33rd International Conference on Acoustics, Speech, and Signal Processing. April, 2008. Las Vegas, Nevada. pp. 5388-5391.

ANALYZING THE SCALABILITY OF SIMD FOR THE NEXT GENERATION SOFTWARE DEFINED RADIO

Mark Woh, Yuan Lin, Sangwon Seo, Trevor Mudge, and Scott Mahlke

ACAL - University of Michigan - Ann Arbor

ABSTRACT

Previous studies have shown that wireless DSP algorithms exhibit high levels of data level parallelism (DLP). Commercial and research work in the field of software defined radio (SDR) has produced designs utilizing single-instruction multiple-data (SIMD) execution units to exploit this high level of parallelism. These designs have been able to deliver the efficiency and computational power needed to process 3G wireless technologies.

Though efficient 3G processing has been achieved, next generation 4G SDR technology requires 10-1000x more computational performance but limits the power budget increase to 2-5x. In this paper, we present a breakdown of 4G and analyze the scalability of SIMD to see if it can help to meet the 4G requirement. We take a proposed SDR architecture, SODA, and modify it for different widths in order to calculate its efficiency. We consider the trade-offs with respect to computation and energy efficiency.

Index Terms— software defined radio, single instruction multiple data, 4G wireless

1. INTRODUCTION

Wireless communication has grown dramatically over the years. Accessing the web, downloading video, and listening to music is a growing demand with wireless users. Third generation wireless (3G) technologies have been able to provide people with support for these services. With the number of users increasing and the demand for higher quality content, the bandwidth needed exceeds what 3G can provide. The Fourth Generation Wireless (4G) has been proposed by the International Telecommunications Union (ITU)[1] to increase the bandwidth to maximum data rates of 100 Mbps for high mobility situations and 1 Gbps for stationary and low mobility situations like internet hot spots. With this increase in bandwidth there will also be an increase in computation needed to processes this standard on SDR systems.

Baseband signal processing for mobile terminals has been a computing challenge for computer architects. Architecting systems that can perform the computations of 3G wireless systems has been successful in meeting the computation and power requirements demanded by these super computer like workloads. Though we were able to meet the requirements for 3G, the next generation 4G seems to be an even larger hurdle. With computational requirements increasing from 10-1000x and a power envelope that is limited to increasing only 2-5x[2], we need even more efficient designs to complete these tasks. We need more powerful processors but device scaling is delivering less in performance improvements.

ITRS [3] suggests that in future generations of process technologies, sub-90nm, we will still be able to scale frequencies higher. ITRS has been very optimistic with it's predictions and manufacturers in the past have been able to meet these targets. This is not the case anymore and current data shows that frequency and power are reaching a plateau[4]. The only benefit for changing technologies would be reduction in transistor size allowing us to pack more logic onto the same size die.

This slowdown in performance gain requires us to extract more computation with better architectural design. One way to extract more computational performance is by increasing the width of the datapath and exploiting more DLP through the use of SIMD. In our study, we analyze the effect that SIMD width has on computational performance and energy for 4G wireless systems. We will show the maximum possible DLP that can be extracted from each kernel and the percent of the algorithm which will benefit from exploiting the DLP.

This paper is organized as follows. In the next section, we present a simplified 4G system and describe some of the major kernels: an OFDM modulator/demodulator, a MIMO modulator/demodulator, and a channel decoder. In Section 3, we present the SDR architecture used for the SIMD study. In Section 4, we analyze the kernels and show their potential DLP and also the instruction breakdown of the algorithm. We also present the effects that varying SIMD width has on the computation and energy efficiency. The summary and concluding remarks are given in Section 4.

2. 4G WIRELESS KERNELS

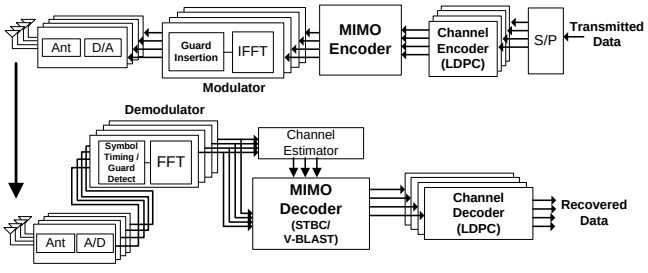

Fig. 1. Block Diagram Overview of a 4G System.

