Journal of Engineering Sciences
Vol 12, Issue 06, JUNE /2021
ISSN NO:0377-9254

CHIP DESIGN FOR TURBO NETWORKS ON CHIP MODULE USING parallel AND serial METHODS

G. MALLESHAM[1],Dr. S. IBRAHIM SADHAR[2]

1. M.Tech Scholar, VLSI SD, Department of Electronics and Communication Engineering.
2. Professor, Department of Electronics and Communication Engineering, J.B. Institute of Engineering and Technology, Hyderabad, Telangana.

Abstract—This paper studies design and implementation of the Turbo encoder to be an embedded module in the In-Vehicle System (IVS) chip. Field Programmable Gate Array (FPGA) is employed to develop the Turbo encoder module. As a high performance on-chip communication method, the Turbo technique has recently been applied to Networks on Chips (NoCs). We propose a new serial and parallel based encoding/decoding methods to leverage the performance and cost of TurboNoCs in area, power assumption, and network throughput. In the transmitter module, source data from different venders are separately encoded with an orthogonal code of a standard basis and these coded data are mixed together by an XOR operation. Then, the sums of data can be transmitted to their destinations through the onchip communication infrastructure. In the receiver module, a sequence of chips is retrieved by taking an AND operation between the sums of data and the corresponding orthogonal code. After a simple accumulation of these chips, original data can be reconstructed. We implement our encoding/decoding method and apply it to a TurboNoC with a star topology.

Index Term: Turbo encoder module; field programmable gate array; emergency call; in-vehicle system chip.

I. INTRODUCTION

The European emergency Call (eCall) system is a telematics system designed to save more lives in vehicle accidents. It is a governmental mandatory system that is to be implemented by March 2018 [1][2]. The EU eCall system provides an immediate voice and data channel between the vehicles and an emergency center after car accidents. The data channel provides the emergency center with the necessary data for emergency aids. The EU eCall system main parts includes the in-vehicle system (IVS), the Public Safety Answering Point (PSAP), a cellular communication channel. The IVS activates the data channel automatically when a car accident occurs. The IVS collects the minimum set of data (MSD) that includes GPS coordinates, the VIN number, and all required data for an emergency aid. It sends the MSD to the closest PSAP through a cellular

channel in up to 4 seconds [1]. The PSAP sends the emergency team to the location of the accidents. The IVS modem employs multiple modules for the MSD signal processing. The modules of the IVS are shown in Figure 1. The IVS employs a Turbo encoder as a Forward Error Correcting (FEC) [1]. The Turbo encoder implements the digital data encoding technique in data transmissions. Turbo coding is one of the most popular and efficient coding technique to improve Bit Error Rate (BER) in digital communications [3] [4]. The Cyclic Redundancy Check (CRC) [5], the modulator[6], the demodulator-decoder [7] modules are projected and implemented on an FPGA device. They are developed to be embedded modules of the IVS chip.

This work studies the hardware development of the Turbo encoder. It employs FPGA technologies to develop the Turbo encoder to be an embedded module in the IVS modem. It discusses serial and parallel computation techniques for the Turbo encoder. It does not only design and implement the Turbo encoder module, but also proposes a better solution for the turbo encoder implementation. The improvement of the chip size and processing time are exhibited by developing the parallel computation technique for the Turbo encoder.

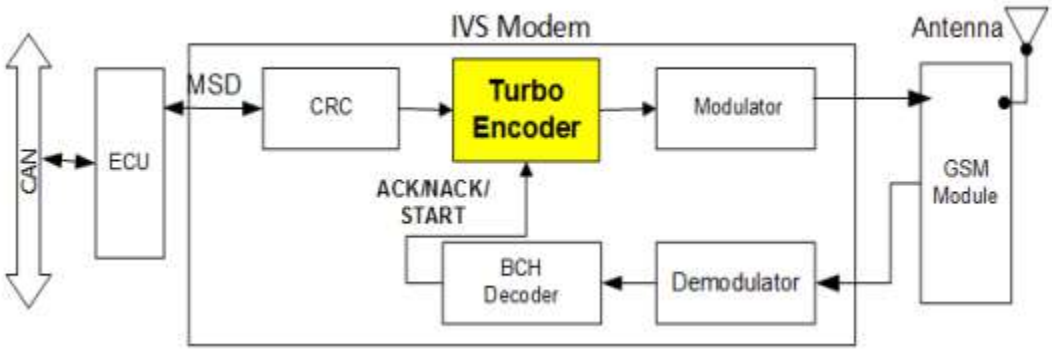

Figure 1: The IVS block diagram.

