Implementing a 2-Gbs 1024-bit ½-rate Low-Density Parity-Check Code Decoder in Three-Dimensional Integrated Circuits*

(Invited Paper)

Lili Zhou, Cherry Wakayama, Robin Panda, Nuttorn Jangkrajarng, Bo Hu, and C.-J. Richard Shi
University of Washington
cjshi@u.washington.edu

Abstract

A 1024-bit, ½-rate fully parallel low-density parity-check (LDPC) code decoder has been designed and implemented using a three-dimensional (3D) 0.18μm fully depleted silicon-on-insulator (FDSOI) CMOS technology based on wafer bonding. The 3D-IC decoder was implemented with about 8M transistors, placed on three tiers, each with one active layer and three metal layers, using 6.9mm by 7.0mm of die area. It was simulated to have a 2Gbps throughput, and consume only 260mW. This first large-scale 3D application-specific integrated circuit (ASIC) with fine-grain (5μm) vertical interconnects is made possible by jointly developing a complete automated 3D design flow from a commercial 2-D design flow combined with the needed 3D-design tools. The 3D implementation is estimated to offer more than 10x power-delay-area product improvement over its corresponding 2D implementation. The work demonstrated the benefits of fine-grain 3D integration for interconnect-heavy very-large-scale digital ASIC implementation.

1. Introduction

Low density parity check (LDPC) codes are emerging as standard means of channel encoding and error correction for wireless communication, data storage, optical communication, and quantum computing. This is due to the near Shannon-limit error correction performance of LDPC codes [1] and the progress in semiconductor fabrication technologies that allow very large scale integration of circuit functionality [2].

* This work was supported by the U.S. Defense Advanced research Projects Agency (DARPA) 3-D IC Program under Grant 66001-05-1-8918, monitored by Navy SPAWAR Systems Center, San Diego, CA, USA.

One great advantage of these codes over other codes such as turbo codes is the absence of a serial dependency in the belief-propagation decoding algorithm, which can be fully parallelized and potentially implemented for high throughput systems in next generation communication [3][4], military and space applications [5][6].

However, a fully parallel architecture LDPC decoder implementation has the following design challenges [2]:

(1) The average wire length can be 3mm for a 7.5mm x 7.0 mm die implementation (half of the die size). Therefore, specific CAD tools are required.

(2) The wiring takes more silicon area and power dissipation than the logic itself.

(3) Only 50% area utilization for logic was achieved due to routing congestion.

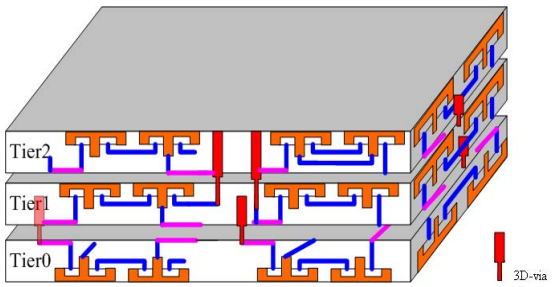

Fig. 1. Cross-section view of 3-tier 3D integration.

To address this interconnect design challenge, we explore the use of a three-dimensional integrated circuit (3D-IC) process. Placing and wiring devices in the third dimension to shorten the long global interconnections promises higher clock rates, less power dissipation, and higher integration density. In general, 3D-ICs are integrated circuits consisting of active devices not confined to a single planar layer [7][8][9]. It can be described as stacking several

integrated circuits vertically with fine-grade third dimensional interconnects, called 3D-vias. Fig. 1 is an illustration of the cross section of three tiers that form a 3D integrated circuit, where each tier consists of a layer of active devices connected with multi-layers of metal layers (as in a conventional two-dimensional integrated circuit). Additional benefits of 3D integration include (1) easy integration of circuit intellectual properties (IPs) fabricated with different technologies, and (2) better overall system-on-chip (SOC) performance and less cost, where analog and radio frequency (RF) functionality can be integrated on one tier, where logic and memory circuitries on other tiers. Despite several theoretical studies of the benefits of 3D integration for high-performance application-specific integrated circuits (ASICs), all the previous 3D IC designs are limited either to simple logic devices, or to circuits of regular structures such as photo sensors, memories and field-programmable gate arrays (FPGAs).