Though there is no standard yet for 4G, Figure 1 shows a high level block diagram of the physical layer of a NTT DoCoMo test 4G wireless system setup [5]. The major components of the physical

This research is supported by ARM Ltd. and the National Science Foundation grant NSF-ITR CCR-0325898 and CCR-0325761

layer consists of 3 major blocks: modulator/demodulator, MIMO encoder/decoder, and the channel encoder/decoder. These blocks compose the majority of the total computation in 4G. The role of the modulator is to map data sequences into amplitudes and phases which then are converted to the time domain and transmitted. The demodulator performs the operations in reverse order to reconstruct the original data sequences. This is typically done by the Fast Fourier Transform (FFT) algorithm.

The Multiple Input Multiple Output (MIMO) encoder multiplexes many data signals over multiple antennae. The MIMO decoder receives all the signals from the antennae and then tries to either decode all the streams for increased data rates or combine all the signals in order to increase the signal strength. The algorithm used to increase data rate is Vertical Bell Laboratories Layered Space-Time (V-BLAST) and the algorithm used to increase the signal quality is Space Time Block Codes (STBC). Finally the channel encoder and decoder performs forward error correction that enables receivers to correct errors in the data sequence without retransmission. There are many FEC algorithms which are used in wireless systems but LDPC is the most computationally intensive kernel and used for the highest data rates. LDPC has also been proposed in many standards like TGnSync and WWise [6] for IEEE 802.11n, which leads us to believe it may be used in 4G systems as well.

2.1. Fast Fourier Transforms (FFT and IFFT)

The transmission path uses an inverse FFT (IFFT) for modulation and the receiver uses an FFT for demodulation. We will only discuss the algorithm for FFT because IFFT is almost identical. The FFT operation consists of a data movement operation followed by a multiplication and addition on a complex number. If we assume an N-point radix-2 decimation in frequency (DIF) FFT, it consists of $\log_2 N$ stages. Between each stage, the N points of data are shuffled in a butterfly pattern. We used a radix-2 FFT because other FFT implementations have more complex shuffle patterns which require more cycles to implement even though the arithmetic may be simpler (as is the case with radix-4 FFT).

The vector width of FFT is equal to the number of data points, N, of the N-point FFT. For 4G, we perform 1024-point FFTs, meaning that the vector width is also 1024. A 1024-point FFT was used in the NTT DoCoMo test setup. FFT has a large amount of DLP because all 2-point FFT operations required for proceeding from one stage to the next can be done in parallel. This means that the SIMD can be utilized almost 100%, suggesting we should increase the SIMD width to be as large as the vector width of the algorithm in order to achieve maximum performance.

2.2. Space Time Block Codes (STBC)

As stated before, STBC is used to increase the signal quality. The same data is transmitted through each antenna but its representation is different for each antennae – the signal is transmitted through the multiple antennae in conjugate forms with different orderings. Signal quality is increased by receiving those redundant copies of the same data signal and optimally combines the information from each receiver to produce better quality estimations of the original data signal. The implementation we used is based on Alamouti's 2x2 scheme[7].

In the STBC encoder and decoder, the vector width is only 4 elements. Though the vector width is small, each data set is independent. This means that we can join many data sets together and process one large set. The set size would be limited only by the amount of data the FFT provides. In our case, this would suggest that a 1024 width data set would be most optimal.

2.3. Vertical Bell Laboratories Layered Space-time

V-BLAST is a spatial multiplexing scheme that improves the data rate by transmitting independent data streams over multiple antennae. This technique combines the multiple signals to obtain higher data rate rather than combine the same signal like STBC. The V-BLAST algorithm we implemented was based on work from [8], which reduces the computational complexity of V-BLAST.

Because our system uses a 4 transmit and 4 receive V-BLAST, the vector width of V-BLAST is 4 elements. The dimension of the channel matrix is 4x4 and the data signal is 4xN, where N is the number of data points in the FFT. The calculations performed are matrix operations between the channel matrix and the data. The operations are between 4 wide vectors. Though the vector width is only 4, we can exploit larger SIMD widths because we can process larger sections of the 4xN data signal. The algorithm itself can support SIMD widths up to 4N.

