

Compact Hardware Accelerator for Functional Verification and Rapid Prototyping of 4G Wireless Communication Systems

Yuanbin Guo, Dennis McCain
Nokia Research Center
Irving, TX, 75039
Email: {Yuanbin.Guo, Dennis.McCain}@nokia.com

Abstract— In this paper, we propose an FPGA-based hardware accelerator platform with Xilinx Virtex-II V3000 in a compact PCMCIA form factor. By partitioning the complex algorithms in the 4G simulator to the hardware accelerator, we apply an efficient Catapult-C methodology to quickly evaluate the area/speed tradeoffs and rapidly schedule synthesizable RTL models for implementation. The simulation time is accelerated by $100\times$ for a QRD-M algorithm. This not only enables much faster verification in the 4G standard environment, but also provides software/hardware co-design and rapid prototyping of the core algorithm in a realistic fixed-point platform.

I. INTRODUCTION

Future 4G systems are emerging to support much higher data rate than today for multimedia services and ubiquitous networking via mobile devices. Much more complicated signal processing algorithms are required to achieve the high throughput. MIMO (Multiple Input Multiple Output) technology [1] using multiple antennas at both the transmitter and receiver sides leads to MIMO-OFDM [2] as a strong candidate for the 4G standards. To achieve a good tradeoff between performance and complexity, a suboptimal QRD-M algorithm was proposed in [3] [4] to approximate the optimal but prohibitively complex Maximum Likelihood(ML) detector. It applies the QR-decomposition [5] to reduce the channel matrices to upper triangular matrices and limited tree search to the M smallest metric branches.

Despite of the significantly reduced complexity, the QRD-M algorithm is still the bottleneck in the receiver design. The very high complexity requires extremely long run-time in software simulators. Days or even weeks are needed to generate one simulation point. This not only slows down the research activity severely, but also gives tremendous challenges for real-time hardware implementation [6]. Due to the advances in chip technology, it is now possible to implement very complex DSP algorithms on FPGAs (Field Programmable Gate Arrays). However, many FPGA hardware platforms usually have large form factors, making it difficult to carry, demonstrate and integrate in the matlab simulation chain. In this paper, we present a FPGA-based hardware accelerator platform with compact form factor for easier integration with the 4G simulator. This hardware accelerator has a single Xilinx Virtex-II V3000 FPGA in a PCMCIA card which supports up to 300K gates with clock rate up to 65 MHz. We derive the interface to

the matlab simulation chain using C-MEX file to talk with the API functions on the host. The APIs communicate with the VLSI design core using DMA and local bus interface on the FPGA side. Extensive peripherals make it a powerful platform for demonstration of the VLSI design of the core algorithms.

The platform is also intended to speed up the research activities and rapidly prototype the VLSI architectures for proof of concept in the future real-time 4G systems. The efficient VLSI architectures are explored by a Catapult-C based high-level synthesis design methodology [7] and implemented in the hardware accelerator. By partitioning the critical algorithms in a simulation chain to the hardware accelerator, the simulation time is reduced significantly, thereby generating results more quickly. Moreover, this enables extensive exploration of the architecture complexity for the partitioned design. Functional verification of the fixed-point VLSI design in a system level simulation chain is achieved. This makes the future commercialization of the core algorithms to the product much easier.

For the chosen QRD-M algorithm, speedup of $100\times$ in the integrated software/hardware co-design accelerator platform is observed with 33 MHz FPGA clock rate competing with the 1.5 GHz Pentium-4 PC clock rate. The scalability in the VLSI architecture enables even faster acceleration with more Processing Elements. The P & R clock rate of the current VLSI architecture can be up to 90 MHz. This provides a reference for the future 4G real-time prototyping.

