FPGA Implementation of MIMO System using Xilinx System Generator for Efficient Hardware/ Software co-design

Sparsh Mittal* and Saket Gupta

**Corresponding Author Address: Department of Electrical and Computer Engineering*
Iowa State University, Ames, Iowa, USA.
Email: sparsh0mittal@gmail.com

Abstract- **To meet rapidly increasing demands for the mulAbstract- To meet rapidly increasing demands for the multimedia services and better quality of service over wireless communication systems, MIMO (Multiple Output Multiple Input) systems have been proposed which promise enhanced system capacity, throughput and data reliability. Moreover, the enormous computational requirements for multimedia data call for use of efficient hardware platforms over and above software platforms. In this work, we implement a transceiver system on hardware test-beds, which is useful in video streaming. We employ the 2*2 Alamouti MIMO technique to develop a transmission system and implement the design on FPGA using Xilinx System Generator (XSG) and AccelDSP (for supporting XSG). The state-of-art design tools and methodology lead to superior performance over traditional approach. This reduces the development time and cost and also enhances the reliability. We discuss the salient features of user-friendly model-based design, which enables creation of reusable models and also provides higher level abstraction. A comparison with traditional design methodology using HDL (hardware description language) shows effectiveness of our approach.**

Keywords- **FPGA (Field Programmable Gate Arrays); STBC (Space Time Block Coding), MIMO on FPGA; Xilinx System Generator; AccelDSP; hardware/software co-design; simulation**

1. INTRODUCTION

E-learning and multimedia services to provide online educational material (such as videos, lectures) has been considered a major and fundamental agent of transformation in the realm of distance education. Many institutes have opened their web servers for free lecture-on-demand on several courses (e.g. http://ocw.mit.edu/index.html). Also, specialized wireless lecture video streaming servers for distance learning are being setup in various universities. The every-day increasing demands for such vital multimedia services along with the rising expectation of quality for these services render many of the currently used techniques and systems less-useful [1,21]. Existing multimedia software (even those specialized to digital image processing) do not provide par excellence multimedia data service to the common user, due to a lack of unified on-chip video processing and transmission capabilities.

We propose to utilize the simulation approach for efficient hardware-software co-design, since this approach has been recently gaining a lot of popularity [5, 20, 31]. We implement wireless transmission system over FPGA [18], addressing the need of dedicated hardware system for video data transmission using MIMO (Multiple Input Multiple Output) transmission techniques and using high-level synthesis and design language, and superior design-methodologies for effective design. Our work has great significance in making the company managers equipped with technical knowledge required for providing e-learning solutions to masses.

The contributions of our work are as follows:

(1) We present a solution using FPGA as target platform (instead of software), for implementing algorithms for transmission of educational videos using MIMO over and above SISO (Single Input Single Output) or MISO (Multiple Input Single Output) systems.

(2) We develop a model of Alamouti MIMO communication technique using STBC over Xilinx System Generator (XSG) using the modelling and simulation capabilities of Simulink.

(3) We design the hardware using Xilinx System Generator, which offers many advantages over existing implementation platforms. We also perform analysis of various features and design issues for HDL and high-level graphic design language.

The rest of the paper is organized as follows. We first discuss the literature review of on-going work in the area of multimedia communication, MIMO systems, FPGA and its design options and set the stage for the need of efficient design tools and methods. We then present our design methodology and implementation of Alamouti scheme (MIMO) on XSG and AccelDSP for FPGA design. We further present the design features which enhance the performance of the system, with special emphasis on the general high level FPGA design options, by taking representative tool, namely XSG. We then present critical evaluation and insights gained from implementation and performance results of our approach and its comparison with traditional HDL. Finally, we present the conclusion and future research directions.

2. LITERATURE REVIEW

Limitation of software platforms

Software implementation of most multimedia processing algorithms has several limitations. At real-time video rates of 25 frames per second a single operation performed on every pixel of a 768 by 576 colour image (PAL frame) equates to 33 million operations per second (excluding the overhead of storing and retrieving pixel values). As a result it is difficult to meet hard real time requirements with software [17].

Need of MIMO Systems

To achieve higher data rates than conventional SISO (Single Input Single Output) and MISO (Multiple Input Single Output) channels, we have employed the techniques of MIMO (Multiple Input Multiple Output), with Alamouti scheme. MIMO systems can be used to increase system capacity as well as data reliability in wireless communication systems [2]. Research has been conducted in developing space-time codes [15] for transmission over MIMO systems. Foschini et al. [12] have shown that channel capacity for MIMO system increases as the number of antennas is increased, proportional to the minimum number of transmit and receive antennas. Parallel transmission ensures high data rate enabling transmission of a large video data in smaller time duration. For guaranteeing best perceptual quality, reliability is an important consideration in video streaming.

The data rate increase obtained through the use of MIMO channels is evident by the following conventional Shannon capacity equations [3], where H denotes the channel matrix or the transfer function of the channel:

$$C = log_2\ (1 + SNR\ \times\ |H|^2)\quad \textit{for SISO channel}$$

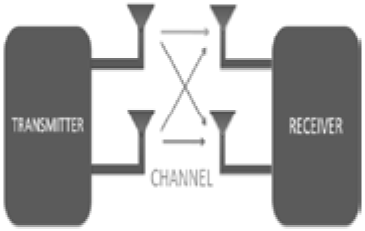
$$C = log_2 (1 + SNR \times HH^*)\quad \textit{for MISO/SIMO channel}$$

$$\mathrm{C} = \log_2\ \left(\det\left(\mathrm{I_M} + \frac{\mathrm{SNR \times HH^*}}{\mathrm{N}} \right) \right)$$

for MXN MIMO channel

Here, *H = abs (H); H* = conj (trans (abs (H)));* IM is the identity matrix; M is the number of transmit antennas, N is the number of receive antennas.