In this paper, we present a 3D implementation of 2Gb/s throughput 260mW 1024-bit ½-rate fully parallel LDPC code decoder consisting of 8 millions of transistors using MIT-Lincoln Lab's 3-tier 0.18µm FDSOI 3D process. We have developed a standard super-cell design methodology and supporting CAD tools to facilitate the 3D design process. We have also implemented the same design in the same FDSOI technology using the corresponding 2D process. The comparison has shown that 3D implementation has an order of magnitude improvement in terms of power-delay-area product over the corresponding 2D implementation. It is predicted that the 3D-via density will increase with the progress in 3D integration technologies. Therefore, as 3D processes become more mature, the 3D IC integration will be promising to implement complicated interconnect-heavy applications such as l00Gb/s throughput LDPC decoders.

This paper is organized as follows. Section 2 reviews the LPDC code, belief-propagation decoding algorithm and the fully-parallel LDPC decoder architecture. Section 3 presents the details of the 3D LDPC decoder implementation. Section 4 shows the implementation results of the 3D design and its comparison with the 2D design. Section 5 concludes the paper.

2. LDPC codes and decoder architecture

2.1. LDPC codes and the message-passing (belief-propagation) algorithm

An LDPC code is often described by a sparse binary parity-check matrix H. Any decoded message vector x is a valid codeword if it satisfies $H \cdot x = 0$. Each row of matrix H corresponds to a parity check and each column represents a demodulated symbol. The number of demodulated symbols N is the LDPC code length. An example 5×10 H matrix of a 10-bit ½-rate regular LDPC code is illustrated in Fig. 2.

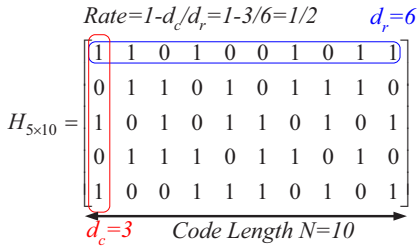

Fig. 2. An example H matrix of 10-bit ½-rate LDPC code.

We often use Tanner graphs to represent the LDPC codes. This bipartite graph expresses how a global function of many variables is factored into a product of local functions [10]. Fig. 3 shows the corresponding Tanner graph of the LDPC code example in the Fig. 2. In the graph, there are two sets of nodes: one set is called *variable nodes*, which correspond to the columns of the parity check matrix, and the other set is called *check nodes*, which correspond to the rows of the H matrix. Each edge between the variable node and check node corresponds to a "1" in the H matrix. Therefore, each variable node represents one bit in the codeword and each check node represents one parity check equation. The edge represents the message passing between these two sets of nodes.

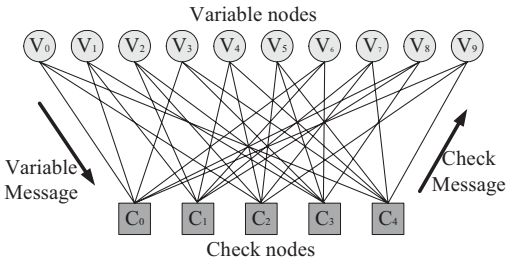

Fig. 3. The Tanner graph for the LDPC code example.

The decoding problem is to find the most probable vector x such that $H \cdot x = 0$. One popular LDPC decoding algorithm is the improved message-passing

algorithm – belief propagation (BP) algorithm [11]. The belief propagation algorithms are often re-formulated as the Log-BP algorithms [12], in which multiplication operations are converted to addition operations to reduce the computational complexity. The message exchanged between variable nodes and check nodes are represented as the log-likelihood ratio (LLR). The detailed Log-BP algorithm is described below [2]:

Step 0: All variable nodes and outgoing messages are initialized to the values of the intrinsic probabilities for every bit from a demodulator γ_i.

$$\gamma_i = \log\left[\frac{P(x_i = 0 \mid y_i)}{P(x_i = 1 \mid y_i)}\right] \quad (1)$$

where y_i is the received symbol and x_i is the transmitted symbol.

Step 1: Propagate messages from the variable nodes to the check nodes along the edges of the graph.

Step 2: Perform a parity check on the incoming messages at the check nodes. The following check node computation is performed:

$$\lambda_{CV} = \mathrm{sgn}\Big(\prod_{j=1}^{d_r-1} \lambda_{V_jC}\Big)\Psi^{-1}\Big(\sum_{j=1}^{d_r-1} \Psi(\lambda_{V_jC})\Big) \quad (2)$$

where d_r is the degree of the check node C (row weight in the H matrix), λ_{V_jC} represents the incoming message from neighboring variable node $V_j \neq V$ to check node C, and λ_{CV} is the outgoing message from check node C to the variable node V. Function Ψ is equal to Ψ^{-1}, which is expressed in the following equation:

$$\Psi(x) = \Psi^{-1}(x) = \log\left\{\frac{1+\exp(-|x|)}{1-\exp(-|x|)}\right\} \quad (3)$$

Step 3: Pass messages from the check nodes back to the variable nodes along the edges of the graph.