II. SYSTEM MODEL

We consider the equivalent baseband signal model for the MIMO OFDM system with N_T transmit and N_R receive antennas. The system model is shown in Fig. 1. We assume that there are N_F subcarriers. At the transmitter, the high data rate bit stream is broken down into N_T lower rate substreams by spatial multiplexing. At the p^{th} transmit antenna, the multiple bit substreams are modulated by constellation mappers to some BPSK, QPSK, 16-QAM or 64-QAM symbols. $d_p(\kappa * N_F + k)$ is the k^{th} frequency domain subcarrier during the $[\kappa T_d^g, (\kappa + 1)T_d^g]$ OFDM symbol epoch and T_d^g is the symbol duration including the guard time. For simplicity in the notation, we focus on the κ^{th} symbol and omit the symbol index κ . During this symbol epoch, the symbols are modulated by N_T parallel N_F -point IFFTs to generate the n^{th} time

III. HARDWARE ACCELERATOR

A. QRD-M Matrix Symbol Detector

It is known that ML sequence detector is the optimal detector. However, it has prohibitively high complexity growing exponentially with the number of antennas, as $\mathcal{O}(C^{N_T})$, where C is the size of the symbol alphabet and N_T the number of transmit antennas. The QRD-M algorithm [3] first decompose the channel matrix for each subcarrier to an upper triangular matrix using the QR decomposition to reduce the number of metric branches in the ML detector. Then, a limited tree-search is applied to approximate the ML detector with much reduced complexity. Despite of the significantly reduced complexity, the QRD-M algorithm is still the bottleneck in the receiver design, especially for the high-order modulation, high MIMO antenna configuration and large M . It is shown that even after the M-algorithm is written in C-MEX file, it alone takes up to 99% simulation time in a MIMO-OFDM simulation chain. It can take days or even weeks to generate one performance point. This becomes a critical bottleneck in research productivity.

B. Hardware Platform

FPGA can act as many parallel ASIC processors to provide mass parallelism. The proposed FPGA-based hardware accelerator platform has a PCMCIA compact form factor as shown in Fig. 2. Such a platform is intended for easy carry and demonstration in a portable laptop or wearable computer environment. It is PCMCIA CardBus release 8.0 compliant. This hardware accelerator contains a single Xilinx Virtex-II XCV3000-4 FPGA which supports designs of up to 300K gates with real-time speeds of up to 65 MHz. Besides, it contains ADC9235 analog to digital converter from analog device and 14 bits TTL digital I/O ports. It also provides 2 MB ZBT SRAM and 64 MB DRAM. A clock manager generates globally distributed clocks. This hardware platform is applied to achieve functional verification of the fixed-point hardware design, speed up the research activity for invention generation and rapidly prototype the VLSI architectures for proof of concept in the future real-time 4G prototyping system.

1) *Host Model*: To integrate with the matlab simulation chain, we design both the host model in the PC domain and the FPGA model in the hardware as shown in Fig. 3. In a matlab simulation chain, the matlab function calls a C-MEX file. The C-MEX file communicates with the WildCard by many API functions. In the WildCard development environment, there are many different classes of API functions. These include the card access API functions to open, close and get device information. We use the PE programming, writing and reading API functions to send and read data to/from the FPGA hardware. We can either use register files or Direct-Memory-Access (DMA) interface for data communication. To achieve DMA communication, DMA-related API functions are applied to do memory allocating, binding, unbinding etc. Finally, interrupt API functions provides status query, interrupt reset, enable as well as interrupt waiting functionalities. In the

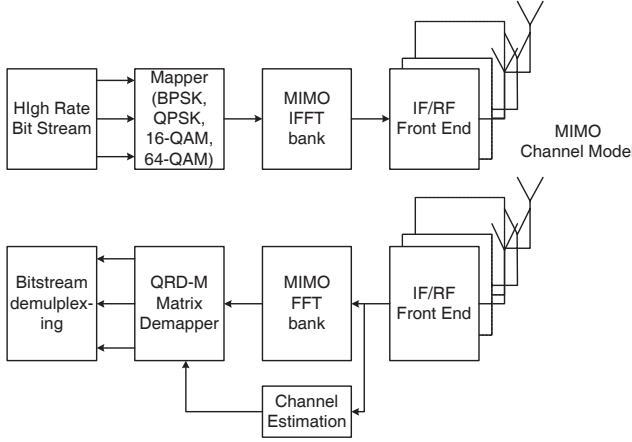


Fig. 1. System Model of the MIMO-OFDM Using Spatial Multiplexing.

domain OFDM subcarrier as

$$s_p(n) = \frac{1}{\sqrt{N_F}} \sum_{k=0}^{N_F-1} d_p(k) e^{j2\pi kn/N_F}. \quad (1)$$