The basic Alamouti scheme uses space time block coding to ensure higher reliability of data throughput. This is achieved through a 2 X 2 transmitter system, composing of 2 transmit antennas and two receive antennas. STBC scheme is shown in the figure 1.



$$Channel\ Matrix\ H = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix}$$

$$ST\ Block\ Code\ Input\ \mathrm{X}_t = \begin{bmatrix} s_1 & s_2 \\ -s_2^* & s_1^* \end{bmatrix}$$

Figure 1: (a) 2X2 Alamouti STBC diagram.
(b) Input and channel matrices

These symbols are then decoded at the input using maximum likelihood decision rule.

FPGA and Hardware Implementation

Recently, image processing and digital signal processing algorithms implemented in hardware have emerged as the most viable solution for improving the performance of these systems. A Field Programmable Gate Array (FPGA) [4] is one such reconfigurable hardware, offering superior features over DSPs and ASICs in many areas such as digital image processing [22,26], digital signal processing [6,24] and so on. Many experiments have been performed which substantiate the idea that FPGAs can outperform DSPs and embedded processors in signal processing, while still being highly energy efficient [10,28]. The high performance and complexity requirements of multimedia applications call for the massive parallel processing capabilities inherent to FPGA. Parker [27] discusses the advantage that FPGAs have due to their product reliability and maintainability which also improve its development process. Moreover, it has been shown that the right tools and techniques coupled with innovative features in silicon architecture can yield complete digital signal processing functions in a single FPGA. The multiple iterative processing of data sets (such as four stages of canny edge detector) required in many DIP algorithms have to be performed sequentially on a general-purpose computer, which can be fused in one pass in FPGA, as their structure is able to exploit both spatial and temporal parallelism. Moreover, use of several optimizations techniques such as loop fusion and loop unrolling lead to improved speeds. The reconfigurable nature of architecture allows a large variety of flexible logic designs dependent on the processor's resources [9].

MIMO systems generally require a large processing time when working with large amount of data (e.g. video lectures), as the blocks of bits generated through compressed data are still quite large in number for real time processing, thus necessitating a huge processing time on software. FPGAs can be employed for increase in speed enhancement, as they offer parallel implementation of time consuming blocks, thus increasing the speed greatly. Since the speed of software is limited by the internal processor clocking and other processes running on the system, dedicated hardware for such MIMO systems can be developed using FPGAs [7,23,30].

FPGA Design Tools

The full potential of FPGAs, however, remains unharnessed due to unavailability of design tools which offer ease of programming, short design and verification time etc. The experience of many designers using low level languages such as VHDL or Verilog for FPGA design shows that it is very difficult to write a code that synthesizes easily, without requiring weeks or months. Creating an FPGA design, using hardware description languages such as VHDL or Verilog with corresponding simulation, synthesis, and layout and timing analysis tools is a daunting task. They work well as HDLs but are poor programming languages. Hence there has been a shift of interest towards newer design methodologies and implementation platforms that provide higher level abstraction, ease and reliability of design and so on [25].

Presently, a range of high-level tools and languages for FPGA design are available including: Celoxica's Handel-C which is a C based parallel language that generates EDIFs (http://www.celoxica.com/methodology/c2rtl.asp); C++ extensions such as SystemC [14]; Xilinx's Forge Java– a Java to Verilog tool (www.xilinx.com/ise/advanced/forge.htm) etc. Annapolis MicroSystems has developed "CoreFire" which uses prebuilt blocks which removes the need for the back-end processes of the FPGA design flow. Ptolemy is a system that allows modelling, design, and simulation of embedded systems. Ptolemy provides software synthesis from models. All these systems are still under development and have their own limitations.

Xilinx System Generator (XSG) is a state of the art tool which offers high performance and requires very small learning and development time [11]. XSG for DSP [24] is a tool which offers block libraries that plugs into Simulink tool (containing bit-true and cycle-accurate models of their FPGA's particular math, logic, and DSP functions). The AccelDSP tool allows DSP algorithm developers to create HDL designs from MATLAB and export them into System Generator for DSP.

The experts from different fields have employed and evaluated XSG for its performance. Frigo et al. [13] implement and compare two approaches using high-level tools, to map a signal processing algorithm to FPGA, the first one uses XSG and the second one uses Streams -C language and compiler . This module is used in a phase modulation sorter that locates binary phase shift keying (BPSK) signals in wide-band data. The results show the superiority of XSG, for area utilization and placement speed.

Hwang et al. [16] deal with System Level Tools for DSP in FPGAs. As a design example they implement a sampling rate conversion module which is used in audio signal applications. They discuss several features and functionalities of XSG such as hardware handshaking, multi-rate systems, bit true and cycle true modelling etc. A 2X to 4X productivity improvement is likely to be achieved using System Generator over conventional HDL language development methods due to its design environment and simulation speed. Mittal et al. [19] report the implementation of the well-known Simplex algorithm on XSG for solving integer linear programming problem. They also discuss the FPGA implementation using Xilinx FPGA and discuss the opportunity of scaling to large problems.

