$\{U_{ji}^{(0)}(0) = P(X_j=0) = p_0 \tag{2}$$

$$\{U_{ji}^{(1)}(1) = P(X_j=1) = 1 - p_0 \tag{3}$$

2) Symbol nodes processing

$V_{ij}^{(t)}()$ is computed by symbol node functional unit (SNFU). Then

$$\{V_{ij}^{(t)}(0) = P(X_j=0|r_i) = P(X_{ij}=r) \tag{4}$$

$$\{V_{ij}^{(t)}(1) = P(X_j=1|r_i) = P(X_{ij}=r-w_l) \tag{5}$$

Because of the weighted sum the distribution of X_{ij} is the convolution of the probability distribution function (PDF) of every bit nodes in $R_{i/j}$ and channel noise e_i , namely

$$P(X_{ij}) = \{m \in R_{i/j}, l_m \neq 1\} \otimes P(e_i) \tag{6}$$

Where \otimes is the convolution of PDFs. The distribution of weighted bit node should be

$$\{P(W_{lm} X_m=0)=U_{mi}^{(t-1)}\} \quad (7)$$

$$\{P(W_{lm} X_m=W_{lm})=U_{mi}^{(t-1)}\} \quad (8)$$

3) **Bit nodes processing**

$U_{ij}^{(t)}$ is computed by bit node functional unit

(BNFU). The message sent by X_j to S_i in t -th iteration is

$$\{U_{ji}^{(t)}(0)=K_{ji} P_0 \prod_{m \in C_j \setminus i} V_{mj}^{(t)}(0)\} \quad (9)$$

$$\{U_{ji}^{(t)}(1)=K_{ji}(1-P_0) \prod_{m \in C_j \setminus i} v_{mj}^{(t)}(1)\} \quad (10)$$

Where K_{ij} is a normalization constant which ensures

$$U_{ji}^{(t)}(0)+U_{ji}^{(t)}(1)=1 \quad (11)$$

4) **Decision and output**

After T iteration, the belief propagation process stops and makes a hard decision on values of each source bit X_j by

$$X_j = \begin{cases} 0, & U_j^{(T)}(0) \geq U_j^{(T)}(1) \\ 1, & \text{otherwise} \end{cases} \quad (12)$$

V. FPGA IMPLEMENTATION RESULTS

1. The number of symbols is 1536 with the 0.073 (bps) clock frequency and the ping pang operation efficiency is 0.0982 (bps) then the improved HUE is about 35%.
2. The number of symbols is 2048 with the 0.067 (bps) clock frequency and the ping pang operation efficiency is 0.0976 (bps) then the improved HUE is about 46%.
3. The number of symbols is 8192 with the 0.033 (bps) clock frequency and the ping pang operation efficiency is 0.0488 (bps) then the improved HUE is about greater than 90%.

By comparing the throughput of the ping pang operation with normal operation it improved almost 90% and the hardware utilization efficiency is about nearly 100% with the number of iteration is 10.

VI. CONCLUSION

A serial architecture decoder of BICS is proposed and implemented in Xilinx Spartan 6 FPGA using verilog. This decoder improves the throughput and HUE by taking advantage of configurable aspect ratio of the block RAM that is NRAM and the ping pang operation. By varying the clock frequency is about 300MHz with the number of iteration is 10 and then the decoder supports a maximum throughput is 23Mbps, which satisfies the requirements of modern wireless networks.

REFERENCES

- [1]. Fang Lu, Wengui Rao, Yan Dong 3rd MECO-2014 “FPGA-Based Implementation of Binary Input Compressive Sensing Decoder”.
- [1]. D. Donoho, vol.52, pp.1289-1306, Apr 2006 “Compressed Sensing”.
- [2]. Dror Baron, Shriram Saravotham and Richard G Baraniuk, vol 58, no.1, pp. 269-280, Jan 2010 “Bayesian Compressive Sensing via Belief Propagation”.
- [3]. M.Wang, J.Wu, S.F.Shi, C.Luo and F.Wu vol.2, no.3, sep 2012. “Fast Decoding and Hardware Design for Binary-Input Compressive Sensing”.
- [4]. H. Cui, C. Luo, K. Tan, F. Wu and C. W. Chen “Seamless Rate Adaptation for Wireless Networking” in proc. 14th ACM Int Conf. Model., Anal, simulate. Wireless mobile syst. (MSWiM ‘11), New York, 2011. Pp. 437-446.
- [5]. Rayan saab , B.E., American university of Beirut 2003 and M. A. Sc., The university of British Columbia, 2005 “Compressed Sensing: Decoding and Quantization”.
- [6]. IP, Jose Bioucas Dias, IST, 2007 “Compressed Sensing”.
- [7]. Eidgen ossische Technische Hochschule, Z Urich and Helmut Bolcskei proposed a masters thesis in Aug 2008, “Compressive Sensing”.
- [8]. Alexander Jung, Georg Taubock, Noebert Gortz “Compressed Sensing Fundamentals” in 2006.