Hindawi Publishing Corporation
VLSI Design
Volume 2011, Article ID 328640, 9 pages
doi:10.1155/2011/328640

Research Article

Efficient Resource Sharing Architecture for Multistandard Communication System

T. Suresh and K. L. Shunmuganathan

Department of CSE, R.M.K Engineering College, Anna University, Chennai 600025, India

Correspondence should be addressed to T. Suresh, fiosuresh@yahoo.co.in

Received 15 October 2010; Revised 8 February 2011; Accepted 29 March 2011

Academic Editor: Yangdong Deng

The Fourth Generation (4G) network is expected to serve mobile subscribers under dynamic network conditions and offer any type service: anytime, anywhere, and anyhow. Two such technologies that can respond to the above said services are Wideband Code Division Multiple Access (WCDMA) and Orthogonal Frequency Division Multiplexing (OFDM). The main contribution of this paper is to propose a dedicated hardware module which can reconfigure itself either to the OFDM Wireless LAN or WCDMA standard. In this paper, Fast Fourier Transform (FFT) algorithm is implemented for OFDM standard, and rake receiver is implemented for WCDMA standard. Initially efficient implementations of these two algorithms are tested separately and identified the resources utilized by them. Then the new hardware architecture, which configures to any one of these two standards on demand, is proposed. This architecture efficiently shares the resources needed for these two standards. The proposed architecture is simulated using ModelSimSE v6.5 and mapped onto a virtex 5 FPGA device (xc5v1x30ff324) using the tool Xilinx ISE 9.2i, and the results are compared with the standard approach. These results show that the proposed hardware architecture utilizes less number of resources compared to the conventional Reconfigurable Receiver Architecture System.

1. Introduction

Next generation wireless and mobile networks are all IP-based heterogeneous networks that allow users to use any system at anytime, anywhere, anyhow, and always-on connectivity in a seamless [1] manner. Users carrying an integrated open terminal can use a wide range of applications provided by multiple wireless networks and access to various air interface standards. The continuous evolution of wireless networks and the emerging variety of different heterogeneous, wireless network platforms with different properties require integration into a single platform [2]. The handoff mechanism allows a network connection on a mobile node to operate over multiple wireless access networks in a way that is completely transparent to end user applications. But there is no single system that is good enough to support all the wireless communication technologies. Instead of putting efforts in developing new radio interfaces and technologies for 4G [3] systems, we believe establishing 4G systems that integrate existing and newly developed wireless systems into one open platform is a more feasible option.

This has led to an increased interest in the design of reconfigurable architecture. The idea of the reconfigurable architecture is that it should be possible to alter the functionality of a mobile device at run time by simply reusing the same hardware for different wireless technologies and ultimately for users to connect to any system that happens to be available at any given time and place. This means that the same hardware should be able to handle many different modulation types as well as different demands on data rate and mobility.

2. Reconfigurable Architecture

There are several options available to implement the base-band signal processing section of wireless technologies. The possible options are the following.

(1) Multimode ASICs where device functionality can be switched according to the mode of operation.

(2) DSP-based implementation.

(3) Programmable logic-based implementation (using FPGA or PLD).

(4) Reconfigurable hardware [4–6].

Multistandard wireless communication applications demand high-computing power [7], flexibility, and scalability. An Application-Specific Integrated Circuit (ASIC) solution would meet the high computing power requirement, but is inflexible [8] and the long design cycle of ASICs makes them unsuitable for prototyping. On the other hand, general purpose microprocessors or Digital Signal Processing (DSP) chips are flexible, but often fail to provide sufficient computing power. Various processor architectures such as Very Long Instruction Word (VLIW) architecture [9] or vector processors have been introduced to increase the computing power, but the computing power is still insufficient or the architectures are too power hungry or expensive in silicon. Field Programmable Gate Arrays (FPGAs) [10] signal processing platforms are now widely being accepted in base_station designs. However, low power and form factor requirements have prevented their use in handsets. Reconfigurable hardware for Digital BaseBand (DBB) [11] processing is rapidly gaining acceptance in multimode handheld devices that support multiple standards. In Reconfigurable Hardware tasks that are nonoverlapping either in time domain or space domain can be mapped onto the same reconfigurable logic. Tasks that are required initially can be configured in the beginning. When another task is required, the configuration to load it can then be triggered. For example, in a typical smartphone environment, different wireless technologies, such as GSM, WCDMA, WLAN, and WiMax in the future, have to be supported [9]. However, it is not likely that these wireless technologies will be used at the same time. Therefore, it is possible to put them into reconfigurable logic and dynamically load the one that is needed. In this paper we present the design methodology for reconfigurable baseband signal processor architecture that supports WCDMA and OFDM wireless LAN standards.

In [12] the flexibility of the MONTIUM architecture was verified by implementing HiperLAN/2 receiver as well as a Bluetooth receiver on the same architecture. In [13], a broadband mobile transceiver and a hardware architecture which can be configured to any cyclic-prefix- (CP-) based system reconfigurable architecture for multicarrier-based CDMA systems is proposed. Reconfigurable Modem (RM) Architecture targeting 3G multistandard wireless communication system was proposed in [7]. This architecture targeted two 3G wireless standards WCDMA and CDMA 2000 and the design objectives are scalability, low power dissipation, and low circuit complexity. It is seen that though different functions can be reconfigured on a reconfigurable hardware, the major challenge is to have an efficient system configuration and management function which will initiate and control the reconfiguration as per the different application requirements.