Related Work on MIMO on FPGA

Dimitrov et al. [8] discuss FPGA implementation of 2*2, 3*4 and 4*2 MIMO systems using Simplify Pro synthesis tool. Like our approach, they also use a high-level design tool to avail the benefit of high-level design methodology. However, their work differs from our work in that they do not provide the implementation results on multiple FPGAs, while we provide these results for multiple FPGA boards (see results section). Veena et al. [29] present the implementation of MIMO using Verilog HDL and Spartan FPGA. However, they use HDL (hardware design language) methodology, while as we use GUI based high-level design flow.

3. SYSTEM ARCHITECTURE

Figure 2 shows the block diagram of our Alamouti MIMO (2*2) system. The design of hardware has been carried out using the Model-based design offered by the Xilinx

System Generator, with total learning time and development time of less than a month by one engineer. To simulate the performance, Rayleigh fading channel has been also simulated by modelling its property such as random noise and channel matrix. Our model-based design also precludes direct design in VHDL/Verilog, thus ensuring fewer bugs and shorter verification cycle. This model takes advantage of modelling and simulation capabilities of Simulink and thus also provides flexibility of design. Visual data flow environments are well suited for modelling DSP systems, which are most naturally specified by signal flow graphs. Moreover, use of Alamouti scheme results in further decrease in BER of the system.

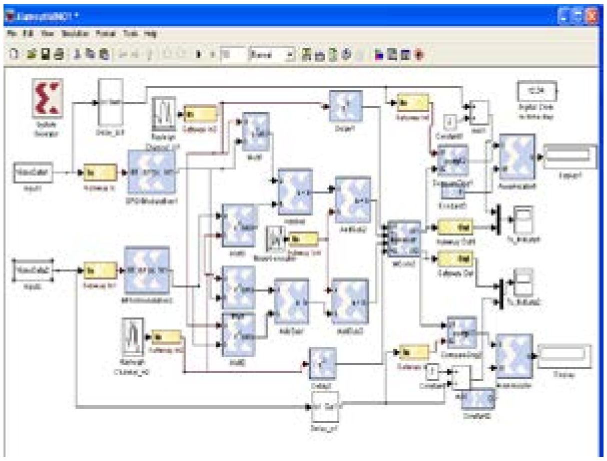


Figure 2 : Block Diagram: Alamouti MIMO (2*2) scheme implemented on XSG

Figure 2 demonstrates the block-diagram of our design. We now explain it in more detail.

1. The block on the transmitter side for carrying out BPSK modulation sends its input to the output of the two antennas at the transmitter side. The input signal is randomly generated and is also used ultimately to verify errors in the transmission. This completes transmitter side.

2. The channel is simulated in the design itself. The channel is modelled as a flat-fading Rayleigh channel with random noise. The H matrix of the channel models its characteristic, which affects the signal in a particular way. The noise is additive in nature, to balance the delays introduced by some of the blocks, intentional delay is introduced in the other paths.

3. The receiver implements Maximum Likelihood decoding which is used to finally take the decision about the transmitted symbol. This is followed by BPSK demodulation, so that symbol may be converted in the original signal format. A comparison of the received signal and the original signal shows the errors in transmission.

4. Performance of the system is assessed by performing simulation of transmission of a large number of bits. Cumulative error is termed as Bit Error Rate (BER) and this is determined by accumulating individual errors. This model can be simulated on software for accuracy.

5. The symbol of 'System Generator' on the left is finally used to convert the design into fixed point model, implementable on the hardware. XSG generates complete files which can be directly used in Xilinx ISE software for creating the final design.

Apart from the blocks provided by the standard library, we have also made custom blocks using both MCode blocks (for non-algorithmic code) and AccelDSP (for algorithmic code). This helped us to extend the utility of the XSG tool. An important attribute of our design using AccelDSP was that the blocksets generated in AccelDSP for XSG, are reusable and can be neatly divided into appropriate libraries each containing blocks specific to a certain field such as (for example) Image Processing Library, MIMO System Library etc. depending on their applications. The MIMO System Library, for example may contain all blocks made by us which are useful in MIMO communication field and so on.

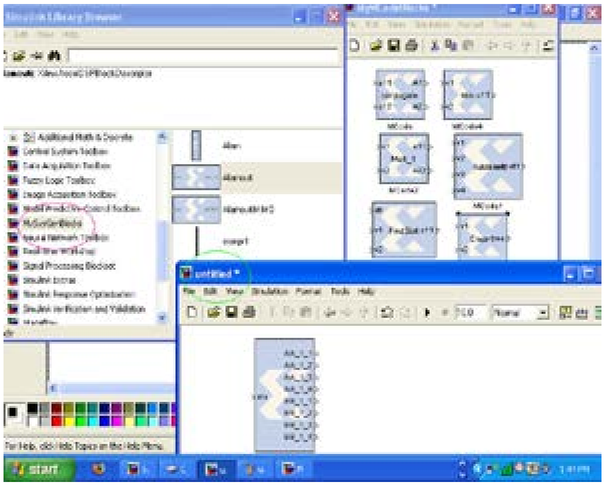


Figure 3 : A screenshot of user-defined blocks augmenting functionality of XSG

Figure 3 shows the library consisting of custom blocks designed by us. *MySysGenBlocks* library in Simulink Library Browser (encircled) contains blocks used in design of MIMO systems. Another Simulink file, named *MyMCodeBlocks* contains the library of blocks made from MCode and gives the user facility to augment the functionality in

XSG. The model file of the user named ‘untitled’ can use blocks from both of these sets also.