2.4. Low Density Parity-Check Codes (LDPC)

LDPC is an error correcting code that can perform closer to Shannon's limit than any other code. This means that LDPC can be used to achieve the highest data transmission rate possible over a wireless channel. LDPC is made up of only simple adds, subtracts and compares. LDPC has no serial dependency in operation unlike Turbo Codes that have to process SISO decoders serially after the interleaver. Our implementation of LDPC is based on [9] which was optimized for the SODA processor and 802.16e.

LDPC can be parallelized unlike other error correction codes. The vector width of the algorithm is related to the z size of the circulant shifted identity matrix *(zxz)*. The z value we used was 96 which corresponds to the maximum LDPC block size in 802.16e [10], which we assume is the highest data rate because it allocates the most number of subchannels. This means that we can benefit from SIMD widths up to 96 elements using this z value. After this limit there is no performance benefit. Unlike the other algorithms, we cannot overlap multiple z wide block rows of the LDPC matrix. This prevents us from utilizing even SIMD widths larger than 96 elements.

3. METHODOLOGY

We used the SODA architecture [11] as our SDR SIMD architecture to explore the scalability of SIMD width. A block diagram of the architecture is shown in figure 2. SODA is a control-data decoupled multi-core architecture. The SODA architecture consists of processing elements which uses a wide 512-bit SIMD unit that is capable of operating on 32 16-bit elements concurrently. SODA also has a non-uniform memory architecture, with local memories on the processing elements and a shared global memory. We used SODA because we could modify the implemented Verilog hardware model for different SIMD widths and modify the kernels' assembly code to support these multiple widths. All assembly code was written based

Algorithm	Overhead Workload (%)	Scalar Workload (%)	Vector Workload (%)	Vector Width (Elements)
FFT/IFFT	61	5	34	1024
STBC	14	5	81	4
V-BLAST	24	6	70	4
LDPC	33	18	49	96

Table 1. Data level parallelism analysis for major 4G kernels.

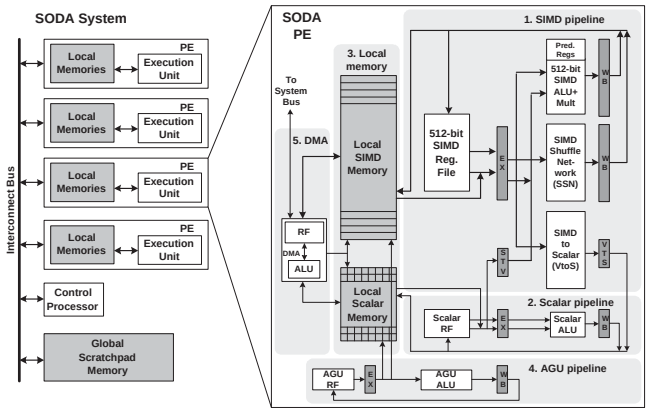

Fig. 2. SODA Architecture for SDR. The system consists of 4 data processing elements (PEs), 1 control processor, and global scratchpad memory, all connected through a shared bus. Each PE consists of a 32–wide 16–bit SIMD pipeline, a 16–bit scalar pipeline, two local scratchpad memories, an Address–Generation–Unit(AGU) for calculating memory addresses, and a Direct–Memory–Access (DMA) unit for inter-processor data transfer.

on implementations cited previously in SODA assembly. We synthesized SODA using Synopsys' Physical Compiler in 0.13 micron technology for 400MHz. Energy values were then extracted from the models and total energy was estimated based on the execution of the kernels.

4. RESULTS AND ANALYSIS

In Table 1, we analyze the DLP contained within the kernels. The instructions are broken down into 3 categories: overhead workload, scalar workload and vector workload. Overhead workload consists of all the instructions that assist SIMD computations, for example SIMD loads, stores and shuffle operations. The scalar workload consists of all the instructions that are not parallelizable and have to be run on the scalar unit. The vector workload consists of all the raw SIMD computations that use the ALU, multiplier, and shift units.

From the table, we can see that FFT is dominated by the overhead workload of loading the SIMD data and shuffling it. STBC and V-BLAST have a very high SIMD computation which suggests that these algorithms are dominated by raw computations and may be adaptable to very wide SIMDs. Finally, LDPC seems to have a mixture of all three types of instructions that suggests that performance may be limited by the non-vector workloads.