After the insertion of the cyclic prefix, the time domain symbol is given by

$$\tilde{s}_p(n) = s_p(n), \quad \text{for } n \in [0, N_F - 1], \quad (2)$$

$$\tilde{s}_p(N_F + i) = s_p(N_F - 2 - i), \quad \text{for } i \in [0, N_g - 1] \quad (3)$$

where N_g is the number of subcarriers in the guard interval. $T_d^g = (N_F + N_g)T_s$ is the symbol period where T_s is the sampling period of the IFFT. Thus the analog signal transmitted at the p^{th} transmit antenna is given by

$$\tilde{s}_p(t) = \sum_n \tilde{s}_p(n) p(t - T_g - nT_d^g) \quad (4)$$

where $p(t)$ is the pulse shape function with support $[0, T_s)$ and $T_g = N_g T_s$ is the guard time. By passing the signal through a multipath fading channel and after sampling at the same rate, the received signal at the q^{th} receive antenna is given by

$$\begin{aligned} r_q(n) &= \sum_{p=1}^{N_T} \sum_{l=0}^{L_{p,q}} \tilde{s}_p(n-l) h_{p,q}(l) + v_q(n) \\ &= \frac{1}{\sqrt{N_F}} \sum_{p=1}^{N_T} \sum_{l=0}^{L_{p,q}} h_{p,q}(l) \sum_{k=0}^{N_F-1} d_p(k) e^{j2\pi kn/N_F} + v_q(n). \end{aligned} \quad (5)$$

where $L_{p,q}$ is the channel delay spread between the p^{th} transmit and the q^{th} receive antennas. $v_q(n)$ is the additive Gaussian noise to the q^{th} antenna. The MIMO channel is characterized by a matrix given by the Tapped Delay Line (TDL) model as

$$h_{p,q}(t) = \sum_{l=0}^{L_{p,q}} h_{p,q}(l) \delta(t - \tau_{p,q,l}) \quad (6)$$

where $\delta(t)$ is the Kronecker Delta function. $h_{p,q}(l) \in \mathbb{C}$ and $\tau_{p,q,l}$ are the amplitude and delay of the l^{th} path channel coefficient respectively. The cyclic prefix guard time satisfies $T_g > \max(\tau_{p,q,l})$ to eliminate the inter-symbol interference.

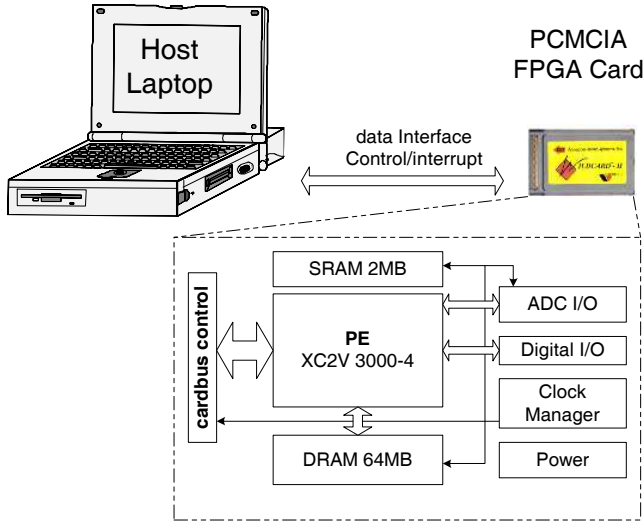


Fig. 2. The hardware block diagram of the PCMCIA FPGA card.

initialization phase, the card is opened and the implemented FPGA bit-stream file is programmed to the PE according to the simulation parameters such as MIMO antenna configuration, modulation scheme etc. The matlab function first passes the data to the C-MEX file with the same interface as the original C-MEX function for processing. The C-MEX function then passes the data to the DMA write API and waits for the hardware interrupt. Once the C-MEX file receives the interrupt, it triggers the DMA read function to get the processed data.