4. SALIENT FEATURES OF OUR APPROACH FOR SUPERIOR PERFORMANCE

Figure 4 shows our over-all design approach for integrated design of hardware and software. In what follows, we briefly discuss several important features which provide flexibility and power to the designer.

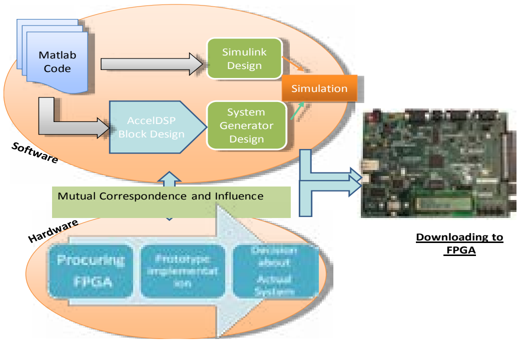
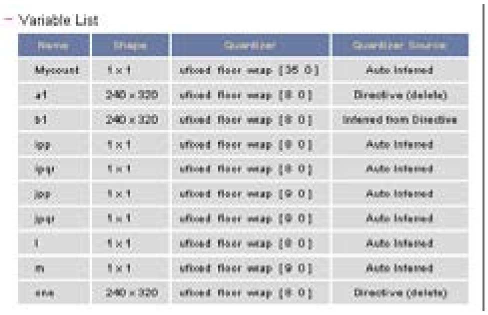


Figure 4: Design approach for hardware/software co-design

An important factor contributing to optimization of code and reduction in design time is option for setting quantization of variables. A sound solution to the general problem faced in programming in HDLs of ambiguity in allocation of sufficient bits for any variable (or identifier) is provided by AccelDSP. Apart from automatically inferring the bit-length of the variable after quantization the tool also provides option to the user for selecting the requisite number of bits manually (Figure 5 shows the output of the ‘Generate Fixed Point Report” where the bitlength for variable a1 has been assigned by the user, for b1 bitlength has been assigned same as that for a1 due to relationship between them and for other variables the bit-length has been auto-inferred.). This option has two-fold advantages; automatic assessment removes the burden of the programmer to allocate the sufficient number of bits for each and every variable in the program (as required in HDLs) and also gives the choice and flexibility to the user to remove the problem of overflow and underflow which are very common in design using HDL.



− Variable List

Name	Shape	Quantizer	Quantizer Source
Mycount	1 x 1	ufixed floor wrap [35 0]	Auto Inferred
a1	240 x 320	ufixed floor wrap [8 0]	Directive (delete)
b1	240 x 320	ufixed floor wrap [8 0]	Inferred from Directive
ipp	1 x 1	ufixed floor wrap [9 0]	Auto Inferred
ipqr	1 x 1	ufixed floor wrap [8 0]	Auto Inferred
jpp	1 x 1	ufixed floor wrap [9 0]	Auto Inferred
jpqr	1 x 1	ufixed floor wrap [9 0]	Auto Inferred
i	1 x 1	ufixed floor wrap [8 0]	Auto Inferred
m	1 x 1	ufixed floor wrap [9 0]	Auto Inferred
ene	240 x 320	ufixed floor wrap [8 0]	Directive (delete)

Figure 5 : of AccelDSP showing quantization of variables

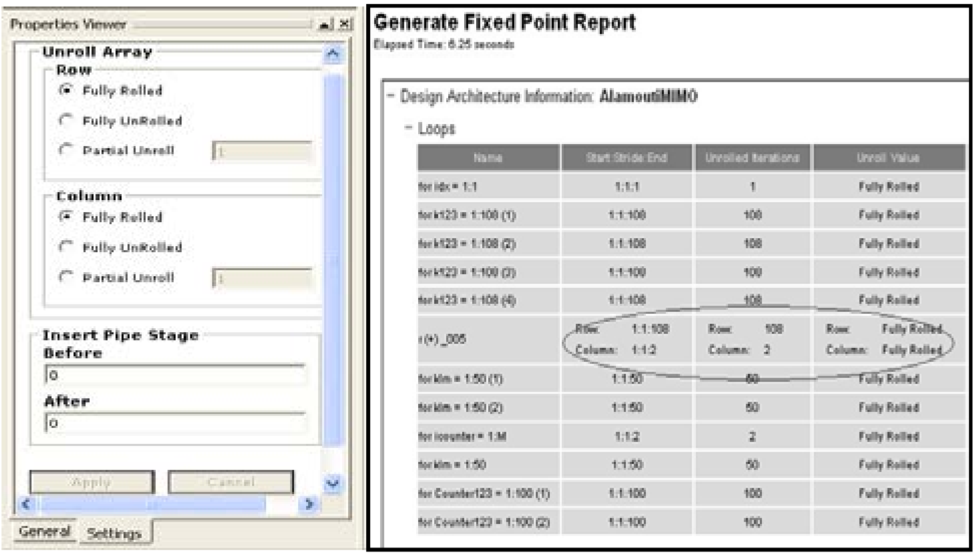
We also employ advanced user-friendly options for achieving compatibility. The use of different syntaxes and keywords in different languages poses problems for the programmers who are familiar with a single programming language. It becomes a severe bottleneck for the programmers to understand the syntax and semantics of top level to bottom level programming. However, the automatic and user-friendly tools such as AccelDSP, which is used to convert the Matlab code into either a System Generator block or hardware netlist, help and guide the user in taking into account the diversity in syntax, keywords, programming constructs etc. between multiple languages, namely Matlab, AccelDSP and Verilog/VHDL.