The table also presents the natural vector widths of the algorithms. Because we are using a 1024 point FFT the natural vector width is 1024. This means that this algorithm can support a SIMD of up to 1024 and still show performance improvement. This is very different from the other 3 algorithms whose vector widths are much narrower. For STBC and V-BLAST the vector widths may be small, but each set of elements are independent of each other allowing us to map multiple sets of 4 element computations onto any larger sized SIMD to increase performance. LDPC vector width may be its limiting factor because after a SIMD width of 96, the algorithm is constrained by the overhead and scalar workload, which prevents mapping of multiple sets of the vector elements. Any SIMD width larger than 96 will see no benefit in performance.

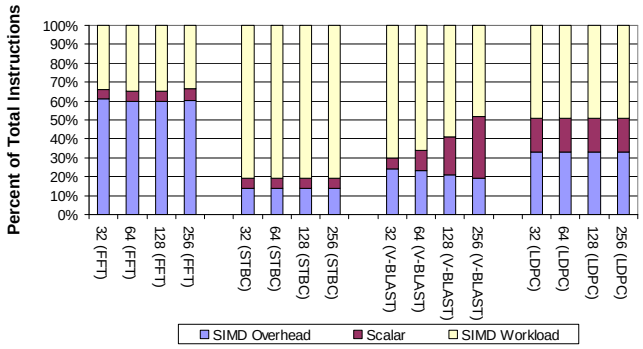

Fig. 3. Instruction breakdown of each kernel with respect to SIMD width.

We took each of the major kernels of the 4G system and mapped them onto wider versions of the SODA architecture. Most of the algorithms parallelized quite easily. FFT and STBC were especially easy to parallelize. As we can see in Figure 3, the instruction breakdown hardly changed from widths 32 to 256 because these algorithms are composed mainly of loops containing SIMD computations. By increasing the SIMD width, only the number of loop iterations changed. Lastly, V-BLAST is somewhat of an exception, because as we increase the width, the scalar instructions start to dominate. Thus the performance of V-BLAST will eventually be bounded by the scalar workload but, as we can see, the benefit of SIMD width can still provide major benefits.

In Figure 4, we show the normalized speed up of the kernels for different SIMD widths. All the values were normalized to the 32 wide SODA implementation. For comparison purposes, the 32 wide SODA implementation can perform FFT almost 10 times faster than the TI TMS320C6203. For FFT, STBC and V-BLAST, the speed up is linear with width: doubling the width, yields 2x return for each of the algorithms. The performance benefit of increasing SIMD width

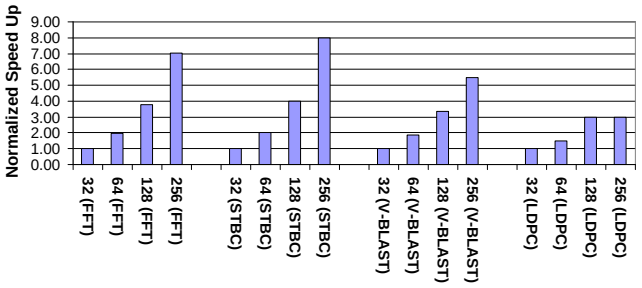

Fig. 4. Normalized Speedup of each kernel with respect to SIMD width.

is apparent. LDPC, though, is a different story. Because of the natural vector width of 96 we can only extract limited parallelism within the kernel. For SIMD widths greater than 128, there is no improvement in performance. The large jump between 64 and 128 width SIMD occurs because we cannot map all 96 values onto on a 64 wide SIMD machine. This forces us to do two iterations instead of the one possible with the 128 and 256 width SIMD. Because LDPC is the major performance bottleneck of 4G, this suggests that increasing SIMD alone may not meet the processing requirements.

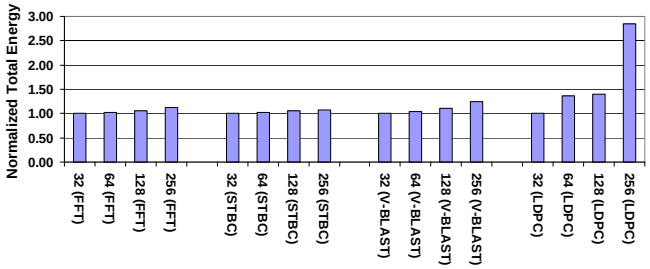

Fig. 5. Total energy consumption of each kernel with respect to SIMD width.