Step 4: At the variable nodes, update the estimation of the decoded bit and outgoing messages for each edge connected to the variable. The following variable node computation is performed:

$$\lambda_{VC} = \sum_{i=0}^{d_c-1} \lambda_{C_iV} \quad (4)$$

where λ_{C_iV} is the incoming message from the neighbor check node $C_i \neq C$ to variable node V and d_c is the number of check nodes connected to V (column weight in the H matrix), and λ_{VC} is the outgoing message from variable node V.

Step 5: When the variable node computation is completed, the LLR of each symbol *i* is updated as:

$$\Lambda_i = \gamma_i + \sum_{i=0}^{d_V-1} \lambda_{C_iV} \quad (5)$$

From the updated LLR vector $\Lambda = \{\Lambda_1, \Lambda_2, ...\Lambda_i, ...\Lambda_N\}$, a hard decision result $X = \{x_1, x_2, ...x_i, ...x_N\}$ is calculated as:

$$x_i = \begin{cases} 1, & if\ \Lambda_i \le 0 \\ 0, & if\ \Lambda_i > 0 \end{cases} \quad (6)$$

The calculated hard decision vector X is then checked against the parity check matrix H. A case of $H \cdot X = 0$ means the iterative process has converged to a codeword and decoding stops. Otherwise, steps 1-4 are repeated until $H \cdot X = 0$ or a fixed number of iterations are reached.

2.2. The fully parallel architecture of LDPC decoder

The message-passing algorithm maps extremely well to the fully parallel decoder architecture, in which the Tanner graph representation of the H-matrix is directly instantiated in hardware [2]. Each of these variable nodes and check nodes are implemented as a Variable Node Unit (VNU) and a Check Node Unit (CNU). The connections among them are routed directly as defined by the H-matrix.

Our H matrix has 3072 non-zero entries. To map the code performance accurately in hardware implementation, the functions (*log*, *atanh*, *exp*, *tanh*) of real numbers are quantized by lookup tables. Experimental simulation results show that 4 bits are required to quantize the message link between VNUs and CNUs without incurring substantial performance degeneration.

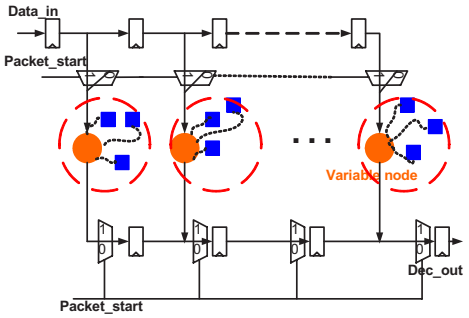

Fig. 4. The three-stage pipeline structure.

Registers are inserted at the VNU output to synchronize the execution of the belief-propagation algorithm. In our design, we set the maximum iteration number to be 64 in the decoding procedure. Since each decoding process needs 64 iterations, which is 64 clock

cycles, if we fully use this time to load the new packet data for the next decoding and load out the decoded bit, the data could be processed continuously.

The three-stage pipeline structure is shown in Fig. 4. One stage is the data being loaded into the decoder, the second is the iterative decoding process and the third is the decoded bit being loaded out the decoder. Every 64 VNUs are grouped into one line and 1024 VNUs are partitioned into 16 pipelines (64*16=1024). So during the 64 iterations (clocks), the new 64 data are shifted in through the loading registers and 64 decoded bits are shifted out through the unloading registers. The signal "Packet_start" is used to control these three-stage pipelines.

2.3. The Logic of VNU and CNU

Based on [2], the basic structures of the VNU and CNU are shown in Fig. 5 and 6, respectively.

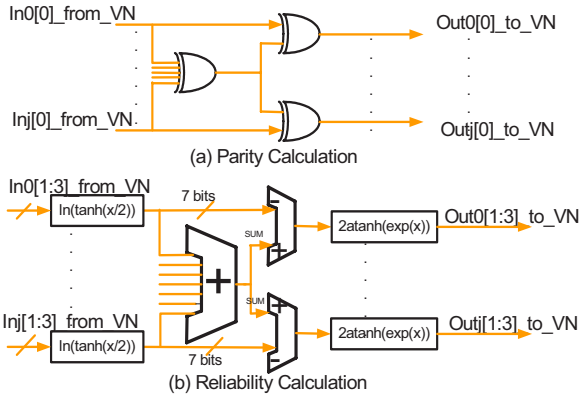

Fig. 5. The block diagram of the check node (CNU).

