Performance Improvement of Low Power Scalable Turbo Decoder using NOC Architecture

M.Loga lakshmi
UG Scholar
Department of Electronics Communication Engineering
CCET, Oddanchatram Tamilnadu – 624619

M. Nithya
UG Scholar
Department of Electronics Communication Engineering
CCET, Oddanchatram Tamilnadu – 624619

J. Nivetha
UG Scholar
Department of Electronics Communication Engineering
CCET, Oddanchatram Tamilnadu – 624619

R. Shruthi Eshwari
Assistant Professor
Department of Electronics Communication Engineering
CCET, Oddanchatram Tamilnadu – 624619

Abstract

this work proposes a general framework for the design and simulation of network on chip based turbo decoder architectures. Network on chip or network on a chip (NoC or NOC)is a communication subsystem on an integrated circuit typically between IP cores in a system on a chip(SOC).this explains a network on chip(NoC)structure for turbo decoders. Based on this analysis finite state machine approach are proposed for reducing the complexity of the NoC. Effective architecture level changes of MASIP is used for performance improvement. Here we are analysing the performance of area, power, speed and comparing the results of existing one.

Keywords: Turbo Decoders, Noc, VLSI, Finite State Machine Approach

I. Introduction

In olden days,The wireless communication system connected with the problem of delivering reliable information which gives high throughput.This problems has to be solved by improving the technology while turbo codes able to correct errors even at low signal frequency.In smartphones and tablet PCs which are throughput intensive,it increases the data rate upto hundreads of kilobits per second to hundreads of mega bits per second.Moreover,several works are addressed for turbo decoder architecture to achieve high throughput.there are many works focused on avoiding and reducing the architecture with parallel architecture.This paper presents a multi-core architecture is based on application specific instruction set processor(ASIP) and it reduces the throughput.This turbo decodes are widely used in the modern wireless system for their near-Shannon performance. The various standards in turbo codes they are code ASIP-balancing based.This would lead to an imbalance of the CALS in RTs and FTs, which could invalidate the CALS-based flow control scheme. In addition, there may be some ASIPs that send too many RTs to the NoC, thus intensifying congestion.lengh, code generator, interleaving pattern. The main issue for these ASIP and NoC architecture is more complex.so, we concentrate on NoC architecture which supports high throughput with low efficiency. A general turbo decoder architecture that covers all the standards is more attractive because of the complexity and upgradability. A throughput and area are the dominant things driving the optimization of turbo decoders. Some recent works deals with the simulation of application specific instruction-set processor (ASIP) architecture for turbo decoders. The effective solution is to achieve both high throughput and flexibility is multi-ASIP.This both interconnects and performs the high throughput with flexible multi-ASIP architecture. Previous research has been done on ASIP [1], [2] and NoC structures for turbo decoding. The main drawback of this ASIP and NoC architecture is more complex. In order to overcome the complexity we are performing NoC on turbo decoders by finite state machine approach. It gives the new effective architecture and effective higher throughput with low NoC area.

II. System requirement analysis

A code frame with N bits of information is split into P overlapped windows. A single window is processed by single ASIP.It generates the memory addresses.ASIP is a component used in SOC design. The instruction set of ASIP is tailored to benefit a specific application. This specification of core provides a trade-offs between the flexibility.ASIP provides a configurable instruction set. Usually these are divided into two parts: static logic-defines a minimum instruction set architecture and configurable logic: defines used to design new instructions. The existing method is based on network topology and NoC data format. The existing NoC architecture is shown in fig [2].in existing, NoC is divided into parallel subnetworks, RTs and WTs from ASIP is distributed to subnetworks.this is completed by DISTRIBUTOR. The buffer data from ASIP ports in each DISTRIBUTOR send them to different sub networks according to the rule .a transaction is sent to the DEST ADDR is the destination memory bank address. A memory bank receives an RT from the sub network and sends a FT back to the subnetwork. This FT is going along routers back to the router corresponding to ASIP through a CONVERGER which has the reverse effect of the DISTRIBUTOR module. The trellis diagram is used for comparing the performances of NoC architecture. In the existing NoC, they adopt the deterministic routing policy for its shorter critical path and complexity. This kind of architecture is more complex and it leads to poor performance.