Finally in Figure 5, we show the energy consumption of each kernel for different widths. We computed the energy consumed taking into account leakage. As we can see, for FFT, STBC and V-BLAST there is a great benefit from increasing SIMD width. We get the greatest performance increase with reasonable increase in energy consumption. The exception again is LDPC. Going from 32 to 64, we take a large energy penalty mainly due to the fact that many SIMD lanes are wasted because the algorithm cannot map perfectly onto widths that don't divide into the natural vector width. Between 64 and 128, there is not much change because the algorithm is actually mapped relatively efficiently on the SIMD. The biggest jump is at 256 because all SIMD widths greater than 96 will waste energy doing unneeded work on the remaining 160 lanes.

5. CONCLUSION

Though the power and performance requirements of 4G is a significant challenge for designers, scaling SIMD width can help us gain major performance with only a modest increase in energy. We have seen almost a doubling of performance with doubling SIMD width. Not all kernels benefited from the increase in SIMD width. LDPC clearly is a major limiting factor in 4G. By increasing the SIMD width, FFT, STBC and V-BLAST benefits but LDPC benefits less. This suggests that LDPC may better be implemented on an accelerator or another specialized core.

SIMD scalability is just one factor in the processor design where we can extract more performance. With a combination of other architectural techniques, we may eventually be able to process 4G efficiently, within the power and performance requirements.

6. REFERENCES

[1] ITU-R M.1645, "Framework and overall objectives of the future development of IMT-2000 and systems beyond IMT-2000," *International Telecommuncations Union M.1645 Recommendation*, see http://www.ieee802.org/secmail/pdf00204.pdf.

[2] M. Woh, S. Seo, H. Lee, Y. Lin, S. Mahlke, T. Mudge, C. Chakrabarti and K. Flautner, "The next generation challenge for software defined radio," in *SAMOS*, Stamatis Vassiliadis, Mladen Berekovic, and Timo D. Hämäläinen, Eds. 2007, vol. 4599 of *Lecture Notes in Computer Science*, pp. 343–354, Springer.

[3] "International technology roadmap for semiconductors", see http://public.itrs.net.

[4] W. Haensch, E. Nowak, R. Dennard, P. Solomon, A. Bryant, O. Dokumaci, A. Kumar, X. Wang, J. Johnson and M. Fischetti, "Silicon cmos devices beyond scaling," *IBM J. of Research and Development*, vol. 50, no. 4/5, pp. 339–361, 2006.

[5] H. Taoka, K. Higuchi and M. Sawahashi, "Field Experiments on Real–Time 1-Gbps High–Speed Packet Transmission in MIMO–OFDM Broadband Packet Radio Access," *Vehicular Technology Conference, 2006. VTC* 2006-Spring. IEEE 63rd , vol.4, no., pp.1812-1816, 7-10 May 2006.

[6] T. Lestable and E. Zimmermann, "LDPC Options for Next Generation Wireless Systems," *Proceedings of the 14th Wireless World Research Forum* (WWRF), San Diego, CA, Jul 2005.

[7] S. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE J. on Select Areas in Communications*, vol. 16, no. 8, pp. 1451–1458, 1998.

[8] H. Zhu, Z. Lei, and F. Chin, "An improved square-root algorithm for blast," *IEEE Signal Processing Letters*, vol. 11, no. 9, pp. 772–775, 2004.

[9] S. Seo, T. Mudge, Y. Zhu, and C. Chakrabarti, "Design and analysis of LDPC decoders for software defined radio," in *Proc. IEEE SiPS*, Shanghai, China, Oct. 17-19 2007.

[10] IEEE Std 802.16e, "Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems", see http://standards.ieee.org/getieee802/download/802.16e-2005.pdf

[11] Y. Lin, H. Lee, M. Woh, Y. Harel, S. Mahlke, T. Mudge, C. Chakrabarti and K. Flautner, "SODA: A low-power architecture for software radio.," in *Proc. ISCA*, Boston, MA, June 17-21 2006, pp. 89–101.