II. Literature survey:

The Turbo technique has recently attracted research attentions in the NoC community. To show the advantages of TurboNoC, Kim et al. [4] used Walsh codes to distinguish different senders and develop a hierarchical star-mesh topology to handle a large number of communication processors. The simulation results show that the TurboNoC has good performance in latency and throughput. In [7], a specific application is scheduled onto a Turbo-based NoC and a conventional crossbar-based packet-switched NoC. The experimental results show that the TurboNoC achieves lower packet transfer latency and less area overhead. To further improve the TurboNoC performance, a Globally Asynchronous Locally Synchronous (GALS) TurboNoC is

modeled and simulated in [8]. By applying the GALS strategy to TurboNoC, the TurboNoC can be used for asynchronous chips. In [9], the multicasting function is realized in TurboNoC to address the hotspot problem at the center of NoC with mesh topology. The results exhibit that traffic congestion at the center of NoC is reduced. Since code utilization rate affects the TurboNoC performance, two methods on code word assignment are separately proposed in [10] and [11]. In [10], the length of code word is adjusted depending on the number of nodes that have packets to send at the same time. In [11], a scheduling method is proposed to make a compromise on the utilization rate of code words. Simulation results show that these methods improve the utilization efficiency of the orthogonal codes. Besides the traditional wired NoC, the Turbo technique has also been applied to photonic NoC [12] and wireless NoC [13]. The results show that their performance can be significantly improved with lower energy and area compared with the crossbar-based NoC.All the above-mentioned TurboNoCs use the parallel encoding/ decoding method, and thus they have the drawbacks mentioned. Our proposed serial encoding/decoding method shall overcome the common weaknesses.

III. NEW Turbo ENCODING AND DECODING METHOD

A. Overall Structure of TurboNoC

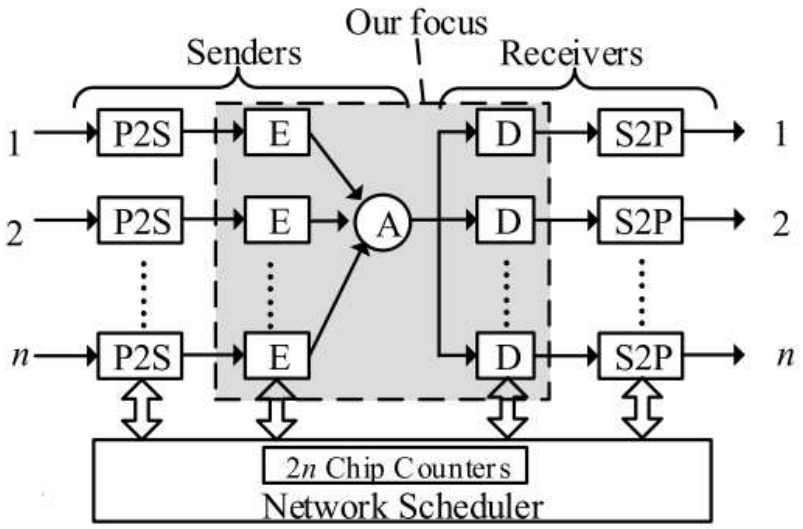

Figure2. Structure of TurboNoC

The basic structure of applying Turbo technique to NoC with a star topology is shown in Figure 2. In this Figure, a PE executes tasks of the application and Network Interface (NI) divides data flows from PE into packets and reconstruct data flows by using packets from NoC. In the sender,

packet flits from NI are transformed to a sequential bit stream via a Parallel-to-Serial (P2S) module. This bit stream is encoded with an orthogonal code in the Encoding module (E in Figure2). The coded data from different encoding modules are added together in the Addition module (A in Figure2).

Then, the sums of data chips are transmitted to receivers. In the receiver, Decoding modules (D in Figure 2) reconstruct original data bits from the sums of data chips. Then these sequential bit streams are transformed to packet flits by Serial-to-Parallel (S2P) modules. Finally, these packet flits are transferred to NI. In the TurboNoC, network scheduler receives the transmitting requests from senders and assigns proper spreading codes to the senders and requested receivers. Note that all-zero codeword is assigned to nodes having no data to transmit/ receive. Moreover, when there are multiple senders requesting the same receiver, the scheduler will apply an arbitration scheme, for example, round-robin.