We can also apply a full system simulation environment written in VHDL for debugging designs at the development stage. The VHDL-based VHDL model encompasses all functionality within the card system. It provides an accurate test bed to validate the completed PE designs prior to synthesis. At this stage, the host computer is simulated through a set of VHDL procedures that map the actual C API functions, e.g. the register reading/writing, DMA reading/writing and interrupt handling. This makes the debugging process much easier.

2) *FPGA Model*: On the FPGA domain, the simulation environment consists of three different levels. The system level contains the board model and the board level contains the CardBus model, the PE model, SRAM and DRAM model as well as the clock model. The CardBus controller is used to handle all PCMCIA bus transactions, configure the clocks and handle the PE interrupts to the host. The board level also provides interconnections between PEs on a multi-PE board. Within the PE model, we integrate the PE interface and the actual user components for the chosen core algorithm. The purpose of the PE interface is to drive the PE FPGA pins. The interrupt management is also designed within the PE component. The PE design is completely synthesizable. The VHDL code of this level eventually becomes the bitstreams that could be programmed to the FPGA PE on the board level.

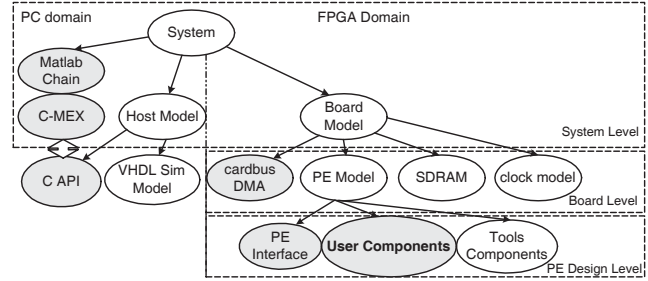


Fig. 3. The logic diagram of the hardware accelerator components.

C. Simulation-Emulation Co-design

To achieve simulation-emulation co-design, an efficient system-level partitioning of the MIMO-OFDM matlab chain is very important. The simulation chain is depicted in Fig. 4. Because the goal is for simulation time acceleration, we only need to implement the core algorithm with dominant complexity in FPGA hardware. In the simplified simulation model, the MIMO transmitter first generates random bits and map them to constellation symbols. Then the symbols are modulated by IFFTs. A multipath channel model distorts the signal and adds AWGN noises. The receiver part is contained in the function *fhqardqrdm_fpga_*, which consists the major subfunctions as demodulator using FFT, sorting, QR decomposition, the M-search algorithm in a C-MEX file, the de-mapping and the BER calculator. Because the M-search C-MEX file dominates more than 90% of the simulation time, the C-MEX file is re-designed in the FPGA hardware accelerator. The C APIs talk with the CardBus controller in the card board. The controller then communicates with the PE FPGA through the LAD Bus standard interface, which is part of the PE design. The data is stored in the input buffer and a hardware “start” signal is asserted by writing to the in-chip register. The actual PE component contains the core FPGA design to utilize both the multi-stage pipelining in the MIMO antenna processing and the parallelism in the subcarrier. After the output buffer is filled with detected symbols, the interrupt generator asserts a hardware interrupt signal, which is captured by the interrupt wait API in the C-MEX file. Then the data is read out from either DMA channel or status register files by the LAD output multiplexer. To achieve the bi-directional data transfer, both the source and destination DMA buffers are needed. Because the focus of this paper is not the VLSI architecture of the M-algorithm, the architecture detail is omitted here.

D. Design Cycle and Space Exploration

Although System-On-Chip (SoC) architectures offer more parallelism and flexibility to utilize the low-level silicon resource than DSP processors, the verification of SoC is a serious bottleneck in the design cycle because the current trial-and-optimize verification using hand-coded VHDL/Verilog or Graphical schematic design tools are falling behind requirement at an increasing rate. The System-C based high-level abstraction is also not intuitive to system engineers and re-

