Design of Efficient High Throughput Pipelined Parallel Turbo Decoder Using QPP Interleaver

S. M. Karim and Indrajit Chakrabarti

Abstract— This paper introduces a novel energy efficient architecture for a turbo decoder using quadratic permutation polynomial (QPP) interleaver. The Add Compare Select Offset (ACSO) unit of the maximum a posteriori probability (MAP) decoder, has been pipelined to a depth of four to reduce the critical path delay and increase the operating clock frequency and throughput as a consequence. The present turbo decoder architecture also benefits from a contention-free quadratic permutation polynomial (QPP) based interleaver, the complexity of which has been considerably reduced by judicious memory partitioning. Typically, as demonstrated in the present work, 32 MAP decoder core can achieve a data rate of 1.138 Gbps at a maximum clock frequency of 486 MHz when implemented in a 90 nm CMOS process.

I. INTRODUCTION

Turbo codes [1] have shown impressive performance as practically realizable channel capacity approaching codes. They have been employed for realizing 3GPP LTE, cdma2000, satellite services for the digital video broadcast standard and many more. Turbo decoding involves iterative processing of a complete data block using two component decoders based on soft in soft out (SISO) decoding algorithm. The reasonably higher decoding latency makes it unsuitable for latency-sensitive applications. Efforts have been made to implement the turbo decoder suitable for low power hand held device, high throughput modern wireless device and high performance storage.

The MAP algorithm can be simplified to an equivalent sliding window (SW) Log-MAP [2] algorithm for efficient hardware realization. Parallel architecture [3]-[5] is an intuitive method to enhance the throughput, where a number of MAP decoders known as workers [4] are assigned to decode concurrently the different sub blocks of the data frame. The design bottleneck due to the critical path delay for the Add Compare Select Offset (ACSO) block can be effectively removed by pipelining ACSO [6]. The initial values at the boundary of a sliding window can be obtained either by acquisition [2] over a several trellis stages or by the next iteration initialization (NII) [3] method. Combination of the two methods can be taken up to reduce the acquisition length which in turn can drastically reduce the storage requirement of various computational metrics. To enhance the throughput realizable by the various approaches undertaken so far, the present work has adopted the following architectural strategies.

Firstly a combination of pipelined ACSO structure and parallel worker architecture has been introduced to utilize the hardware resources efficiently. Secondly, NII assisted acquisition technique is followed in the present work. Thirdly, by reducing acquisition length, the state metric storage within individual workers is reduced, leading to economy in area as well as power. Fourthly, by configuring a single memory block in alternate half iteration as an interleaver and a de-interleaver, area has been reduced by half. Moreover, memory partitioning in interleaver has been performed so that it supports and works in harmony with the pipelined parallel structure of the decoder. By incorporating all the above stated measures, the proposed turbo decoder has been implemented, comparison of the proposed architecture with several existing implementation have established its superiority in key design characteristics.

The remainder of this paper is organized as follows. Section II describes the Log-MAP turbo decoding. Section III illustrates the pipelined parallel architecture of the MAP decoder. The contention free interleaver is illustrated in section IV. The VLSI implementation results of the proposed turbo decoder are summarized and compared with the existing turbo decoders in Section V. Finally, Section VI concludes this paper.

II. LOG-MAP TURBO DECODER

Log-MAP algorithm determines the logarithmic value of the likelihood ratio via equation (1), where \( P(\cdot) \) denotes the a posteriori probability for each transmitted bit \( u_k \) given the received symbol sequence \( y \).

\[
L(u_k) = \ln \frac{P(u_k = +1 / y)}{P(u_k = -1 / y)}
\]  

(1)

The equation (2) describes the LLR calculation based on the state metrics associated with the trellis stage of the encoder.

\[
L(u_k) = \max_{\epsilon \in \epsilon_{u_k}} [\alpha_k(s') + \beta_k(s) + \gamma_k(s', s)] - \max_{\epsilon \in \epsilon_{u_k}} [\alpha_k(s') + \beta_k(s) + \gamma_k(s', s)]
\]  

(2)
where \( \alpha \), \( \beta \), and \( \gamma \) refer to the forward state metrics, backward state metrics, and branch metrics represented by the recursive equation (3), (4), and (5)

\[
\alpha_{s+1}(s) = \max_k [\alpha_k(s) + \gamma_k(s', s)] \\
\beta_{s+1}(s') = \max_k [\beta_k(s) + \gamma_k(s', s)] \\
\gamma_k(s', s) = \ln P(y_i / s, s') + \ln P(s, s')
\]

The \( \max^* \) operation is defined as

\[
\max^*(a, b) = \max(a, b) + \ln(1 + e^{-|a-b|}).
\]

The state metrics \((\alpha_{k+1}, \beta_{k-1})\) associated with the trellis stage depend on the state metrics of the previous trellis steps. The Add Compare Select Offset (ACSO) unit calculates it recursively. The speed bottleneck of the decoder design at high operating frequency arises fundamentally due to the ACSO unit [7] making the decoder unsuitable for high throughput application. As shown in Fig. 1, critical path can be reduced by inserting registers at appropriate locations to make it pipelined ACSO.