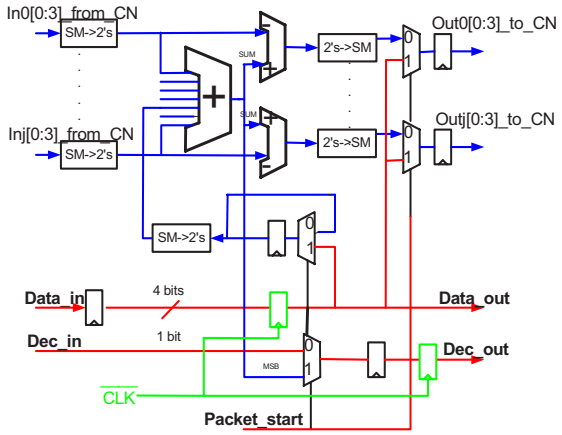

Fig. 6. The block diagram of the variable node (VNU) with pipeline registers.

Each check node performs a parity check across all variables in a row of H matrix. The check node design can be separated into the parity calculation part and reliability calculation part. The messages propagated between the variable nodes and check nodes are 4-bit sign-magnitude binaries in this design. The parity calculation takes the sign bit and the reliability calculation takes the other three magnitude bits. The calculation is performed in the log domain to avoid the multiplication and division operations according to the log-MAP algorithm. The hyperbolic trigonometric function in Eq.(5) is realized by the lookup tables. The variable node contains the decoding message registers and the pipeline registers. When the control signal Packet_start is "1', the new packet data for decoding are loaded in and the previous decoded bit is loaded out. When the Packet_start is "0", it means that current decoding process is not finished. The messages for check nodes go through the message registers for each iteration and the new packet data for next decoding go through the pipeline registers to the other variable nodes that are grouped in the same line. The gray lines in the variable node diagram show the datapath for pipeline and the black lines show the combinational logic in the variable node.

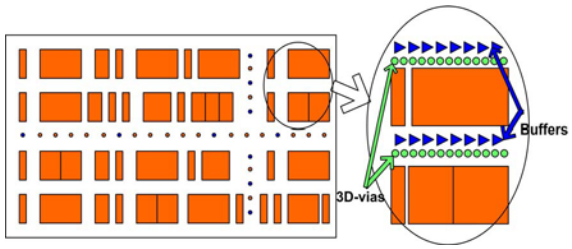

Fig. 7. Proposed "standard super-cell" layout scheme.

3. 3D LDPC design

We have developed a complete 3D design flow and methodology – standard super-cell based 3D ASIC design flow with corresponding CAD tools [13].

The target 3D technology is called "wafer bonding". The individual wafers in a 3D IC are fabricated using conventional means and fused together with inter-wafer electrical and mechanical interconnects. Wafer bonding methods differ in terms of the bonding material and the order of fabrication operations. For instance, the MIT-LL uses parallel oxide bonding approach [8]. The individual wafers are processed before bonding. Formation of inter-wafer interconnects is the remaining back-end step. Inter-wafer interconnects, referred to as ***"3D-vias"*** in

this paper, are etched through the entire metallization stack on the top of each wafer. The size and density of 3D-vias are determined mainly by the alignment requirements and via formation process. For MIT-LL's 3-tier 0.18µm FDSOI 3D process, each 3D-via occupies 1.75µm × 1.75µm, and the minimum distance between any two 3D-vias is from 1.8µm to 5µm depending on the 3D-via density required to meet the stringent alignment constraints. Compared to traditional multi-chip module (MCM) and system-in-package (SiP) technologies, 3D IC offers much finer-grain 3D interconnects, and thus offers much greater potential for improving system performance and integration density.

On the other hand, a 3D-via still uses much bigger silicon area than a conventional inter-metal via, introduces more delays, and causes the fabrication and reliability problem. Specific 3D-via placement strategy should be performed not only to satisfy stringent aliment constraints between neighboring tiers but also to reduce the routing density and achieve more uniform 3D-via distribution. Furthermore, whenever possible, the number of 3D-vias used should be reduced.

With these observations and with the target to implement interconnect-heavy very large-scale integrated circuits, we have developed the following "*standard super-cell*" based layout scheme. As illustrated in Fig. 7, *standard* super-cells are layout macros with the *same height and varying widths*. Standard super-cells are placed in rows. Spaces are reserved between standard super-cell rows for 3D-vias and buffers. Using this scheme, the 3D-via alignment constraint can be easily satisfied and does not affect super-cell placement. Buffer insertion is also less constrained with respect to the silicon resource; therefore both the 2D long wires and 3D long wires crossing different tiers can be handled easily. To accommodate routing congestion for interconnect-heavy systems and especially for the MIT-LL's 3-tier 0.18µm FDSOI 3D CMOS process where only 3 metal layers are available per tier, vertical routing channels are also reserved.