One of the powerful and valuable design option for efficient hardware generation in AccelDSP is the “Unrolling a Loop to Increase Hardware Performance”. Thus traditional trade-off between hardware resource availability and degree of parallelism can be well resolved by flexibility to directly alter low level design factors. Figure 6(a) shows the option of hardware optimization possible by choosing proper rolling factor in any ‘for’ loop in AccelDSP. A general ‘for’, ‘while’ loop implemented in software or in hardware (with no optimization) needs to be executed as many times as the iteration count (say N) of the loop. This would mean applying the data to a single data path for each time. However, if the loop construct is completely unrolled, then the AccelDSP Synthesis Tool builds a different hardware structure where N data samples are applied simultaneously to N identical parallel data paths. Performance is increased N times at a cost of increasing the hardware area by N times. For a nested loop, the option is also provided to separately optimize both the loops by choosing loop rolling factors individually for row and column loops. If a full unroll consumes too much hardware, then the user can go for a partial unroll making a suitable balance between area and performance requirements. The encircled portion in the figure 4 refers to the optimization of the following statement in the AccelDSP,

$$r = r + 0.5\,;$$

Here r is a two dimensional (108*2) array. The statement adds .05 to each of the element of r. Note that this statement has been fully unrolled in both the dimensions, this serves as a typical example of optimization of N-dimensional loop in AccelDSP.

Figure 6(b) shows the ‘Fixed Point Report’ generated by the AccelDSP for a design of MIMO block. Thus, the possibility of using the ‘loop unrolling’ gives the designer a choice to trade-off between area and performance. This serves as an example of unavoidable nature of mutual influence between hardware and software. An insight into hardware resources availability (depending on hardware platform) and utilization (depending on nature of program and software) can lead to optimal match between efficient design and resource allocation. This validates our approach of co-design for selection of design parameters.



Unroll Array

Row
● Fully Rolled
○ Fully UnRolled
○ Partial Unroll 1

Column
● Fully Rolled
○ Fully UnRolled
○ Partial Unroll 1

Insert Pipe Stage

Before
0

After
0

Generate Fixed Point Report

Elapsed Time: 6.25 seconds

− Design Architecture Information: **AlamoutiMIMO**

− Loops

Name	Start:Stride:End	Unrolled Iterations	Unroll Value
for idx = 1:1	1:1:1	1	Fully Rolled
for k123 = 1:100 (1)	1:1:100	100	Fully Rolled
for k123 = 1:108 (2)	1:1:108	108	Fully Rolled
for k123 = 1:100 (3)	1:1:100	100	Fully Rolled
for k123 = 1:108 (4)	1:1:108	108	Fully Rolled
r(+)_005	Row: 1:1:108 Column: 1:1:2	Row: 108 Column: 2	Row: Fully Rolled Column: Fully Rolled
for klm = 1:50 (1)	1:1:50	50	Fully Rolled
for klm = 1:50 (2)	1:1:50	50	Fully Rolled
for icounter = 1:M	1:1:2	2	Fully Rolled
for klm = 1:50	1:1:50	50	Fully Rolled
for Counter123 = 1:100 (1)	1:1:100	100	Fully Rolled
for Counter123 = 1:100 (2)	1:1:100	100	Fully Rolled

Figure 6 : Option to unroll an Array, both row and column wise.
(b) Output of AccelDSP "Generate Fixed Point Report" showing loop rolling

5. EXPERIMENTS AND RESULTS

BER through XSG implementation of Hybrid Alamouti

Figure 7 shows the output of the System Generator block simulation for transmission of 500 bits. The two figures correspond to two receiver antennas at the receiver side in 2*2 schemes. The encircled values of the signal correspond to error in transmission due to noise and inter-symbol interference. Clearly, the design shows a very small BER value of 10/500 (.025), which shows the superiority of our design. Additionally, a graphical design and output along with the power of Simulink comes as a great boon for the programmer.