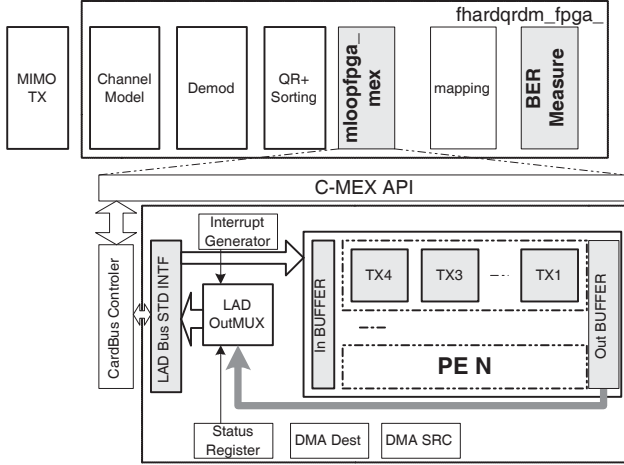


Fig. 4. The system partitioning of the MIMO-OFDM simu/emulation co-design and PE architecture of the M-algorithm.

quires very detailed hardware specification in the language. Extensive tradeoff study is very difficult from the manual parallelism/pipelining design and the still low-level hardware specifications. In this paper, we apply an un-timed C/C++ level verification methodology that integrates key technologies for truly high-level VLSI modelling. A Catapult-C based architecture scheduler is applied to explore the VLSI design space extensively. The major workload is transferred to the algorithmic C/C++ fixed-point design and high-level architecture scheduling. Synthesizable RTL is generated directly from a C/C++ level design and imported to the graphical tools for module binding. This significantly shortens the FPGA design cycle of the complex core algorithms.

IV. EMULATION RESULTS AND ACCELERATION

A. Emulation Performance

For a VLSI implementation, the algorithm needs to be converted to fixed-point numerical computations. The reduction of input bit-width directly leads to smaller size of the VLSI design. However, we need to also keep the performance almost the same as the floating-point design. To support a wide range of the simulation specification, we choose the input word length to be 10 bits. The bit-error rate (BER) performance for both the floating point simulation and the FPGA fixed-point emulation is shown in Fig. 5 for a 2×2 MIMO configuration with QPSK. The BER results for 4×4 configuration are shown in Fig. 6 for 16-QAM. It is shown that for the 2×2 case, $M = 4$ gives almost the same result as $M = 8$ and higher M . For the 4×4 scenario, $M = 16$ offers very small performance gain over $M = 8$ for the 16-QAM. In all cases, the measured FPGA BER curves almost overlap with the floating-point curves. This verifies that the fixed-point FPGA implementation is functionally correct.

B. Run-time Acceleration

According to the design area specification in terms of number of multipliers and slice usage, we can fit five 2×2

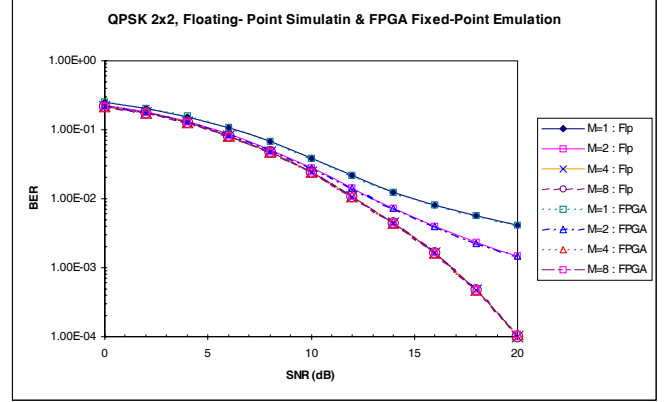


Fig. 5. Measured Bit Error Rate performance for fixed-point FPGA implementation: 2×2 , QPSK.

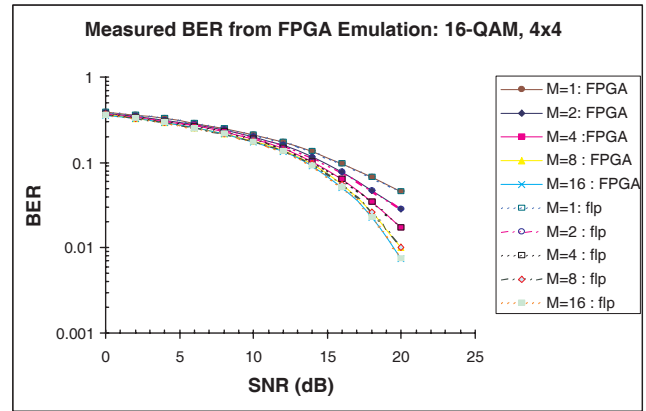


Fig. 6. Measured Bit Error Rate performance for fixed-point FPGA implementation: 4×4 , 16-QAM.