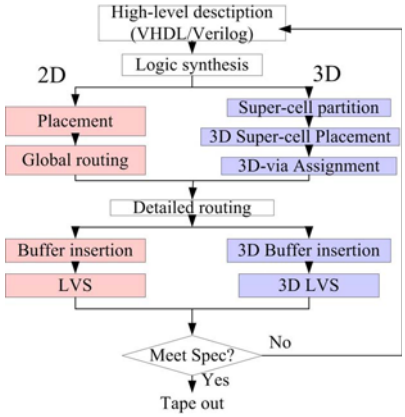

Fig. 8. Simplified flowchart for the automated design of 2D and 3D ASICs.

With this proposed 3D standard super-cell layout scheme, 3D design can be accomplished with a flow similar to typical 2D design, as shown in Fig. 8. For 3D design, a design is first partitioned into maximum of a

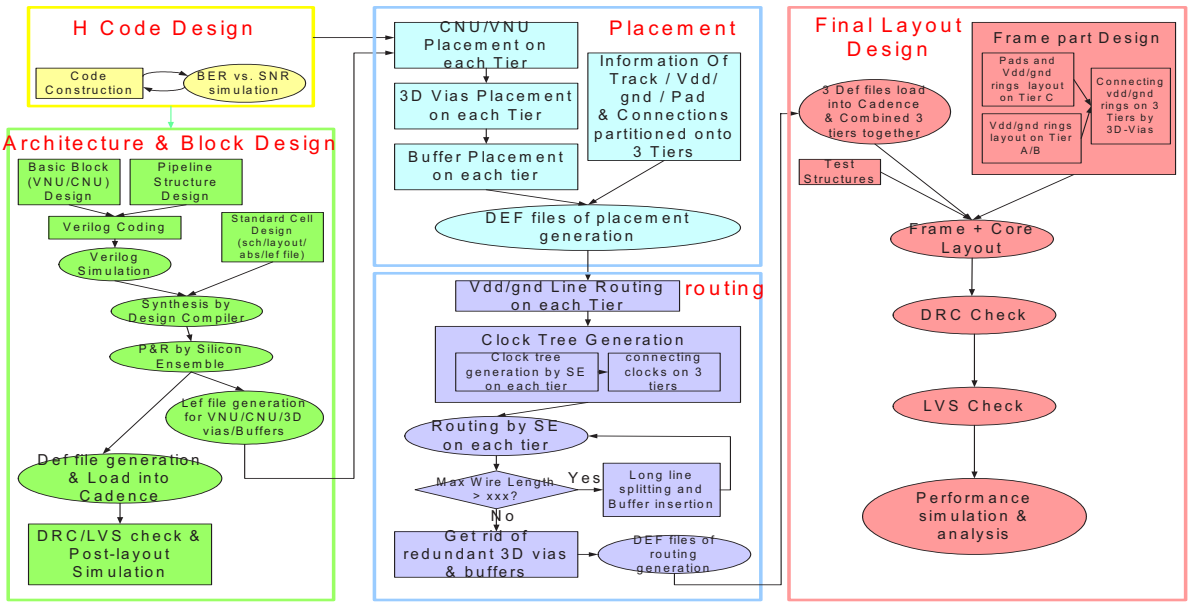

Fig. 9. 3D LDPC design flow.

few thousands of blocks. Then, for each block, traditional 2D logic and physical design tools are used to synthesize super-cells with the same height and varying width. The super-cell height is determined heuristically with consideration of chip aspect ratio and area constraints. Next, a 3D placer finds an optimal 3D super-cell placement. Once 3D placement of super-cells is fixed, the number of 3D-vias to be used is determined. Then 3D-via assignment is performed to assign 3D-vias to the reserved 3D-via locations (as shown in Fig. 7) with the objectives to reduce the 3D-via distribution density and the routing density on each tier. After 3D-via assignment, commercial 2D detailed routing tools are used for routing on each tier. Since super-cells are distributed over several tiers, tools for 3D buffer insertion and clock distribution are developed to handle long interconnects across tiers. Finally, 3D layout-vs-schematic (LVS) verification is performed to ensure the consistence between the final physical design and the original circuit netlist. 3D-specific tools have been developed including a 3D placer, a 3D-via assignment tool, a 3D buffer insertion and clock distribution tool, and finally a 3D LVS tool. Fig. 9 shows the detailed steps of our 3D LDPC design procedure.