A. Multi-Asip for Turbo Decoding

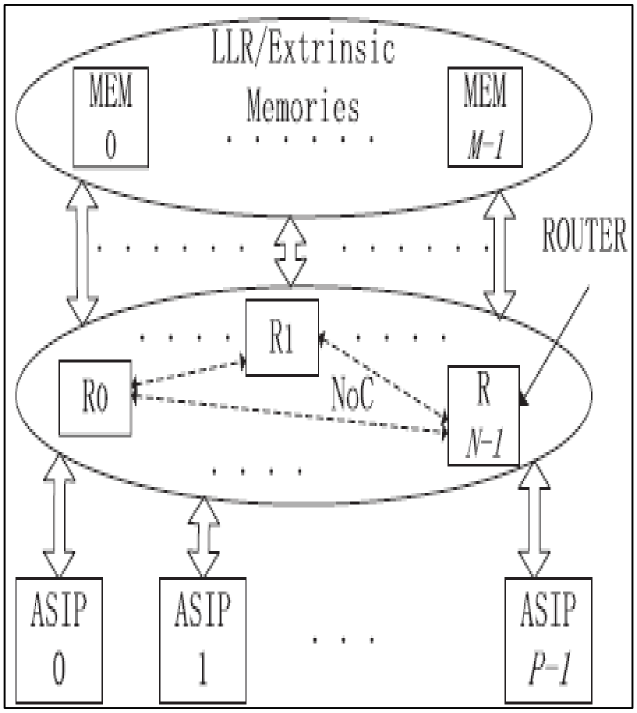


Fig. 1: Example of multi ASIP for turbo decoding

III. EXISTING METHOD

A. *Noc Architecture*

In existing NoC, we are calculating the deterministic routing policy for its shorter critical path and lower complexity.tranmisssion will be routed along this path and low complexity. The routing policy can reduce the total number of hops with cost of same delay. This kind of priority scheme provides quality of service but also has a high complexity and a long critical path.

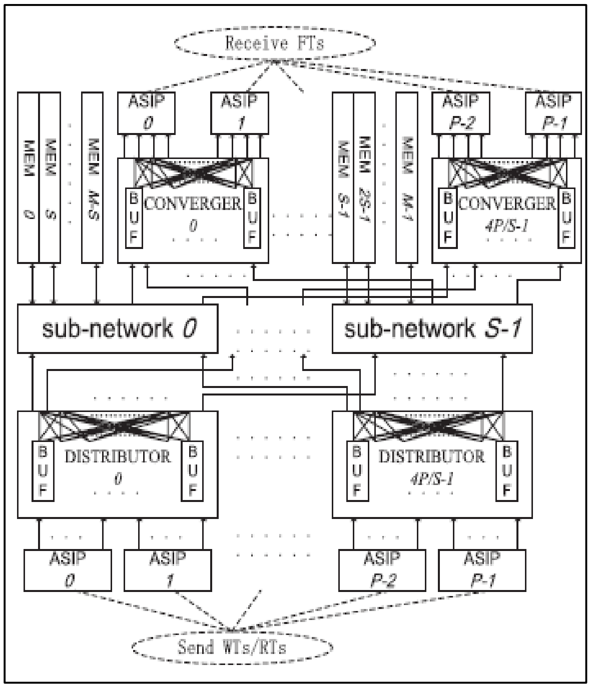


Fig. 2: Existing NoC architecture

B. *Flow Control*

The flow control is used to solve the deadlock problem. Here they are using two kinds of schemes: 1) CALS based and 2) When the difference in halted cycles exceeds a given limit called Nhalt_limit, the ASIPs having the smallest Nhalt will be blocked. The CALS-based scheme is to constrain the number of transactionsin the NoC for each ASIP. When the difference in halted cycles exceeds a given limit called Nhalt_limit, the ASIPs having the smallest Nhalt will be blocked.