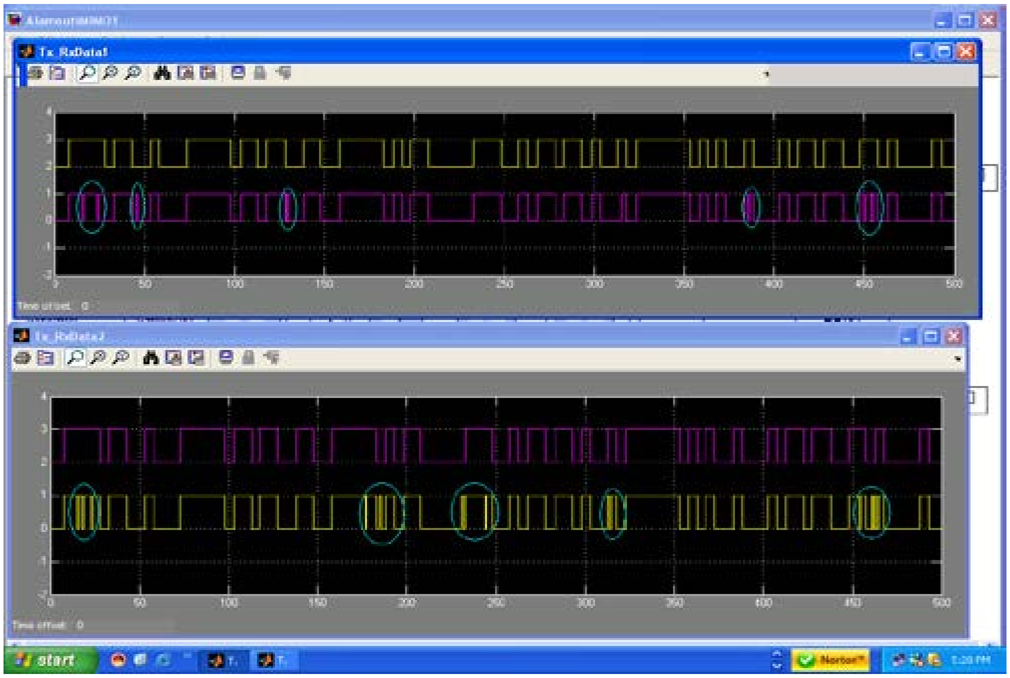


Figure 7 : Bit Error Rate Measurement using XSG Simulation

Downloading the Design to FPGA

Using XSG design explained in previous sections, we downloaded our design on FPGA. To test this, inputs were sent to the communication system through the channels modelled (see Figure 2) and the outputs were measured, which were as they were expected (as shown earlier). This verifies both the FPGA and XSG design.

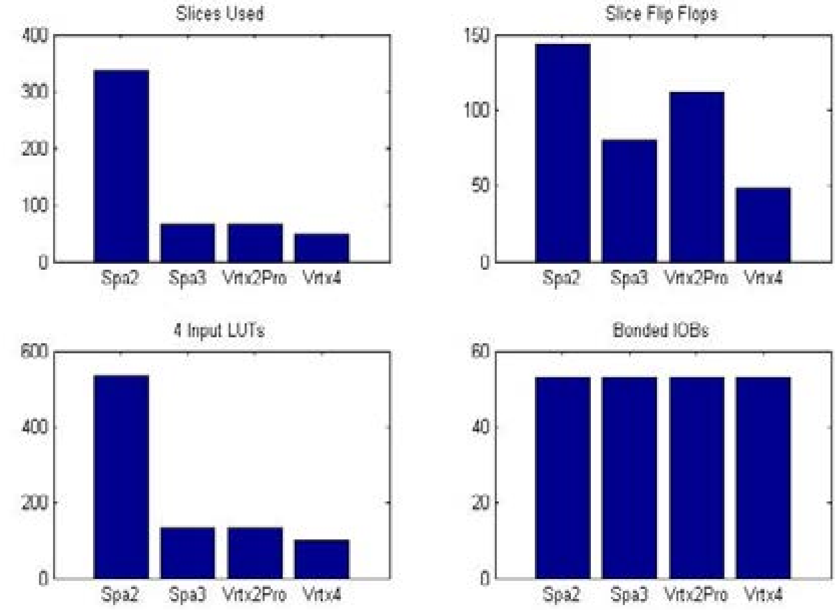


Figure 8 : Resource utilization in different target platforms. (Where Spa2= Spartan2 xc2s200-6fg456, Spa3= Spartan3E xc3s500e -5fg320, Vrtx2Pro = Virtex2Pro xc2vp30 ff6896, Vrtx4= Virtex4 xc4vfx12 ML402, LUTs = Look up tables, IOBs = Input or output blocks)

Figure 8 shows the resource utilization in different target platforms in terms of four parameters which are common to all the platforms. The comparison shows clearly the optimum nature of the platforms on the higher ends. The Spartan 2 platform uses the highest number of resources in all the parameters, while as Virtex 4 kit uses the lowest. It is to be noted that the resources depend critically on the nature of the platform (called specification of the family) and may also vary with the design.

Critical Evaluation and Insights Gained

It is estimated that verification of the code takes 60% to 80% of the effort in the hardware design groups. In such scenario, the boost in performance also results from the relatively reliable and error-free model-based design in Simulink and XSG, where the errors are discovered and removed early in the design flow at the modelling stage and not lat the VHDL behavioural test stage, which can be difficult and time consuming. Moreover, unlike traditional low-level design flow, in XSG, all the necessary clocking signals are generated automatically, and components are easily connected, thus leading to simplified clocking and interfacing. However, there are still some limitations of the tool itself which hinder improvement in speed. System Generator is a relatively new product and has not been fully developed yet. One limitation of XSG is inability to accept multi-dimensional data directly and many of the standard functions are thus deprecated. So, such data has to be given sequentially as one dimensional data and those functions which accept multi-dimensional input have to be redefined or the algorithm needs to be altered. Optimal usage of hardware and better clock speed are some other design issues where XSG tool should be improved upon.

To help and guide the computer architects and designers wTo help and guide the computer architects and designers we present the following table, summarizing the difference between design using low-level and high-level design methodologies. The insights discussed here are generic and also apply to other tools or design paradigms. In today's highly competitive market, reduced production cycle time and cost through use of high-level design tools can prove to be significant value to the companies.