In our standard super-cell based congestion driven 3D placer, a congestion-driven simulated annealing based 3D placement algorithm has been proposed, which allows us to target multiple design objectives such as total wire length, maximum wire length, 3D-via numbers and routing congestion. Routing congestion and interconnect wire length are the main difficulties for LDPC decoder hardware design as we introduced in Section 1. To efficiently analysis the 3D routing congestion, we extend the 2D probability based routing congestion estimation method to 3D. Also the proposed fast routing congestion calculation method makes the possibility of inclusion the congestion metrics within the 3D placement. With the number of super-cells in the order of thousands, simulated annealing is capable of finding high-quality solutions efficiently to meet all design specifications. After the super-cell placement, we assign 3D-via locations, which optimize the routing congestion.

4. Results and 2-D comparison

We have implemented a fully parallel architecture of a 512 × 1024 LDPC decoder using a MIT Lincoln Lab's 3-tier 0.18μm FDSOI 3D CMOS process. The final chip size is 7.04mm × 6.86mm.

This LDPC decoder is partitioned into 1,536 super-cells of two types: check nodes and variable nodes. A check-node super-cell is implemented by an average of 812 (509 ~ 1121) standard cells depends on different input size, and a variable-node super-cell is implemented by 339 standard cells. There are 24,576 nets connecting these super-cells. Table 1 shows the statistics of detailed gates and physical area in the super-cell design.

Table 1. Statistics of super-cell design.

Total no. of gates	509	812	867	1121	339
block area (mm x mm)	0.01919	0.02850	0.03419	0.04293	0.01574
SE auto layout (mm x mm)	0.02864	0.04336	0.05664	0.06912	0.0224
Area utilization (%)	67.02	65.72	60.35	62.11	70.2812
Block Count	18	477	16	1	1024
Total Block Area (mm x mm)	0.34550	13.5927	0.54692	0.04293	16.1206

Table 2. Statistics of final layout on wirelength.

	Total Wire Number	Total Reg. Signal WL (m)	Avg. Reg. Signal WL (mm)	Max Reg. WL (mm)	Buffers used	Buffers used for 3D-vias	Total WL (m)
TierA	189,089	22.39	0.11841	4	3,810	2671	23.372543
TierB	212,935	22.57	0.10599	4.17	6,405	5319	23.552543
TierC	190,676	22.46	0.11779	4.76	3,790	2641	23.442543

The final layout view of the three-tier design, connected by the densely distributed 10,631 3D-vias for signals and 22,960 3D-vias for power-ground lines, is shown in Fig. 10. Table 2 shows the statistics of detailed wire length, 3D-vias and buffers in the final layout design.

To verify the functional correctness and error-correction capability of the LDPC decoder, the relationships of the BER (Block Error Rate) vs. SNR (Signal to Noise Ratio), as well as the decoder convergence iteration vs. SNR, were simulated using Cadence's Verilog-XL simulator. The results are shown in Fig. 11 and 12. We used a LDPC encoding and channel transmission software package [14] developed by Neal at the University of Toronto to perform BER simulations to obtain initial LDPC code. For BER simulations, different levels of noise were added into the original message and transmitted through an Additive White Gaussian Noise (AWGN) channel and the number of information bits that are not corrected through the decoder was counted. With the simulated clock speed at 128MHz, the throughput of the LDPC is 16 b/s * 128 MHz = 2 GB/s.

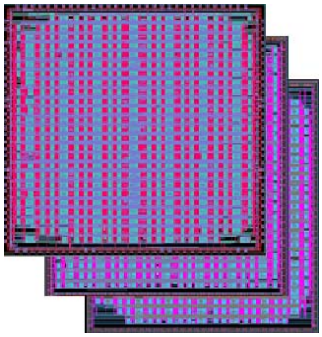

Fig. 10. Final layout view of 3D LDPC structure.

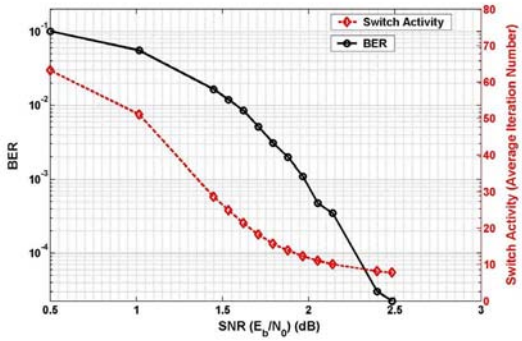

Fig. 11. The simulated LDPC decoder performance.