The chip counters calculate how many orthogonal chips are used in one encoding/decoding operation. Each node needs two chip counters, one for the sender and the other for the receiver. Note that packet flits from NI can also be transformed to multiple bit streams in the P2S module to make tradeoffs between power/area cost and packet transfer latency, and the scheduler should provide a bit-synchronous scheme to maintain the orthogonality of the transmitted channels, as discussed in [8]. In this brief, we focus on the design and comparison of parallel- and serial-based Turbo encoding/decoding method, which corresponds to E, A, and D modules in Figure2.

B. Turbo Encoder

Two different encoding methods, parallel encoder and serial encoder, are compared in Figure3.

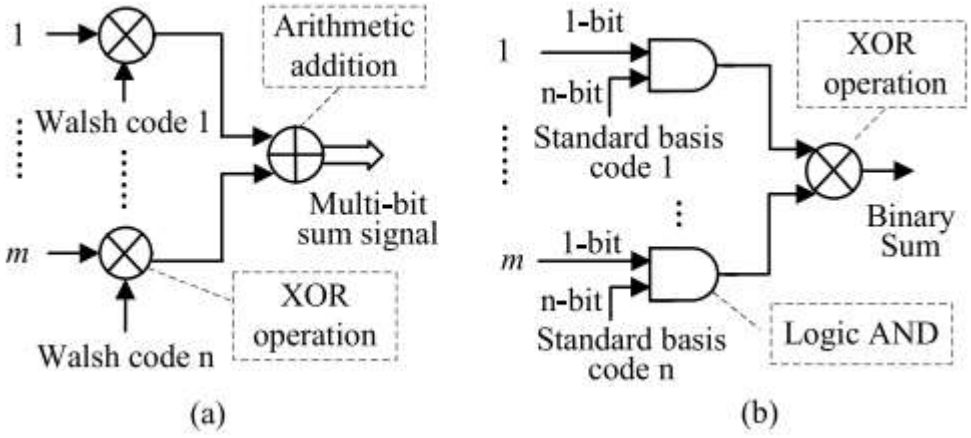

Figure3. Block diagram of encoding scheme. (a) parallel encoder. (b) serial encoder.

Figure3(a) shows the parallel encoder architecture. An original data bit is first encoded with a Walsh code by taking an XOR operation. Then, these encoded data are added up to a multibit sum signal by taking arithmetical additions. Each sender needs an XOR gate, and multiple wires are used to express the sum signal if we have two or more senders. Moreover, the number of wires increases as the number of senders increases.

Figure3(b) shows our serial encoding scheme. An original data bit from a sender is fed into an AND gate in a chip-by-chip manner, and it will be spread to n-chip encoded data with an orthogonal code of a standard basis. The relationship between a bit and a chip is shown in Figure4. Then, the encoded data from different senders are mixed together through an XOR operation, and a binary sum signal is generated. Therefore, the output signal is always a sequence of binary signal transferred to destination using one single wire. The progressions of both the encoding schemes are depicted in Figure4. Figure4(a) and (b) illustrates the parallel encoding process with four-chip Walsh codes and the serial encoding process with four-chip standard orthogonal codes, respectively.