TABLE 1
COMPARISON OF LOW-LEVEL AND HIGH-LEVEL DESIGN METHODOLOGY FOR FPGA IMPLEMENTATION

	Low-level Design Methodology	High Level Design Methodology
Learning and Design time	3 months	4 weeks
Debugging time	Larger	Small (GUI design)
Code reuse	Difficult and likely to create error	Easy to reuse the blocks by integrating into library
Design overheads	User must involve himself in many low-level design issues	Tool automatically generates HDL netlist, place and route info etc.
Flexibility	Modifying the program is very error-prone	Model based design: modification very easy
Simulation and debugging support	Large time required in simulating and creating waveforms	Avails the power of simulation of high level languages
Simulation Time	High	Faster than HDL simulator
Higher level routines	Limited	90 DSP blocks available, can be extended

CONCLUSION

Advances in FPGA technology along with development of efficient tools for modelling, simulation and synthesis have made FPGA a highly useful platform and they are expected to provide the breakthrough for systems dealing with large resource (data) such as educational video transfer. Currently, it is the medium of choice for the hardware development and implementation of high-performance applications, requiring rigorous computations. Xilinx System Generator is a system level modelling tool that facilitates FPGA hardware design by extending Simulink/Matlab in numerous ways in order to provide a powerful modelling environment.

In this work, we present a solution to the growing need of computational capability using FPGA as a target platform for implementing MIMO (over and above SISO or MISO) systems. Such a system also achieves higher data rates with lower error rates in transmission and has practical utility in transmission of educational video in multimedia services. We developed a model of Alamouti MIMO communication technique over Xilinx System Generator and presented several outstanding features of our design methodology which lead to integrated hardware-software design. We analysed various features of Xilinx System Generator from the perspective of a designer.

Future research should bring System Generator as an efficient all-in-one design tool valuable for wide areas of application. Our future work will focus on implementing more complex and advanced MIMO communication techniques, such as 4×4 MIMO on FPGA. We will develop our approach further and experiment with the system by integrating our design in a larger real-time system providing multimedia e-learning to the students.

REFERENCES

[1] S. Gupta, S. Mittal, S. Dasgupta. "Guaranteed QoS with MIMO Systems for Scalable Low Motion Video Streaming Over Scarce Resource Wireless Channels." in *Proceedings Second International Conference On Information Processing*, pp. 452-466, 2008.

[2] S. Gupta, S. Mittal, S. Dasgupta, A. Mittal. "MIMO Systems For Ensuring Multimedia QoS Over Scarce Resource Wireless Networks." in *ACM International Conference On Advance Computing*, India 2008.

[3] S.M. Alamouti. "A simple transmit diversity technique for wireless communications." *IEEE Journal on Selected Areas in Communications,* Vol. 16, no. 8 ,pp. 1451-1458,1998.

[4] S.D. Brown, J.F. Robert, R. Jonathan, G.V. Zvonko. *Field programmable gate arrays*, Vol. 180, Kluwer Academic Pub, 1992.

[5] S. Mittal, Z. Zhang, "ESTO : A Performance Estimation Approach for Efficient Design Space Exploration", in *Design Contest at 26th IEEE International Conference on VLSI Design*, 2013.

[6] S. Choi, R. Scrofano, V. K. Prasanna, J.W. Jang. "Energy-efficient signal processing using FPGAs." in *Proceedings of the 2003 ACM/SIGDA eleventh international symposium on Field programmable gate arrays*, ACM, pp. 225-234, 2003.

[7] M. Cui, H. Murata, K. Araki. "FPGA implementation of 4× 4 MIMO test-bed for spatial multiplexing systems." in *15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, vol. 4, pp. 3045-3048, 2004.

[8] E. Dimitrov, C. Kupferschmidt, T. Kaiser, J. Korpi, R. Nordman, A. Anttonen, A. Giorgetti, M. Chiani. "Design and implementation of advanced algorithms for MIMO-UWB wireless communications." in *IEEE International Conference on Ultra-Wideband (ICUWB)*, pp. 288-292, 2011.

[9] A. Draper, Bruce, J. R. Beveridge, A.P.W. Bohm, C. Ross, M. Chawathe. "Accelerated image processing on FPGAs." *IEEE Transactions on Image Processing*, Vol. 12, no. 12, pp. 1543-1551,2003.

[10] R. Duren, J. Stevenson, M. Thompson. "A comparison of FPGA and DSP development environments and performance for acoustic array processing." in *50th Midwest Symposium on Circuits and Systems MWSCAS, IEEE*, pp. 1177-1180, 2007.

[11] P. Fandén. "Evaluation of Xilinx System Generator." *PhD diss., Södertörn University*, 2001.

[12] G.J. Foschini, M.J. Gans. "On limits of wireless communications in a fading environment when using multiple antennas." *Wireless personal communications*, vol. 6, no. 3, pp. 311-335,1998.

[13] J. Frigo, T. Braun, J. Arrowood, M. Gokhale. "Comparison of high-level FPGA design tools for a bpsk signal detection application." in *Proceedings, SDR Forum Technical Conference*, 2003.

[14] J. Gerlach, W. Rosenstiel. "System level design using the SystemC modeling platform." in *System Design Automation*, Springer US, pp. 27-34, 2001.

[15] D. Gesbert, M. Shafi, D. Shiu, P.J. Smith, A. Naguib. "From theory to practice: an overview of MIMO space-time coded wireless systems." *IEEE Journal on Selected Areas in Communications*,Vol. 21, no. 3, pp. 281-302, 2003.

