



All
 Articles
 Products
 Courses
 Papers
 News
 Webinars
 Web

Site Search:

[LOGIN](#)
[REGISTER](#)
 Welcome, Guest
 [Home](#)
[RSS](#)
[Design Center](#)
[Learning Center](#)
[Product Center](#)
[News](#)
[Blogs](#)
[Forums](#)
[Careers](#)
[Site Features](#)




January 13, 2010

Low power LDPC decoder created using high level synthesis

By Yang Sun and Joseph R. Cavallaro, Rice University, Houston, Texas, Tai Ly, Synfora Inc., Mountain View, Calif.

Introduction

With the popularity of mobile wireless devices soaring, the wireless communication market continues to see rapid growth. However, with this growth comes a significant challenge. Many applications, such as digital video, need new high data rate wireless communication algorithms. The continuous evolution of these wireless specifications is constantly widening the gap between wireless algorithmic innovation and hardware implementation. In addition, low power consumption is now a critical design issue, since the life of a battery is a key differentiator among consumer mobile devices. The chip designer's most important task is to implement highly complex algorithms into hardware as quickly as possible, while still retaining power efficiency. High Level Synthesis (HLS) methodology has already been widely adopted as the best way to meet the challenge. This article gives an example in which an HLS tool is used, together with architectural innovation, to create a low power LDPC decoder.

High Level Synthesis Methodology

HLS methodology allows the hardware design to be completed at a higher level of abstraction such as C/C++ algorithmic description. This provides significant time and cost savings, and paves the way for designers to handle complex designs quickly and efficiently, producing results that compare favorably with hand design.

HLS tools also offer specific power-saving features, designed to solve the problems of power optimization. In any design, there are huge opportunities for power reduction at both the system and the architecture levels. HLS can make a significant contribution to power reduction at the architecture level, specifically by offering the following:

- Ease of architecture and micro-architecture exploration

- Ease of frequency and voltage exploration

- Use of high level power reduction techniques such as multi-level clock gating, which are time-consuming and error-prone when done manually at the RTL level

Power-saving opportunities at the RTL and gate-level are limited and have a much smaller impact on the total power consumption.

Low-Density, Parity-Check decoders

Forward Error Correction (FEC) coding, a core technology in wireless communications, has already advanced from 2G convolutional/block codes to more powerful 3G Turbo codes. Recently, designers have been looking elsewhere for help with the more complex 4G systems. A Low-Density, Parity-Check (LDPC) encoding scheme is an attractive proposition for these systems, because of its excellent error correction performance and highly parallel decoding scheme.

Nevertheless, it is a major challenge for any designer to create quickly and efficiently a high performance LDPC decoder which also meets the data rate and power consumption constraints in wireless handsets.

LDPC decoders vary significantly in their levels of parallelism, which range from fully parallel to partially parallel to fully sequential. A fully parallel decoder requires a large amount of hardware resources. Moreover, it hard-wires the entire parity matrix into hardware, and therefore can only support one particular LDPC code. This makes it impractical to implement in a wireless system-on-a-chip (SoC) because different or multiple LDPC codes might need to be supported eventually. Partial parallel architectures can achieve high throughput decoding at a reduced hardware complexity. However, the level of parallelism in these instances has to be at the sub-circulant (shifted identity matrix) level, which makes it code-specific as well and therefore can be too inflexible for the wireless SoC.

This article looks at exploring the design space of scalable parallel realizations of LDPC decoders using a high level synthesis (HLS) methodology. Under the guidance of the designers, HLS can effectively exploit the parallelism of a given algorithm. The article demonstrates how two scalable parallel LDPC decoding algorithms can be implemented by the HLS tool to produce area and power-efficient hardware.

High Level Synthesis

For this example, the high-level synthesis tool used was PICO C-Synthesis tool, which creates application

accelerators from untimed C for complex processing hardware within the video, audio, imaging, wireless and encryption domains.

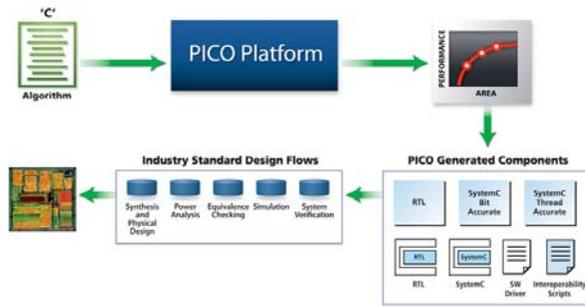


Figure 1: PICO High Level Synthesis
[Click on image to enlarge.](#)

Figure 1 shows the overall design flow for creating application accelerators using HLS. The HLS system automatically generates synthesizable RTL, customized test benches and SystemC models, as well as synthesis and simulation scripts. Its methodology is based on an advanced parallelizing compiler, which finds and exploits parallelism at all levels in the C code, and on a multi-level hierarchical synthesis technology, giving results which compete favorably with manual design.

