Energy Efficient Decoder Architecture For Communication System

Chandana P, Agalya P, B. N Shoba

Abstract— In the communication domain low complexity and low power are the main criteria to be considered for the efficient utilization of the communication channel, which also requires reduction in the error rate. Turbo codes aids the energy constrained wireless communication by providing less energy consumption during data sending and receiving operation. In this paper, we are breaking down the LUT-Log-BCJR architecture into most fundamental add compare select unit and perform them using novel low complexity ACS unit, since we are using ACS unit with LUT architecture it employs an order of magnitude of fewer gates than previous LUT-log BCJR architecture, providing approximately 72% energy consumption reduction compared to existing previous max log algorithm.

Index Terms — Energy-efficient, error-correcting code (ECC), LUT-Log-BCJR algorithm, turbo code.

I. INTRODUCTION

MODERN wireless communication standards rely on powerful channel coding to ensure reliable (error free) transmission. Turbo codes have recently been considered for energy-constrained wireless communication applications, since they facilitate a low transmission energy consumption. However, in order to reduce the overall energy consumption, look up table-log-BCJR (LUT-Log-BCJR) architectures having a low processing energy consumption are required. In this paper, we decompose the LUT-Log-BCJR architecture into its most fundamental add compare select (ACS) operations and perform them using a novel low-complexity ACS unit.

II. RELATED WORK

Recent application-specific integrated circuit (ASIC) –based turbo decoder architectures [5]–[7] have been designed for achieving a high transmission throughput, rather than for a low transmission energy.

For example, turbo codes have facilitated transmission throughputs in excess of 50 Mbit/s in cellular standards, such as the 3rd Generation Partnership Project 3GPP Long Term Evolution (LTE) and recent ASIC turbo decoder architectures have been designed for throughputs that are in excess of 100 Mbit/s [5], [6]. This has been achieved by employing the Max – Log - BCJR turbo decoding algorithm, which is a low-complexity approximation of the optimal Logarithmic Bahl-Cocke-Jelinek-Raviv (Log-BCJR) algorithm [8].

The Max-Log-BCJR algorithm appears to lend itself to both high-throughput scenarios, as well as to the above-mentioned energy-constrained scenarios. This is because a low turbo decoder energy consumption is implied by Max-Log-BCJR algorithm’s low complexity. However, this is achieved at the cost of degrading the coding gain by 0.5 dB compared to the optimal Log-BCJR algorithm [9], increasing the required transmission energy by 10%. This disadvantage of the Max-Log-BCJR outweighs its attractively low complexity, when optimizing the overall energy consumption of sensor nodes that are separated by dozens of meters. This motivates the employment of the lookup-table-log- BCJR (LUT-Log-BCJR) algorithm [8] in energy-constrained scenarios, since it approximates the optimal Log-BCJR more closely than the Max-Log-BCJR and therefore does not suffer from the associated coding gain degradation. However, to the best of our knowledge, no LUT-Log-BCJR ASICs have been specifically designed for energy-constrained scenarios. Previous LUT-Log-BCJR turbo decoder designs [10]–[13] were developed as a part of the on-going drive for higher and higher processing throughputs, although their throughputs have since been eclipsed by the Max-Log-BCJR architectures. This opens the door for a new generation of LUT-Log-BCJR ASICs that exchange processing throughput for energy efficiency. The energy consumption of conventional LUT-Log-BCJR architectures cannot be significantly reduced by simply reducing their clock frequency and throughput. This motivates our novel architecture of, which is specifically designed to have a minimal hardware complexity and hence a low energy consumption.

we validate our architecture in the context of an LTE turbo decoder and demonstrate that it has an order of magnitude lower chip area, hence reducing the energy consumption of the state-of-the-art LUT-Log-BCJR implementation by approximately 72%. Compared to state-of-the-art Max-Log-BCJR implementations, our approach facilitates approximately 10% reduction in the overall energy consumption of at transmission ranges above 50 m.
III. PROPOSED SYSTEM

The lookup-table-log-BCJR (LUT-Log-BCJR) algorithm is useful in energy-constrained scenarios, since it approximates the optimal Log-BCJR more closely than the Max-Log-BCJR and therefore does not suffer from the associated coding gain degradation. We validate our architecture and demonstrate that it has an order of magnitude of lower chip area, hence reducing the energy consumption of the state-of-the-art LUT-Log-BCJR implementation by 72%. We can keep on changing the architecture of lookup-table-log-BCJR to achieve an efficient frequency which in turn helps in Power Consumption.

One of the most interesting characteristics of a turbo code is that it is not just a single code. It is, in fact, a combination of two codes1 that work together to achieve a synergy that would not be possible by merely using one code by itself. In particular, a turbo code is formed from the parallel concatenation of two constituent codes separated by an interleaver. Each constituent code may be any type of FEC code used for conventional.

The interleaver is a critical part of the turbo code. It is a simple device that rearranges the order of the data bits in a prescribed, but irregular, manner. Although the same set of data bits is present at the output of the interleaver, the order of these bits has been changed, much like a 1Turbo codes can also be constructed using three or more constituent codes. Such structures are called multiple turbo codes. However, they are not used in any standards and are therefore not discussed here.