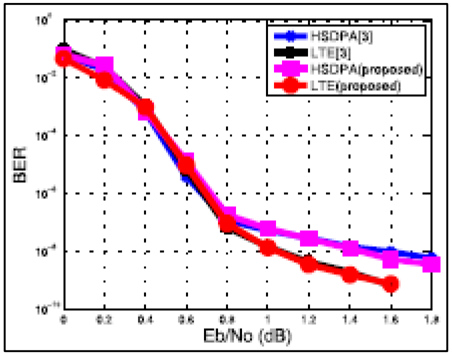


Fig. 3: BER performance

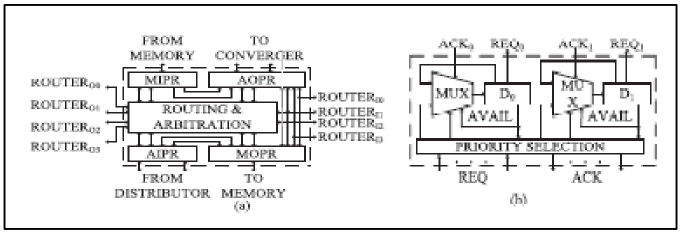


Fig. 4: Router architecture. (b) Pipeline registers module

They implemented a cycle-accurate multi-ASIP architecture simulatorin System C. This was used to generate the test bench and estimatethe decoding throughput. Then, we implemented the NoC circuit withVerilog-Hardware Description Language and synthesized it in a 0.13-µm ASIC standard cell library. We choose six for Itseq and five for limit. The LLR and LE are 6 and 8 bits, with three fractional bits.Fig.3 gives the simulated bit error rate (BER)performance curve, which is very close to that in [3]. The maximum frequency is 300MHz. Through simulations, we found 5 bits to be sufficient for Class we chose R = 32.In addition, CAL limit is 10 and Nhalt_limit is Six in both LTE and HSDPA cases. They have proposed a novel NoC architecture for amulet-ASIP turbo decoder. By dividing the entire network into several sub networks and adopting the notion of the calculation sequence, we successfully achieved a much higher throughput with a lower NoCtotal area. Comparisons with previous designs for area, throughput, and power efficiency have been given. Our design technique scan also be used in other applications, such as Low Density Parity Check decoding. The sub network scheme can be employed wherever there is a uniform addressing mode, and the CALS method is appropriate for trellis-based decoding.

IV. Proposed method

A. New Efficient Architecture

The multi application specific instruction processor is a promising area for turbo decoders. An effective architecture level change of MASIP is used for performance improvement. Performance analysis and comparison. In modern wireless systems, turbo codes are widely used for their near-Shannon performance. In various standards, their turbo codes vary mainly in three aspects: 1.code length, 2.code generator and 3.interleaving patterns. The main issue for these ASIP and NoC architectures is the complexity overhead and thus low efficiency compared with dedicated decoders. So, we focus on NoC architecture which supports high throughput with low efficiency. Computational complexity is less. Performance improved in FPGA (area, power, speed).Advantage of these proposed method is new effective architecture.