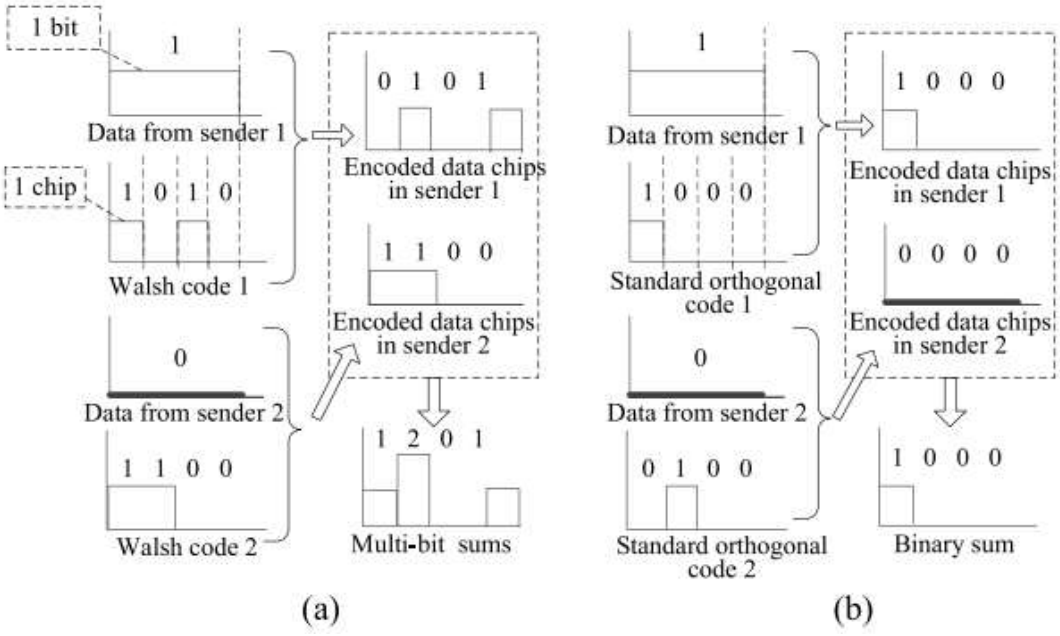

Figure 4: Data encoding example. (a) parallel encoding. (b) serial encoding.

C. Turbo Decoder

The parallel decoding scheme is presented in Figure5(a). According to the chip value of Walsh code, the received multibit sums are accumulated into positive part (if the chip value is 0) or negative part (if the chip value is 1). Therefore, the two accumulators in the parallel decoder

separately contain a multibit adder to accumulate the coming chips and a group of registers to hold the accumulated value. Through the comparison module after the two accumulators, the original data is reconstructed. If the value of positive part is large, the original data is 1. Otherwise, the original data is 0. The serial decoding scheme is shown in Figure5(b).

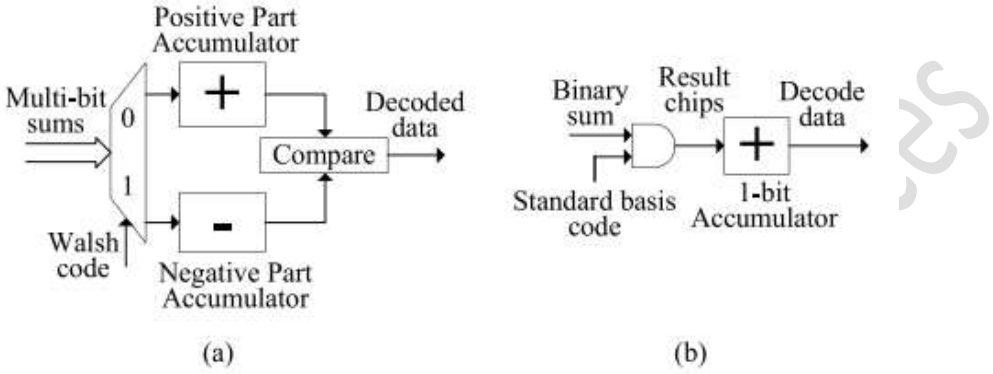

Figure 5. Block diagram of decoding scheme. (a) parallel decoder. (b) serial decoder.