Our architecture implements the entire algorithm using ACS units in parallel, each of which performs one ACS operation per clock cycle. Furthermore, the proposed architecture employs a twin-level register structure to minimize the highly energy-consuming main-memory access operations. At the first register level, each ACS unit is paired with a set of general purpose registers R1, R2, and R3. These are used to store intermediate results that are required by the same ACS unit in consecutive clock cycles. For example, this allows the four ACS operations equivalent to a calculation to be performed in four consecutive clock cycles using a single ACS unit, as Fig. 4. Energy-efficient LUT-Log-BCJR architecture. The second register level comprises REG bank 1 and REG bank 2 of Fig. 4, which are used to temporarily store the LUT-Log-BCJR variables between consecutive values of the bit index during the
recursions decoding processes. The REG bank 1 comprises registers for input variables \(C_1\), \(C_3\). Meanwhile, the sets of input \(C_3\), \(C_2\) are stored in REG bank 2 of Fig. 4. The main memory stores all the calculated ACS outputs for the decoding process. Since the proposed architecture supports a fully parallel arrangement of an arbitrary number of ACS units of Fig. 4, it may be readily applied to any LUT-Log-BCJR decoder, regardless of the specific convolution encoder parameters employed. Note that in contrast to the different-length data paths of Fig. 2, the identical parallel data paths shown in Fig.4. have equal lengths, which avoids energy wastage, as described above.

Fig 5. ACS Unit

A. Novel ACS Unit

In this section Fig.5 shows, the novel low-gate-count ACS unit, which performs one ACS operation per clock cycle. The control signals of the ACS unit are provided by the operation code \(O=\{0,1,2,3,4,5\}\), which can be used to perform the functions listed in Table I. The ACS unit is responsible for implementing the state metric computation. A direct implementation requires two identical data paths each with two additions, a comparison, and a selection.

Op 1 In this clock cycle max* the calculation is activated by using the operation code \(O=101100\) of Table I and loading operands \(p^\gamma\) and \(q^\gamma\) from the registers \(R_1\) and \(R_2\) of Fig. 5, respectively. The result \(r^\gamma\) is then stored in register \(R_3\), which is the approximated as \(|R_1-R_2|\). The result \(C_0\) determines max\((R_1,R_2)\).

Op 2 The MAX* calculation of (2) is completed in the fourth clock cycle by using the operation code \(O=000000\) of Table I. Here the operand \(p^\gamma\) is provided by the maximum of \(R_1\) and \(R_2\), as identified by \(C_0\) of Fig.5. Meanwhile, a value for the operand \(Q^\gamma\) is selected from the set \(\{0.75, 0.5, 0.25, 0\}\), depending on the contents of \(C_1\) and \(C_2n\) of Fig. 5. As a result, we have

\[
\bar{r} = \max(R_1,R_2) + \begin{cases} 
0.75 & \text{if } C_1 = 0, C_2 = 0 \\
0.5 & \text{if } C_1 = 0, C_2 = 1 \\
0.25 & \text{if } C_1 = 1, C_2 = 0 \\
0 & \text{if } C_1 = 1, C_2 = 1 
\end{cases}
\]

TABLE I

<table>
<thead>
<tr>
<th>(O)</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000</td>
<td>(\bar{r} = p + q)</td>
</tr>
<tr>
<td>100000</td>
<td>(\bar{r} = p - q)</td>
</tr>
</tbody>
</table>
| 101100 | \(\bar{r} = \begin{cases} 
(p - q) - 0.25 & \text{if } p \geq q \\
0 & \text{if } p < q 
\end{cases} \) |

TABLE II

<table>
<thead>
<tr>
<th>Publication</th>
<th>Proposed</th>
<th>Existing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
<td>LUT-Log</td>
<td>Max-Log</td>
</tr>
<tr>
<td>Block size(bit)</td>
<td>6144</td>
<td>6144</td>
</tr>
<tr>
<td>Area(mm²)</td>
<td>0.35</td>
<td>3.57</td>
</tr>
<tr>
<td>Power consumption(mW)</td>
<td>2</td>
<td>788.9</td>
</tr>
<tr>
<td>Clock Frequency F (MHz)</td>
<td>333</td>
<td>390.6</td>
</tr>
</tbody>
</table>

In Table II, we compare the proposed architecture to the latest LUT-Log-BCJR and Max-Log-BCJR decoder architectures. The area and energy consumptions are estimated based on post-layout simulations. The implementation results arising from different technologies are also scaled to give a fair comparison. As shown in Table II, the energy consumption of the proposed architecture is significantly lower than that of the conventional LUT-Log-BCJR architectures. Furthermore, our proposed architecture has a similar energy consumption to that of the recent Max-Log-BCJR decoders, but facilitates a 10% lower transmission energy, as discussed in Section I.

IV. SYNTHESIS RESULT

The Figures Fig.6.1 , Fig.6.2 , Fig.6.3 and Fig.6.4 of this section are the final output waveform generated using Xilinx tool.

In Fig.6.1 Encoder output waveform along with RCS and Interlever outputs are shown. In Fig.6.2 and Fig.6.3 decoder output waveform along with Upper and Lower LUT-Log BCJR outputs are shown. In Fig.6.4 de-interleaved output and final result of Decoder outputs are shown.
V. CONCLUSION

In this paper, we have proposed low complexity energy-efficient architecture providing low area and low energy consumption by breaking down the LUT log BCJR into its most fundamental ACS operations. In the previous existing architecture i.e conventional LUT log architecture may have wasteful design requiring high chip area leading high energy consumption, where in the proposed architecture consider different turbo codes for encoding and decoding algorithm by validating the architecture using an LTE turbo decoder, to have an order of magnitude of lower area than the existing conventional LUT-log-BCJR architecture and our proposed system provides approximately 72% lower energy consumption compared to max-log BCJR implementation.

REFERENCES


Chandana P
The author is currently pursuing M.tech in VLSI and Embedded System at Sapthagiri College of Engineering, Bengaluru, Karnataka, affiliated to Visvesvaraya Technological University.

Agalya P, Associate Professor in Dept. of Electronic and Communication Engineering, Sapthagiri College of Engineering, Bengaluru, Karnataka.

Shoba B N, Associate Professor in Dept. of Electronic and Communication Engineering, Sapthagiri College of Engineering, Bengaluru, Karnataka.