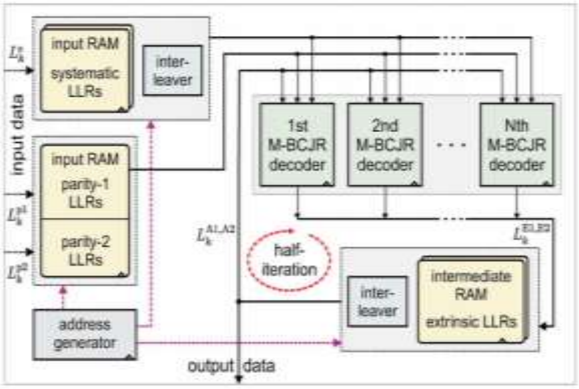


Fig. 5: Proposed NoC architecture

Due to dramatic increase in portable and battery operated applications, lower power consumption has become a necessity in order to prolong the battery life. We focus on NoC architecture which supports high throughput with low efficiency. Reconfigurable computing refers to systems incorporating some form of hardware programmability, that customizes how the hardware is used using a number of physical control points.These control points can be changed periodically in order to execute different applications using the same hardware. Since, the inconsistent requirements of modern applications T. Vogt and N. Wehn, “A reconfigurable ASIP for convolutional and Turbo decoding in an SDR environment,” *IEEE Trans. Very Large ScaleIntegr. (VLSI) Syst.*, vol. 16, no. 10, pp. 1309–1320, Oct. 2008. for both flexibility and implementation efficiency, cannot be satisfied by conventional instruction-set processors and application-specific circuits, reconfigurable hardware offers a good balance between implementation efficiency and flexibility. This is because the reconfigurable hardware combines post-fabrication programmability with the parallel computation style of application specific circuits, which is more efficient in comparison to the sequential computation style of instruction-set processors. There are additional reasons for using reconfigurable resources in System-on-Chip (SoC) design. The increasing non-recurring engineering (NRE) costs push designers to use the same SoC in several applications and products for achieving low cost per chip. The presence of reconfigurable resources allows the fine tuning of the chip for different products or product variations. Also, the increasing complexity in future designs adds the possibility of using design flows, which can require costly and slow redesign of the chip. In this way: Reconfigurable elements are often homogenous arrays, which can be pre-verified to minimize the possibility of design errors. Post manufacturing programmability of reconfigurable elements allows correction of problems. In order to reconfigure a statically reconfigurable architecture, the system has to be halted while the reconfiguration is in progress and then restarted with the new configuration. Traditional FPGA architectures are primarily statically programmed devices, allowing only one configuration to be loaded at a time. This type of FPGAs is programmed using a serial stream of configuration information (stored in an SRAM), requiring a full reconfiguration if any change is needed. Whereas static reconfiguration allocates logic for the duration of an application, dynamic reconfiguration (often referred as run-time reconfiguration) uses a dynamic allocation scheme that re-allocates hardware at run time (i.e. during execution of the application).The physical hardware is smaller than the sum of required resources. With dynamic reconfiguration we swap the number of configurations in and out of the actual hardware, as they are needed. Problems is Divide the algorithms into time-exclusive segments that do not need to run concurrently and manage the transmission of intermediate results from one configuration to the next. Each FPGA slice contains two basic reconfigurable logic blocks. The 4-bit look-up table (LUT) is implemented with a multiplexer whose select lines are the inputs of the LUT and whose inputs are constants. A typical block for logic reconfiguration contains a look-up table (LUT), an optional D flip-flop (latch) and additional combinational logic.The LUT allows any logic function to implemented, providing generic logic. The latch can be used pipelining reasons, registers for holding logic values or any other situation where clocking is required. The additional combinational logic is usually ‘ carry logic ’ used to speed up carry-based computations (e.g. additions). In addition to operating as a function generator, each LUT can provide RAM functionality. Furthermore, two or more logic blocks can be combined to implement more complex functions.

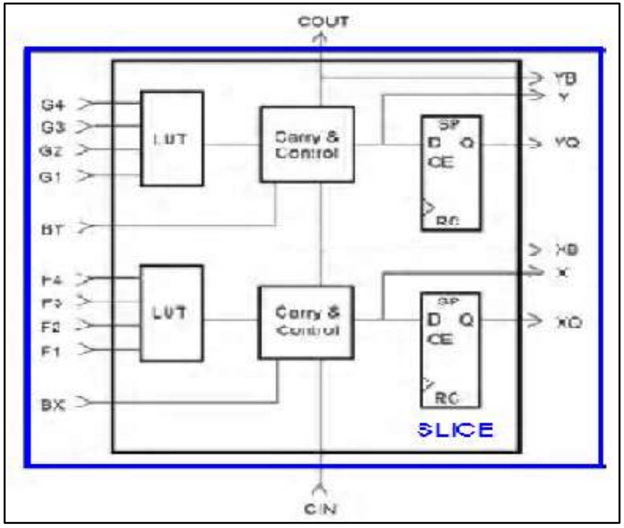
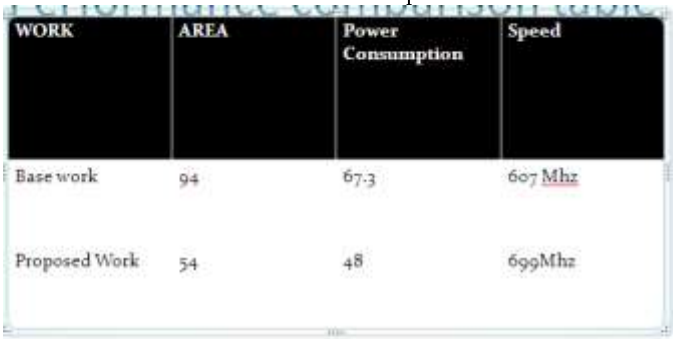