**III. ARCHITECTURE FOR PIPELINED PARALLEL MAP DECODER**

The key method to accelerate the decoding process is to deploy multiple workers [3]-[5] to decode several data blocks in parallel. The proposed worker shown in Fig. 2 consists of an \( \alpha \) unit (BMC), two \( \beta \) unit (BSMC_D) and (BSMC), an \( \gamma \) unit (FSMC), a log-likelihood-ratio calculator (LLRC) and two memory blocks RAM\(_A\) and RAM\(_B\). The BSMC_D calculates the dummy \( \beta \) with the help of the initial beta values from RAM\(_B\) and the branch metric values from the BMC. The BSMC calculates the actual \( \beta \) using the delayed branch metrics. The FSMC calculates \( \alpha \) which are to be stored in last in first out (LIFO) RAM, while the RAM\(_A\) and RAM\(_B\) are used to hold the state metric border values, namely \( \alpha_{in} \) and \( \beta_{in} \) to be used in the next iteration obtained from the previous and the next workers respectively.

The data frame of size \( K \) is divided into \( N \) subblocks of size \( W \) such that \( K = N \times W \). Moreover, each subblock is assigned to one of the \( N \) decoder modules to enable concurrent processing. The decoding speed is enhanced further by using the pipelined technique along with the parallel processing provided the computation within blocks are recursive and the partitioned subblocks are processed independently. As shown in Fig. 3 the proposed design divides the subblock of size \( W \) further into segments of length \( L_s \) such that \( K = N \times W = N \times P \times L_s \).

The principle outlined above leads to the pipelined parallel architecture of the log-MAP decoder. The \( i \)th worker of the proposed architecture processes the trellis sections \([i-1]_{\frac{K}{N}}, i_{\frac{K}{N}}\) where \( 0 \leq i \leq N - 1 \). In fact, every worker will process \( k, k + L_s, k + 2L_s, ..., k + (P-1)L_s \) in consecutive \( W \) clock cycles from the subblock assigned to it. The gain achieved by the pipelined approach consisting of \( P \) stage is the reuse of different metric calculation unit with a higher operating clock frequency at the cost of an increase in the number of pipelining registers.

**IV. CONTENTION FREE INTERLEAVER ARCHITECTURE**

The highly parallel turbo decoder requires a high bandwidth memory access to satisfy the throughput requirements. The interleaver memory is thereby divided into smaller memory blocks to support the parallel access by different SISO workers. Attempt of concurrent access from the same memory block leads to contentions. The chance of collision, that affects the achievable throughput increases with degree of parallelism. Therefore, designing a collision-free interleaver becomes a challenging task. Adequate buffering [8] to resolve the collision and specially designed interleaver [9]-[10] to prevent the contentions have been proposed. Quadratic polynomial permutation (QPP) interleaver [11] provides efficient collision free interleaving, suitable for high throughput architectures [11]-[12], [14]. Low complexity architecture of multiple QPP address generator and memory bank architecture for the interleaver has been exploited in the present work in order to provide collision free access to a parallel MAP decoder made of multiple workers.

Consider a \( K \) bit long frame divided into \( N \) subblocks each consisting of \( W \) bits. Recall moreover that the ACSO module of the turbo decoder is pipelined with \( P \) stages. Careful observation of QPP address generation [11] reveals that only the most significant bits of \( N \) parallel memory addresses differ.
For example, for an interleaver size of 1024 with degree of parallelism as 8, the QPP will generate eight 10 bit addresses. Out of this, the seven least significant bits have same values, whereas the three most significant bits represent 8 different values. This observation leads to memory folding where $N$ parallel interleaved addresses will be the same row of the different memory blocks. To this end, QPP address generation can be divided into two parts to facilitate working with smaller memory segments. One module generates the upper address to route the output from $N$ parallel workers to the $N$ destination memory blocks while permuting towards interleaving/ deinterleaving is accomplished. The second module generates the remaining lower address bits of the QPP which can be used as the row address of the $N$ memory blocks. In order to support the pipelined operation of the overall turbo decoder each of the $N$ memory blocks is split into $P$ memory subblocks whose read/ write operation will be controlled by the derived clocks based on the pipelined schedule. The architecture of such a partitioned interleaver memory has been shown in Fig. 4. The memory subblocks are arranged in $N$ rows and $P$ columns in such a manner that the $P$ sub blocks from the $i^{th}$ row are connected to the $i^{th}$ SISO module.