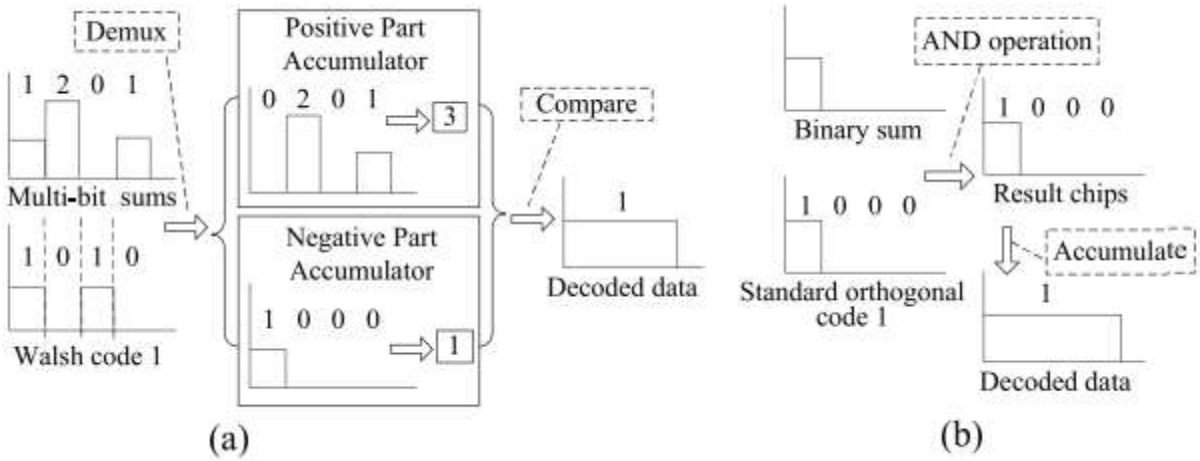

Figure 6. Data decoding example. (a) parallel decoding at receiver 1. (b) serial decoding at receiver 1.

When the binary sum signal arrives at receivers, an AND operation is taken between the binary sum and the corresponding orthogonal code in chip-bychip manner. Then, the result chips are sent to an accumulator. After m-chips are accumulated (m is the length of the orthogonal code), the output value of the accumulator will be the corresponding original data. Note that there is always only one chip equal to 1 and all otherchips are equal to 0 for an orthogonal code in

standard basis. Hence, the maximal accumulated value in the serial accumulator is 1 and it can be stored in a 1-bit register. Therefore, in the serial decoding module, only one AND gate and an accumulator with one 1-bit register are used, resulting in less logical resources.

An example of the decoding process is illustrated in Figure6. In Figure6(a), at the parallel decoder of receiver 1, the accumulated value 3 in the positive part is larger than the accumulated value 1 in the negative part. By the parallel decoding scheme, the decoded data is 1, which is equal to the source data bit from sender 1. In Figure6(b), at the serial decoder of receiver 1, the output value of the accumulator is 1, which is also equal to the source data bit from sender 1. Note that the decoding results in receiver 2 are also correct, but are not shown in the Figure. Hence, both methods can reconstruct the original data bit from the sum signal by using their respective spreading codes.