The Decoding Algorithm

This article focuses on a special class of LDPC codes called block-structured LDPC codes, which have been adopted by many new wireless standards (IEEE 802.11n, IEEE 802.16e). As shown in figure 2, a block-structured parity check matrix can be viewed as a two-dimensional array of square sub-matrices.

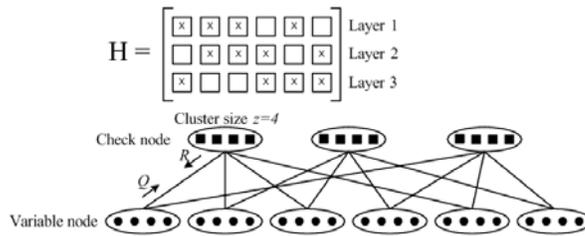


Figure 2: An example of 3x6 block-structured parity check matrix
[Click on image to enlarge.](#)

For this project, the scaled-minsum decoding algorithm is used, with a layered message passing scheme. This algorithm can reduce significantly both memory usage and logic complexity, while still offering a nearly optimal decoding performance.

Architectural Design

We explored two design architectures.

1) Per-layer decoding architecture

One of the key challenges for the task was the absence of literature about the VLSI design of LDPC decoders with scalable parallelism. The parity check matrices vary widely between different wireless standards, so the designers initially had difficulty in finding the right matrices for this particular decoder. In this example, the HLS tool was used to create scalable decoder architectures automatically. By instructing the HLS compiler to unroll loops in a particular sequence, it was possible to realize multiple parallelism levels which could tailor the throughput according to the application requirement.

To implement the algorithm in hardware, a block-serial scheduling algorithm was used, in which the data in each layer was processed block-column by block-column. For this case study, a WiMAX LDPC decoder with a code length of 2304 and 1/2 rate was described using sequential untimed C Code. Figure 3 shows the corresponding HLS generated hardware architecture block diagram. The C code on the left has been mapped onto a Pipeline of Processing Arrays (PPA) template architecture.

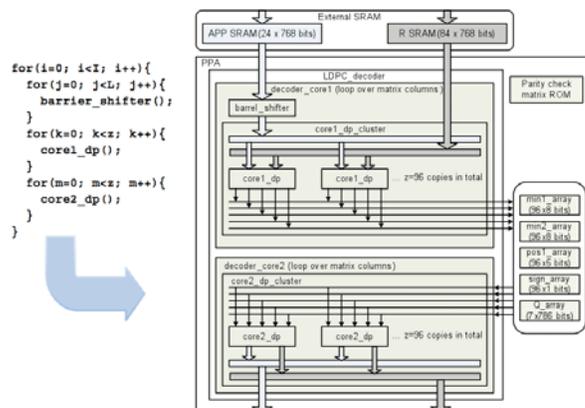


Fig. 3: HLS hardware architecture block diagram for per-layer decoding of a (2304, 1/2) WiMAX LDPC code
[Click on image to enlarge.](#)

The design determines the following steps:

The top level LDPC decoder() will loop over I iterations
 During each iteration, it loops over L layers of the parity check matrix, and calls decoder-core1() and decoder_core2()
 The for loops in both decoders are unrolled
 The for loop in barrel_shifter is also unrolled to shift P message as an array of 8-bit
 If all parity checks are satisfied, or the maximum number of iterations is reached, the top level function can return early

2) Multi-layer Pipelined Decoder Architecture

It is also possible to do pipelined processing between layers, using additional conflict detection logic. The per-layer decoding architecture described above, in which each core is independent because there is no data dependence between them, only uses about 50 percent of its capacity. Multi-layer architecture, however, allows one core to operate on the current layer while the other begins work on the next layer of the matrix. This architecture is similar to the per-layer variant, but each core has its own copies of arrays, and the core utilization is now almost 100 percent.

Figure 4 compares the latency and area of these two architectures. In the analysis, RTLs for both designs are generated by the HLS tool, and are synthesized using Synopsys Design Compiler on a TSMC 65nm technology. Note that the area value shown in **Figure 4** is the total standard cell area. This gives a fair comparison because the two architectures require the same amount of external SRAMs. The diagram shows that both latency and area increase as clock frequency increases. This is expected, as the HLS tool adjusts the design to find the best solution for a given target clock frequency. The two graphs also show that the two-layer pipelined design gives almost twice the performance of the per-layer design at a cost of only about 20-25 percent more area.

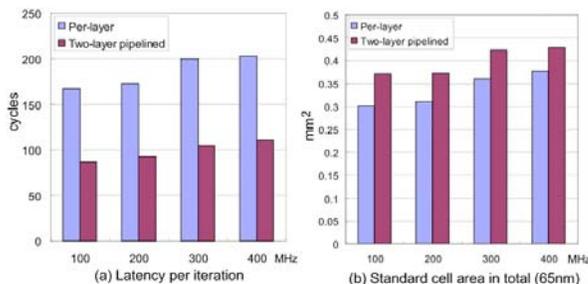


Fig. 4. Latency and area (65nm) comparisons of two hardware architectures synthesized using HLS for different target clock frequency goals
[Click on image to enlarge.](#)

Low Power Implementation via Clock-Gating

Hardware generated by this particular HLS tool is inherently power-aware. Moreover, it allows power to be significantly decreased through the use of architecture-level clock-gating.