Fig. 6: Example of basic logic block (Xilinx Vertex FPGA)

V. Implementation Result

In proposed method, new efficient architecture is used to improve the performance of turbo decoders and the following changes are made Fault tolerant Architecture Reconfigurable memory instead of conventional memory. Power consumption is an important part of the equation determining the end product's size, weight, and efficiency. Selecting an appropriate FPGA architecture is critical in achieving the best static and dynamic power consumption. Flash-based FPGAs by Micro semi are the low-power leaders in the industry. In addition to utilizing the low power attributes of flash-based FPGAs, you can deploy several design techniques to further reduce overall power. The important FPGA power components to consider in the following sections: Configuration power: Configuration power is the amount of power required during the loading of the FPGA upon power-up (specific to SRAM-based programmable logic devices). Static (standby) power: Static power is the amount of power the device consumes when it is powered-up but not actively performing any operation. Sleep power (low-power mode): Some FPGA devices offer low-power or sleep modes. In some cases, this may be different from static power. This application note focuses on reducing the dynamic power. Currently, most circuits adopt static random access memory (SRAM) plus some control/addressing logic to implement delay buffers. For smaller length delay buffers, shift register can be used instead. The former approach is convenient, since SRAM compilers are readily available and they are optimized to generate memory modules with low power consumption and high operation speed with a compact cell size.

Table - 1 Performance Comparison Table



WORK	AREA	Power Consumption	Speed
Base work	94	67.3	607 Mhz
Proposed Work	54	48	699Mhz

VI. Conclusion

So far work completed in FPGA platform. Both proposed is compared with conventional work (base work).Its proving better result when compare with existing work.After base work architecture implementation the following results areachieved.AREA=54slicesPOWER=48%,SPEED=699Mhz.Similarly, the proposed effective architecture encapsulates with base work we achieved following result. We successfully achieved a much higher throughput with a lower NoC total area.comparision with previous designs for area, speed, and power efficiency has been given. Our design techniques can also be used in other applications, such as multimedia communication.

REFERENCES

[1] M. Martina and G. Masera, “Improving network-on-chip-based turbo decoder architectures,” *J. Signal Process.Syst.*, vol. 73, no. 1, pp. 83–100,Oct. 2013.

[2] M. Martina and G. Masera, “Turbo NOC: A framework for the design of network-on-chip-based turbo decoder architectures,” IEEETrans. Circuits Syst. I, Reg. Papers, vol. 57, no. 10, pp. 2776–2789,Oct. 2010.

[3] M. Martina, G. Masera, H. Moussa, and A. Baghdadi, “On chip interconnects for multiprocessor turbo decoding architectures,” MicroprocessorsMicrosyst., vol. 35, no. 2, pp. 167–181, Mar. 2011.

[4] O. Muller, A. Baghdadi, and M. Jézéquel, “From parallelism levels toa multi-ASIP architecture for turbo decoding,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 17, no. 1, pp. 92–102, Jan. 2009.

[5] F. Naessens et al., “A 10.37 mm2 675 mWreconfigurable LDPC and turbo encoder and decoder for 802.11n, 802.16e and 3GPP-LTE,” in Proc. IEEESymp. VLSI Circuits (VLSIC), Jun. 2010, pp. 213–214.

[6] C. Shung, P. Siegel, G. Ungerboeck, and H. Thapar, VLSI architectures for metric normalization in the Viterbialgorithm,” in Proc. GLOBECOM,Apr. 1990, pp. 1723–1728.

[7] C. Studer, C. Benkeser, S. Belfanti, and Q. Huang, “Design and implementation of a parallel turbo-decoder ASIC for 3GPP-LTE,” IEEE J.Solid-State Circuits, vol. 46, no. 1, pp. 8–17, Jan. 2011.

[8] T. Vogt and N. Wehn, “A reconfigurable ASIP for convolutional and turbo decoding in an SDR environment,” IEEE Trans. Very Large ScaleIntegr.(VLSI) Syst., vol. 16, no. 10, pp. 1309–1320, Oct. 2008.