IV. SIMULATION RESULTS

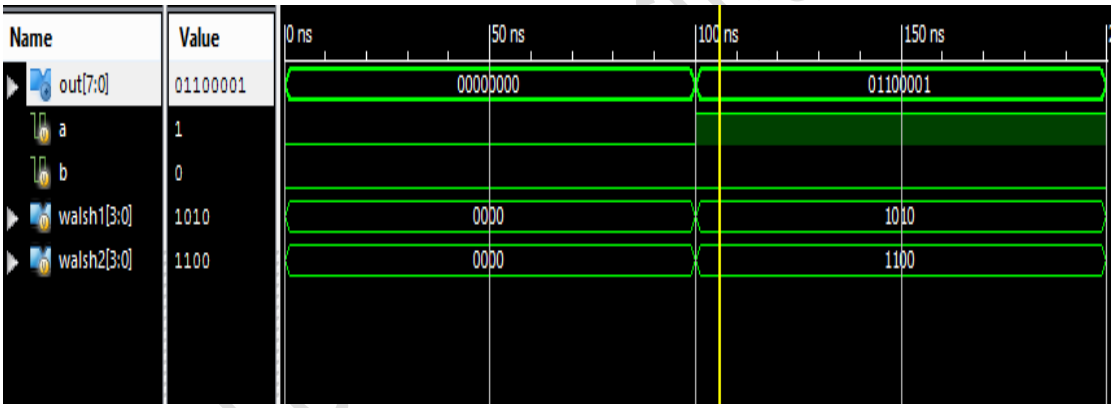

Figure 7:parallel encoder

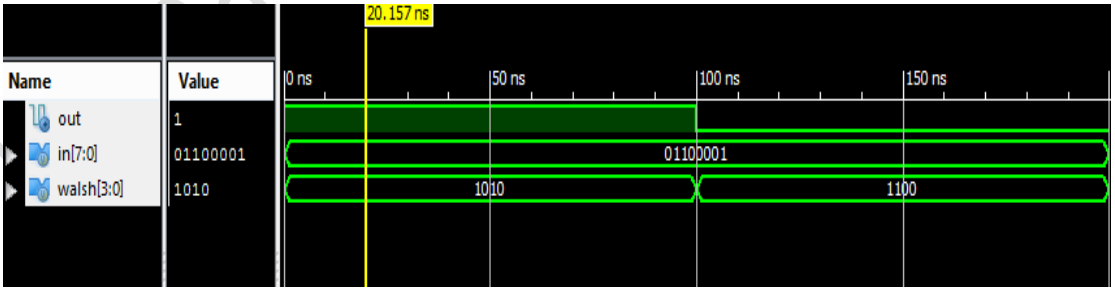

Figure 8: parallel decoder

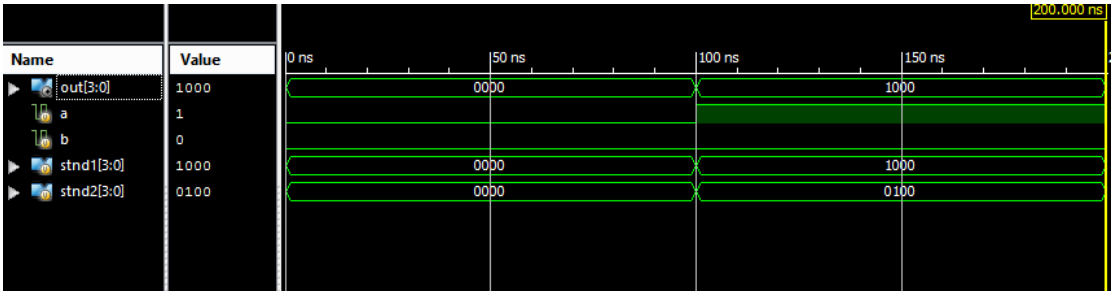

Figure 9: serial encoder

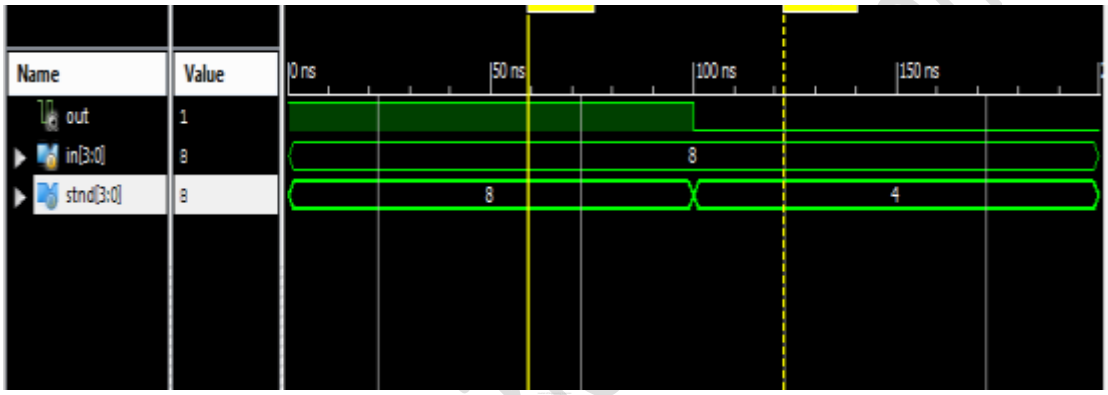

Figure 10: serial decoder

From Figure 7 and Figure 8, the encoded data from two senders are mixed together through xor operation, and a binary sum signal is generated.Therefore, the output signal is always a sequence of binary signal transferred to destination using one single wire. The progression of both the encoding schemes is depicted from Figure 7 and Figure 8. In parallel decoding schemethe chip value of Walsh code, the received multi bit sums are accumulated positive part or negative part by using comparator we have to compare positive and negative parts, if positive is greater than negative then the original data is 1, otherwise the original data is 0.

From Figure 9 and Figure 10 in serial encoding scheme original data bit from a sender is fed into an AND gate in chip by chip manner and encoded data from a different senders are mixed together by an XOR operation and a binary sum signal is generated. In serial decoding scheme the binary sum signal arrives at receivers,an AND operation is taken between binary sum and corresponding sum then the result is send to an accumulator the output of the accumulator will be the corresponding original data.