Table 3 summarizes the layout statistics of the 2D and 3D implementations. The 2D design is accomplished by placing all the devices on one tier using the same technology and the same number of standard super-cells as in the 3D implementation. We can see that the 3D implementation achieves a significant advantage over the same 2D implementation in terms of wire length, area, clock skew, and buffer used. The improvement in terms of area-delay-power product is about an order of magnitude ($2.5 \times 1.75 \times 2.5 = 10.9$).

Fig. 12 plots the power consumption results of post-layout netlist, obtained by running Cadence power analyzer [15]. It shows a significant power reduction with increasing switching activities (required for the low SNR channel decoding).

Fig. 13 (a)(b)(c)(d) show, respectively, the wire length distributions of the 2D design and the 3D design before and after buffer insertion, respectively. From the statistics, we can see that 3-tier 3D integration has changed the wire length distribution from what similar to the Normal distribution (as shown in Fig. 13(a) and (b)) to an exponentially decayed distribution. We note that even the previously reported 2D design with 6-layer's metal showed a similar distribution of wire length as what we have observed. This is a significant result considering for deep-submicron designs, both delay and power are more functions of wire lengths than that of logic gates. And also shorter wires lead to less parasitics variability and more predictable timing, which also means better yield.

Table 3. The comparison between 3D and 2D designs.

	2D design	3D design
Area (mm*mm)	18.238*15.92 =**290.35**	(6.4*6.227)*3 = **119.56**
Total wire length (m)	**182.42**	22.39+22.57+22.46 =**67.42**
Max WL before buffer insertion (mm)	**13.82**	**8.68**
Max WL after buffer insertion (mm)	**4**	**4**
Buffer used	**32900**	**24636**
Clock skew (ns)	**2.33**	**1**
Power dissipation (mw)	**646.2**	**260.2**

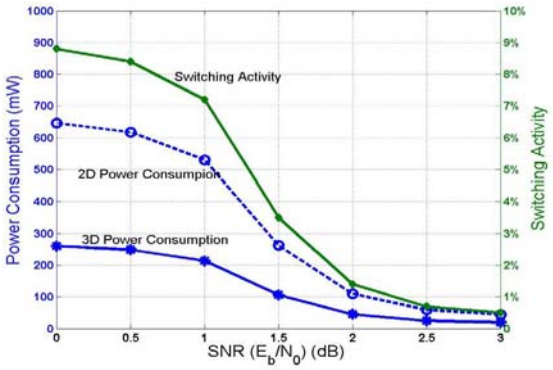

Fig. 12. Post-layout power of the LDPC decoder (2D vs 3D).

5. Conclusion

A fully-parallel LDPC decoder has been implemented on a 3-tier 3D IC process with 2 Gb/s throughput and 260mW power consumption. The significance of this work is three-fold: (1) it is among the first large-scale 3D ASIC implementation and also the first 3D LDPC decoder exploring fine-grain vertical interconnection. (2) It is for the first time, by real silicon tape out and simulation, 3D IC process with 3-tier integration was shown to yield an order of magnitude improvement over the corresponding 2D