[16] J. Hwang, B. Milne, N. Shirazi, J. D. Stroomer. "System level tools for DSP in FPGAs." in *Field-Programmable Logic and Applications,* Springer, Berlin Heidelberg, pp. 534-543, 2001.

[17] S. Klupsch, M. Ernst, A. H. Sorin. Integrierte Schaltungen und Systeme, M. Rumpf, and R. Strzodka. "Real time image processing based on reconfigurable hardware acceleration." In *Workshop Heterogeneous Reconfigurable Systems on Chip (SoC)*, p. 1. 2002.

[18] S. Mittal, S. Gupta, S. Dasgupta. "FPGA: An Efficient And Promising Platform For Real-Time Image Processing Applications." in *National Conference On Research Development In Hardware Systems (CSI-RDHS)*, 2008.

[19] S. Mittal, L. Wang, A. Pande, P. Kumar. "Design exploration and implementation of simplex algorithm over reconfigurable computing platforms." in *IEEE International Conference on Digital Convergence*, pp. 204-20, 2011.

[20] S. Mittal, Z. Zhang. "Integrating Sampling Approach with Full System Simulation: Bringing Together the Best of Both." in *IEEE International Conference On Electro/Information Technology (EIT), Indianapolis, USA, 2012.*

[21] S. Mittal, S. Gupta, A. Mittal. "BioinQA: metadata–based multi–document QA system for addressing the issues in biomedical domain." *International Journal of Data Mining, Modelling and Management,* Vol. 5, no. 1, pp. 37-56, 2013.

[22] A.E. Nelson. "Implementation of image processing algorithms on FPGA hardware." *PhD diss., Vanderbilt University*, 2000.

[23] M. Numan, Wasiuddin, M.T. Islam, N. Misran. "Implementation of Alamouti Encoder Using FPGA for Mimo Testbed." in *Advanced Computer Control, 2009. ICACC'09. International Conference on*, IEEE, pp. 188-192, 2009.

[24] S. Mittal, S. Gupta, S. Dasgupta. "System generator: The state-of-art FPGA design tool for dsp applications." in *Third International Innovative Conference On Embedded Systems, Mobile Communication And Computing (ICEMC2)*, 2008.

[25] M. Ownby, W. H. Mahmoud. "A design methodology for implementing DSP with Xilinx System Generator for Matlab." *in Proceedings of the 35th Southeastern Symposium on System Theory*, IEEE, pp. 404-408, 2003.

[26] A. Pande, S. Mittal. "BayWave: BAYesian WAVElet-based image estimation." *International Journal of Signal and Imaging Systems Engineering,* vol. 2, no. 4, pp. 155-162, 2009.

[27] M. Parker. "FPGA versus DSP design reliability and maintenance." 2010, http://www.dsp-fpga.com/articles/id 2207 (2010).

[28] R.Scrofano, S. Choi, V.K. Prasanna. "Energy efficiency of FPGAs and programmable *processors for matrix multiplication." in Proceedings of IEEE International Conference on Field-Programmable Technology, (FPT)., IEEE*, pp. 422-425, 2002.

[29] Veena, M.N.S. Swamy. "DESIGN & IMPLEMENTATION OF RECONFIGURABLE FRONT END FOR MIMO-OFDM." *International Journal of Engineering Science*, vol. 3, 2011.

[30] X. Wu, J.S. Thompson. "A fixed-complexity soft-MIMO detector via parallel candidate adding scheme and its FPGA implementation." *Communications Letters, IEEE*, vol. 15, no. 2, pp. 241-243, 2011.

[31] S. Mittal, "OPNET: An Integrated Design Paradigm for Simulations", S*oftware Engineering : An International Journal (SEIJ)*, vol. 2,no. 2, pp. 57-67, 2012.

ABOUT THE AUTHORS

Sparsh Mittal. Sparsh Mittal received his BTech in Electronics and Communications Engineering from the Indian Institute of Technology, Roorkee, India. He was the graduating topper of his batch and also received best B. Tech project award. Currently, he is pursuing his PhD in Electrical and Computer Engineering at Iowa State University (ISU), USA. He has been awarded scholarship and fellowship from IIT Roorkee and ISU. He has been a reviewer of conferences like IEEE Intelligent Systems, IEEE ISVLSI (IEEE Annual Symposium on VLSI), IEEE ISWTA (IEEE Symposium on Wireless Telecommunications Applications), and book series such as Understanding Complex Systems (Springer) etc. He has published more than 15 papers in conferences and journals. His research interests include FPGA, simulation modeling and computer architecture.

Saket Gupta. Saket Gupta received the M.S. and PhD. degree in Electrical and Computer engineering from the University of Minnesota, Twin Cities, in 2011 and 2012, respectively, and the B.Tech. degree from the Indian Institute of Technology, Roorkee, India, in 2008. He is currently a Staff Scientist in Central Engineering, Library Development at Broadcom Corporation. His research interests include the development and implementations of VLSI circuits and systems and CAD-related timing tools. He has published international conference and journal papers in the area of variability and reliability, low power and high-performance circuits and architectures, with best paper nomination in IEEE/ACM ASPDAC 2010. His current role includes design of deep submicorn SRAMs, variability sensors for SRAMs and silicon validation. He has served as reviewer for IEEE/ACM international conferences such as DAC, ICCAD, VLSID, and ISVLSI, and IEEE/ACM international journals such as TVLSI, TCAS-I, TODAES, and Integration:the VLSI journal.