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	4	5888	0%
Number of 4 input LUTs	8	11776	0%
Number of bonded IOBs	18	372	4%

Figure 11: Design Summary

```
Timing constraint: Default path analysis
  Total number of paths / destination ports: 32 / 8
--------------------------------------------------------------------------------
Delay:               7.337ns (Levels of Logic = 3)
  Source:            b (PAD)
  Destination:       out<6> (PAD)

  Data Path: b to out<6>
                              Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             8   0.849   0.900  b_IBUF (b_IBUF)
     LUT4:I0->O            1   0.648   0.420  out<6>1 (out_6_OBUF)
     OBUF:I->O                 4.520          out_6_OBUF (out<6>)
    ----------------------------------------
    Total                      7.337ns (6.017ns logic, 1.320ns route)
                               (82.0% logic, 18.0% route)
```

Figure 12: TimeSummary

V. Conclusion: The Turbo encoder module is designed and implemented to be an embedded module in the IVS modem. FPGA technologies are employed to develop the Turbo encoder module. Xilinx tools and Verilog are employed to design and simulate the module. We propose a new Turbo encoding/decoding method for on-chip communication. It can be realized by using simple logic and costs less power and area. The standard basis other than the Walsh code is used as the spreading code in our method. It thus decreases the encoding/decoding latency and increases the maximum throughput of NoCs. Mathematical proof is conducted to prove the correctness of our method. From the experimental results, we find that our method outperforms the parallel encoding/decoding scheme, and the TurboNoC performance is also improved when our method is applied.

References:

[1]. D. Sigüenza-Tortosa, T. Ahonen, and J. Nurmi, "Issues in the development of a practical NoC: The Proteo concept," Integr., VLSI J., vol. 38, no. 1, pp. 95–105, 2004.

[2]. A. J. Viterbi, Turbo: Principles of Spread Spectrum Communication. Reading, MA, USA: Addison-Wesley, 1995.

[3]. S. Shimizu, T. Matsuoka, and K. Taniguchi, "Parallel bus systems using code-division multiple access technique," in Proc. Int. Symp. Circuits Syst., May 2003, pp. II-240–II-243.

[4]. D. Kim, M. Kim, and G. E. Sobelman, "Turbo-based network-on-chip architecture," in Proc. IEEE Asia-Pacific Conf. Circuits Syst., Dec. 2004, pp. 137–140.

[5]. X. Wang and J. Nurmi, "An on-chip Turbo communication network," in Proc. Int. Symp. Syst.-Chip, Nov. 2005, pp. 155–160.

[6]. E. H. Dinan and B. Jabbari, "Spreading codes for direct sequence Turbo and wideband Turbo cellular networks," IEEE Commun. Mag., vol. 36, no. 9, pp. 48–54, Sep. 1998.

[7]. M. Kim, D. Kim, and G. E. Sobelman, "MPEG-4 performance analysis for a Turbo network-on-chip," in Proc. Int. Conf. Commun., Circuits, Syst., May 2005, pp. 493–496.

[8]. X. Wang, T. Ahonen, and J. Nurmi, "Applying Turbo technique to network-on-chip," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 15, no. 10, pp. 1091–1100, Oct. 2007.

[9]. W. Lee and G. E. Sobelman, "Mesh-star hybrid NoC architecture with Turbo switch," in Proc. IEEE Int. Symp. Circuits Syst., May 2009, pp. 1349–1352.

[10]. M. Kim, D. Kim, and G. E. Sobelman, "Adaptive scheduling for Turbo-based networks-on-chip," in Proc. 3rd Int. IEEE-NEWCAS Conf., Jun. 2005, pp. 357–360.

[11]. W. Lee and G. E. Sobelman, "Semi-distributed scheduling for flexible codeword assignment in a Turbo network-on-chip," in Proc. IEEE 8th Int. Conf. ASIC, Oct. 2009, pp. 431–434.

[12]. S. Poddar, P. Ghosal, P. Mukherjee, S. Samui, and H. Rahaman, "Design of an NoC with on-chip photonic interconnects using adaptive Turbo links," in Proc. IEEE Int. Conf. SOC, Sep. 2012, pp. 352–357.

[13]. A. Vidapalapati, V. Vijayakumaran, A. Ganguly, and A. Kwasinski, "NoC architectures with adaptive code division multiple access based wireless links," in Proc. IEEE Int. Symp. Circuits Syst., May 2012, pp. 636–639.

[14]. X. Wang and J. Nurmi, "Modeling a code-division multiple-access network-on-chip using SystemC," in Proc. Norchip, Nov. 2007, pp. 1–5.