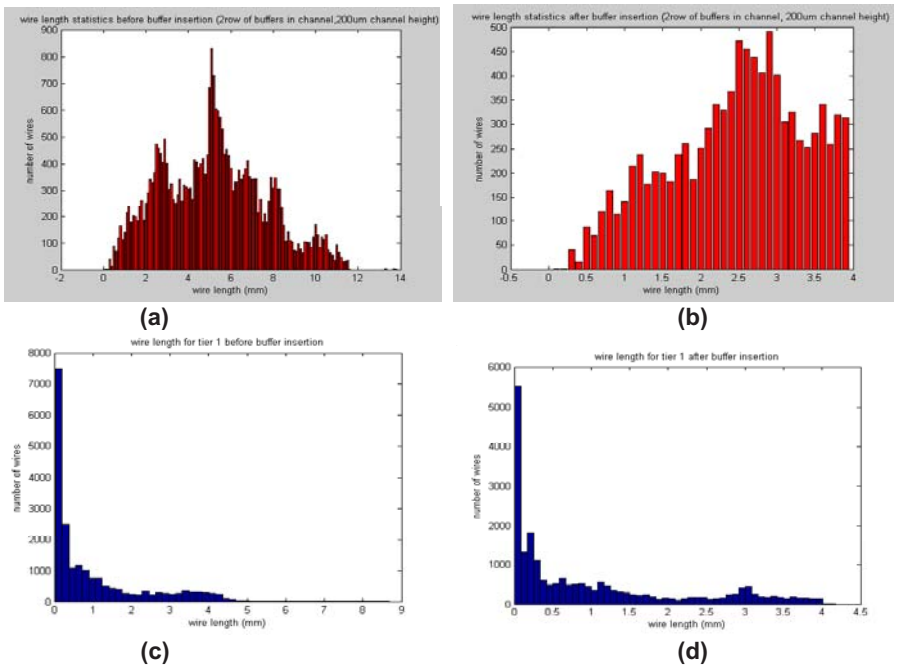

Fig. 13. (a) interconnect wire length distribution for 2D design before buffer insertion. (b) interconnect wire length distribution for 2D design after buffer insertion. (c) interconnect wire length distribution for 3D design before buffer insertion. (d) interconnect wire length distribution for 2D design after buffer insertion.

process, in terms of power-delay-area product.. (3) It is for the first time that an automated 3D design flow has been developed and used to tape out a large-scale silicon ASIC design.

ACKNOWLEDGMENT

The authors wish to thank the 3D-IC group at MIT Lincoln Lab for their contribution in 3D fabrication. The authors are grateful to Dr. Daniel Radack from DARPA/MTO for his vision and support of this research.

References

[1] D. Mackay and R.M. Neal, "Near Shannon limit performance LDPC Codes," *Electron. Lett.*, vol. 32, no. 18, pp. 1645, Aug. 1996.

[2] A. Blanksby and C. J. Howland, "A 690-mW 1-Gb/s 1024-b, rate ½ low-density parity-check code decoder," *IEEE J. Solid-State Circuits*, vol. 37, no. 3, pp. 404 – 412, Mar. 2002.

[3] I.B. Djordjevic and B. Vasic," 100-Gb/s Transmission Using Orthogonal Frequency-Division Multiplexing," *IEEE Photonics Technology Lett.*, vol. 18, no. 15, pp. 1576 – 1578, Aug. 2006.

[4] I.B. Djordjevic, S. Sankaranarayanan, S.K. Chilappagari, and B. Vasic, "Low-density parity-check codes for 40-Gb/s optical transmission systems," *IEEE J. Selected Topics in Quantum Electronics*, vol. 12, no. 4, pp. 555 –562, Jul.-Aug. 2006.

[5] [Online]. Available: http://ipinspace.gsfc.nasa.gov/

[6] [Online]. Available: http://www.grc.nasa.gov/WWW/hrdd/portfolio/bn/nextgenhrchannelcodingschemes%20.htm

[7] K. Banerjee, S. Souri, P. Kapur, and K. Saraswat, "3-D ICs: A novel chip design for improving deep-submicro-meter interconnect performance and systems-on-chip integration," in *Proc. IEEE*, vol. 89, no. 5, May 2001, pp. 602 – 633.

[8] J. Burns, L. McIlrath, J. Hopwood, C. Keast, D. P. Vu, K.Warner, and P. Wyatt, "An SOI-based three dimensional integrated circuit technology," in *Proc. IEEE International SOI Conference*, Oct. 2000, pp. 20 – 21.

[9] J.U. Knickerbocker, C.S. Patel, P.S. Andry, C.K. Tsang, L.P. Buchwalter, E.J. Sprogis, H. Gan, R.R. Horton, R.J. Polastre, S.L. Wright, and J.M. Cotte, "3-D silicon integration and silicon packaging technology using silicon through-vas," *IEEE J. Solid-State Circuits,* vol. 41, no. 8, pp. 1718 – 1725, Aug. 2006.

[10] R. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Info. Theory*, vol.27, no. 9, pp533 – 547, Sep. 1981.

[11] R. Gallager, "Low-density parity-check codes," *IRE Trans. Info. Theory*, vol. 7, no. 1, pp. 21 – 28, Jan. 1962.

[12] M. Chiani, A.Conti, and A.Ventura, "Evaluation of low-density parity-check codes over block fading channels," in *Proc. IEEE International Conference on Communications*, 2000, pp. 1183 – 1187.

[13] L. Zhou, C. Wakayama, and C.-J. R. Shi, "CASCADE: A standard super-cell design methodology with congestion-driven placement for three-dimensional interconnect-heavy very large scale integrated circuits," *IEEE Trans. Computer-Aided Design*, Issue 7, Vol. 26, pp1270-1282, July 2007.

[14] R.M. Neal. Software for low density parity check (LDPC) codes. [Online]. Available: http://www.cs.toronto.edu/~radford/ldpc.software.html.

[15] Power Analyzer, Version 5.4.122, Cadence Design Systems, April 10, 2003.