parallel processing elements (PE) or two 4×4 PEs in the V3000 FPGA. In this section, we analyze the run-time profile in a simplified 4G MIMO-OFDM simulation chain in matlab. The top five most time consuming functions in the simulation chain are shown in Fig. 7 and 8 for both the original C-MEX design and the FPGA implementation for 64-QAM and 2×2 configuration. The run-time is obtained by the matlab “*profile*” function. Function “*fhardqrdm_*” is the receiver function including all “*m_mex_orig*”, “*channel*”, “*qr*” and “*mapping*” sub-functions, where the QR-decomposition calls the matlab built-in function. It is shown that for the original floating-point C-MEX implementation, the C-MEX implementation of the M-search function “*m_mex_orig*” consumes most of the simulation time. Moreover, all the other functions consumes negligible time compared with the M-search function. So only the M-search function is implemented in FPGA hardware with the proposed complexity optimizations. In this case, 5 parallel processing elements are running in the Virtex-II V3000 at 33 MHz clock, which is competing with the 1.5 GHz clock of the PC processor. It is shown that the “*mloopfpga_mex*” now consumes a much smaller portion of the simulation time, which does not increase dramatically with higher M .

The run-times for the M-search function for both the orig-

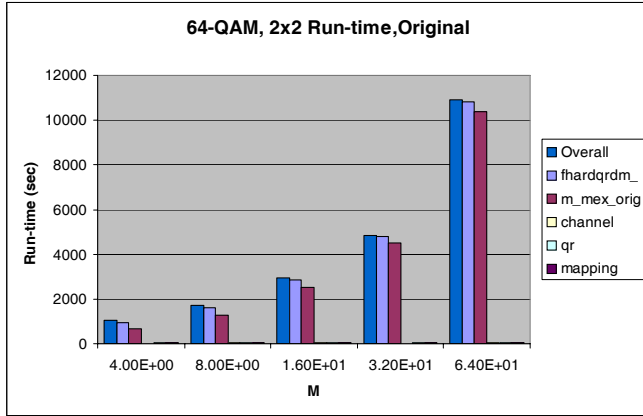


Fig. 7. Original floating-point C-MEX simulation profile: 2×2 , 64-QAM.

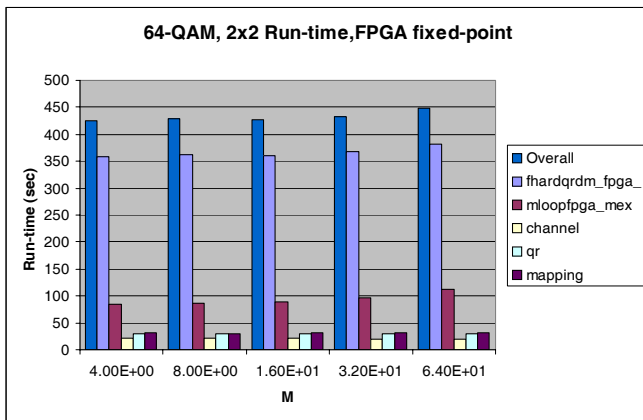


Fig. 8. FPGA fixed-point simu/emulation profile: 2×2 , 64-QAM.

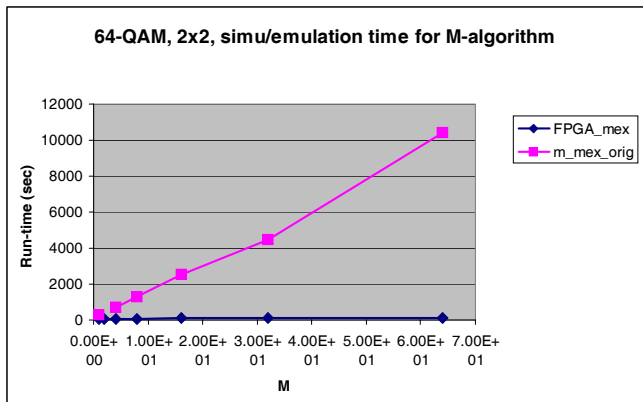


Fig. 9. Measured simulation speedup for the M-algorithm: 2×2 , 64-QAM.

inal floating point C-MEX design and the FPGA implementation are compared in Fig. 9. From this profile, the speed-up is around $100\times$. For the 4×4 case, we implemented two parallel processing elements in the V3000. The run-time is shown in Fig. 10 for the 64-QAM modulation. The achieved acceleration for all cases is obvious and significant.