The automatic, multi-level clock-gating technique offered by the HLS tool enables a designer to optimize power at the system level, eliminating all manual work. The tool provides block-level clock-gating, shutting off the clocks to entire processing blocks at any level of the hierarchy to minimize power at an architectural level. A designer can use directives to specify where to insert clock-gating, and leave the rest to be done automatically. The clock-gating feature also has the added benefit of allowing the designer to make changes at any time without having to impact the algorithm or the code. In traditional RTL design, designers can only insert clock-gating at a block level if they already know when the block will be inactive, which requires significant manual analysis. For this design, the HLS tool built a clock-gating infrastructure which could turn off complete blocks at the top level of the design, using control logic to indicate precisely when each block could be left idle.

Table 1 below compares the power consumption of a pipelined LDPC decoder with and without clock-gating (external SRAMs are not included in the analysis). A 29-percent reduction in "sequential internal power" and a 20-percent reduction in total power consumption was achieved by using multi-level clock-gating feature of the HLS tool. These power savings are in addition to the power savings given by register-level clock gating.

| Power | Leakage | Internal | Switching | Total |
|------------------|---------|----------|-----------|--------|
| W clock-gating | 3.43mW | 46.1mW | 22.5mW | 72.0mW |
| W/O clock-gating | 3.43mW | 64.5mW | 22.5mW | 90.4mW |

Table 1: Power estimates with and without multi-level clock gating
[Click on image to enlarge.](#)

Results

An LDPC decoder supporting the IEEE 802.16e WiMAX standard was implemented in untimed C. The HLS tool was then used to produce Verilog RTL for both the per-layer and the two-layer pipelined architectures. **Table 2** below compares the current design to a manual design*. Since the two designs operate at different clock frequencies, the table shows the normalized area, power and performance numbers. The current design achieves significantly better performance at less than half the power consumption and the same area as the manual design.

| | The Current Design | Manual Design |
|---|---------------------------|---------------------------|
| Max Frequency | 400 MHz | 240 MHz |
| Max Code Length | 2304 | 1944 |
| Core Area | 0.003mm ² /MHz | 0.003mm ² /MHz |
| Max Power | 0.45 mW/MHz | 0.98 mW/MHz |
| Throughput for max code length at code rate 1/2 | 1.04 Mbps/MHz | 0.74 Mbps/MHz |

Table 2: Design comparison with an existing LDPC decoder
[Click on image to enlarge.](#)

These results are achieved both by architectural innovation and HLS tool utilization.

Conclusion

The chip designer's challenge today is to rise to the consumer demand for increasingly complex wireless devices, without compromising on power consumption. New ways of looking at chip architecture, and new tools to create it, can provide the solution. In this example, HLS methodology with multi-level clock-gating for low power design and innovative architectures were the keys to achieving a low power, LDPC decoder for today's consumer mobile device market.

* M. Rovini, G. Gentile, F. Rossi, and L. Fanucci, "A Scalable Decoder Architecture for IEEE 802.11n LDPC Codes," in GLOBECOM, 2007, pp. 3270-3274.



ACE AWARDS EE Times

CALL FOR ENTRIES
DEADLINE IS APPROACHING!

CLICK HERE ENTER TODAY!
 Entry Deadline
 January 22, 2010

2010 EE Times ACE Awards
 EE Times presents The 6th Annual
 EE Times ACE Awards

SILVER SPONSOR **XILINX** MEDIA SPONSOR **spectrum** **EETimes**

TechOnline Communities

Audio DesignLine | Automotive DesignLine | CommsDesign | Digital Home DesignLine | DSP DesignLine | EDA DesignLine
 eeProductCenter | Embedded Internet | **Green SupplyLine** | Industrial Control DesignLine | Mobile Handset DesignLine | Planet Analog | Power Management DesignLine
 Programmable Logic DesignLine | RF DesignLine | Teardown.com | TechOnline | Video/Imaging DesignLine | Wireless Net DesignLine

EE Times

United States | Asia | China | Europe | France | Germany | India | Japan | Korea | Taiwan | United Kingdom | EE Times Supply Network

Additional Network Sites

Analog Europe | Automotive Designline Europe | DeepChip | Design & Reuse | Elektronik iNorden | Embedded.com
 | Microwave Engineering Europe | Portelligent | Power Management Designline Europe
 The RF Edge | Semiconductor Insights | Embedded-europe.com | Mechanical Design | TechOnline India

All materials on this site [Copyright © 2010 TechInsights, a Division of United Business Media LLC](#). All rights reserved.
[Privacy Statement](#) | [Terms of Service](#)