The proposed interleaver architecture divides each memory block into $P$ subblocks to access the memory in $P$ alternate clock cycles in order to support the pipelined parallel MAP decoding. During these $P$ clock cycles, the row address will be unchanged, which will in turn reduce the excessive switching in the router. The QPPs are required to generate the addresses at the interval of $P$ clock cycles. The $i^{th}$ SISO module can read from $M_{ij}$ memory subblock, while memory subblock $M_{ik}$ (where $j \neq k$) is updated independently by the data coming from the router. Therefore, with the appropriate pipelining schedule, use of single port memory block $M_{ij}$ ($0 \leq i \leq N - 1, 0 \leq j \leq P - 1$) as shown in Fig. 4 can reduce the interleaving complexity.

The crossbar network is usually implemented by the multiplexer circuit which becomes highly complex and a major source of wiring overheads for high level of parallelism. However, as shown in Fig. 5, use of pass transistor along with wired OR logic can greatly simplify the multiplexer based router implementation of the crossbar network.

**Figure 3.** Division of a frame into subblocks to be processed by $N$ parallel Workers each configured as a $P$ stage pipeline.

**Figure 4.** Interleaver architecture for pipelined parallel turbo decoder.

**Figure 5.** Partial architecture of the crossbar network for pipelined parallel turbo decoder.
V. IMPLEMENTATION RESULTS AND PERFORMANCE

A pipelined parallel turbo decoder module consisting of a number of parallel workers has been described in verilog HDL and implemented using standard cell based design technology. Each submodule in RTL behavioral descriptor of MAP core architecture has been synthesized into gate level netlist and then placed and routed to generate layouts. The estimated key design characteristic of the turbo decoder ASIC as listed in Table I demonstrates the optimized design for the interleaver size of 4096. In the present work two different designs involving 8 and 32 parallel workers have been demonstrated. As shown in the table, design I which employs 8 parallel SISO workers and 4 pipelined stages, achieves a throughput of 332.6 Mbps that can comfortably meet the throughput specified by LTE peak data rate i.e. 326.4 Mbps,. Design II attains 1.138 Gbps throughput with a maximum clock frequency of 486 MHz by utilizing 32 parallel SISO modules.

<table>
<thead>
<tr>
<th>No. of parallel workers</th>
<th>Design I</th>
<th>Design II</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pipelined stages</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Power (mW)</td>
<td>310</td>
<td>361</td>
</tr>
<tr>
<td>Energy Efficiency (nJ/bit/iter.)</td>
<td>0.24</td>
<td>0.19</td>
</tr>
<tr>
<td>Throughput (Mbps)</td>
<td>230</td>
<td>332.8</td>
</tr>
<tr>
<td>Core area (mm²)</td>
<td>9.91</td>
<td>10.15</td>
</tr>
</tbody>
</table>

Both the designs have been developed while utilizing two stage and four stage pipelined ACSO block and the corresponding interleaver has been optimized for that architecture. The routing complexity increases exponentially which will restrict further parallelism. Power consumed by the turbo decoder as shown in Table I is dominated by the interleaver memory for a low degree of parallelism, whereas better energy efficiency could be achieved by increasing parallelism degree at the cost of additional silicon area. This design can be augmented to double its throughput by employing as many as 64 SISO modules at the cost of increase in the interleaver complexity.

Table II demonstrates the key characteristics of the proposed turbo decoder and has been compared with the published works [4], [13]-[14] which clearly shows the superiority of the proposed design for high throughput application based on parallel architecture. The design comprises 3.52M gate equivalent, operating with 486MHz clock to achieve 1.138 Gbps throughput.

VI. CONCLUSION

The present paper describes a novel pipelined parallel turbo decoder architecture which is made of $N$ ($N$ can be 32 or 64) parallel workers such that each worker is pipelined consists of 2 or 4 stages. It has been demonstrated that the proposed architecture has outperformed several published works. As it achieves a throughput of 1.138Gbps at 486 MHz, the proposed decoder can find application in 3GPP-LTE standard in wireless communication and high performance storage.

REFERENCES