In the integrated software/hardware co-design accelerator platform, speedup of $100\times$ is observed with 33 MHz FPGA clock rate competing with the 1.5 GHz Pentium-4 clock rate

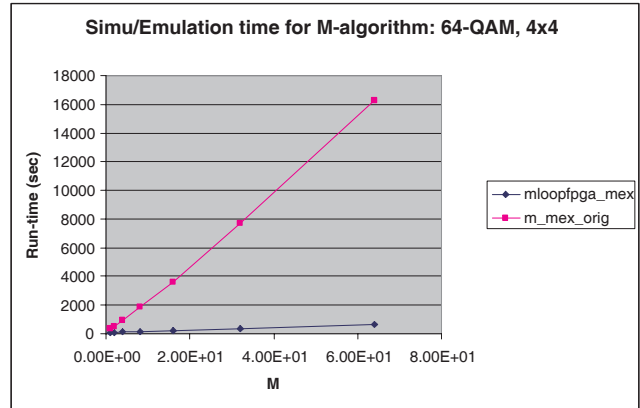


Fig. 10. Measured simulation speedup for the M-algorithm: 4×4 , 64-QAM.

for a $C = 64$ (64-QAM) and $M = 64$ system. Faster acceleration is achievable using more Processing Elements with the scalable VLSI architecture and clock rate from P & R result can be up to 90 MHz.

V. CONCLUSION

In this paper, we present a hardware accelerator for 4G simulation and functional verification with a PCMCIA compact form factor. The partitioned QRD-M VLSI architecture is implemented in the FPGA platform to accelerate the simulation time significantly. The speedup could be up to $100\times$ compared with the floating-point implementation for the conventional QRD-M algorithm. The compact form factor accelerator platform and design methodology is reusable for other critical algorithms.

ACKNOWLEDGMENT

The authors would like to appreciate Dr. Tony Reid in Nokia Research Center Dallas for the funding of this project and Dr. Kyeong Jin Kim for his help in providing the simulation chain.

REFERENCES

- [1] G. D. Golden, J. G. Foschini, R. A. Valenzuela and P. W. Wolniansky, *Detection algorithm and initial laboratory results using V-BLAST space-time communication architecture*, Electron. Lett., Vol. 35, pp.14-15, Jan. 1999.
- [2] Y. Li, J. H. Winters and N. R. Sollenberger, *MIMO-OFDM for wireless communications: signal detection with enhanced channel estimation*, IEEE Transactions on Communications, vol. 50, pp. 1471-1477, Sept. 2002.
- [3] K. J. Kim, J. Yue, R. A. Iltis and J. D. Gibson, *A QRD-M/Kalman filter-based detection and channel estimation algorithm for MIMO-OFDM systems*, IEEE Trans. Wireless Communications., to appear.
- [4] J. Yue, K. J. Kim, J. D. Gibson and R. A. Iltis, *Channel estimation and data detection for MIMO-OFDM systems*, IEEE Globecom, vol. 22, no. 1, pp. 581 - 585, Dec 2003.
- [5] G. H. Golub and C. F. V. Loan, *Matrix Computations*. The Jones Hopkins University Press, 1996.
- [6] Y. Guo, J. Zhang, D. McCain and J. R. Cavallaro, *Scalable FPGA architectures for LMMSE-based SIMO chip equalizer in HSDPA downlink*, 37th IEEE Asilomar Conference, Monterey, CA, 2003.
- [7] Y. Guo, G. Xu, D. McCain, J. R. Cavallaro, *Rapid scheduling of efficient VLSI architectures for next-generation HSDPA wire-less system using Precision-C synthesizer*, Proc. IEEE Intl. Workshop on Rapid System Prototyping'03, San Diego, CA, pp. 179-185, June 2003.