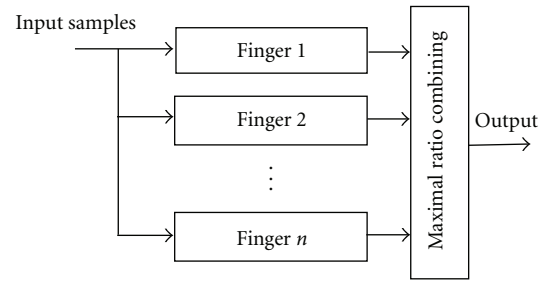


Figure 1: Components of the rake Receiver.

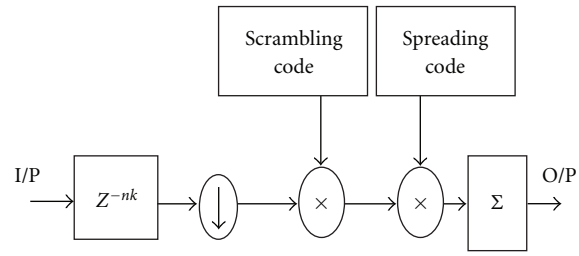


Figure 2: Schematic diagram of a rake finger.

3. Wideband Code Division Multiple Access (WCDMA)

Wideband Code Division Multiple Access (WCDMA) is a third-generation mobile wireless technology that is based on an ITU standard derived from CDMA [14–16] technology. WCDMA can support mobile/portable voice, images, data, and video communications up to 2 Mbps (local area access) or 384 kbps (wide area access). One of the more demanding functions on the WCDMA Baseband receiver module is the Rake Receiver [17]. A Rake Receiver allows each arriving multipath signal to be individually demodulated and then combined to produce a stronger and more accurate signal. Figure 1 shows the components of a Rake Receiver: a set of fingers and a combiner block.

The actual number of Rake fingers is not specified by WCDMA specifications but typically 4–8 fingers are employed. The demodulation is performed in the Rake fingers by correlating the received signal with a spreading code over a period corresponding to the spreading factor. Each Rake finger consists of two multipliers for applying the spreading and scrambling codes and an accumulator of length scrambling factor, input sample delay memory block, downsampling block, scrambling and spreading code generators. Figure 2 shows the schematic diagram of a Rake finger.

After the demodulation, maximal ratio combining is applied to the symbol dumps from the fingers. In maximal ratio combining, the phases of the symbols are aligned and their amplitudes are weighted according to the complex tap coefficients acquired by the complex channel estimator. After the combining, decision of the transmitted symbol is made and the resulting bit stream is de interleaved and decoded. The downlink scrambling code generator is a set

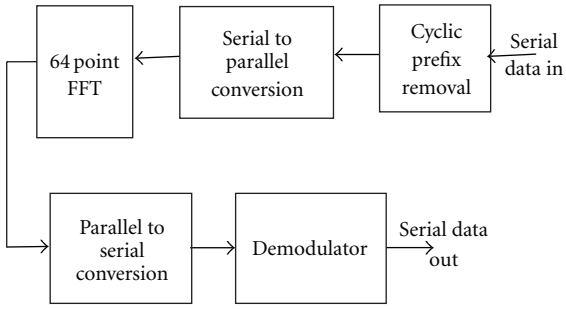


Figure 3: Simplified OFDM receiver block diagram.

of shift registers easy to implement in hardware or software. The number of scrambling code generators required for the RAKE Receiver depends on the receiver architecture chosen as well as on the number of simultaneous downlink cells supported and the number of fingers.

4. Orthogonal Frequency Division Multiplexing (OFDM)

Broadband WLAN standards (IEEE 802.11a/g, IEEE 802.16, Digital Audio Broadcast (DAB), and HIPERLAN/2) are based on the Orthogonal Frequency Division Multiplexing (OFDM) [18–20] modulation scheme because of its superior performance in various multipath environments, such as indoor wireless networks and metropolitan area network. OFDM can efficiently deal with multipath fading, channel delay spread, enhanced channel capacity, adaptively modifies modulation density and robust to narrowband interference. Figure 3 shows the basic block diagram for OFDM Receiver module. At the receiver, the received RF signal is down-converted to baseband frequency, digitized, and fed to the baseband section for further processing. Here the data is first cleaved off the cyclic prefix, followed by Fast Fourier Transform (FFT) [21, 22] operation and demodulated to obtain the received data bits. The key kernel in an OFDM receiver is the FFT processor. FFT-based processing is used to convert the signals from time domain to frequency domain and vice versa in OFDM modulation. The idea of using FFT instead of DFT is that the computation can be made faster where this is the main criteria for implementation. In direct computation of DFT the computation for **N**-point DFT will be calculated one by one for each point. But for FFT, the computation is done simultaneously and this method helps to save a lot of time. In WLAN standards it works with 64 carriers at a sampling rate of 20 MHz, so a 64-point FFT processor is required. 64 time domain samples represent the useful data part of the OFDM symbol that has to be demodulated.

The equations for FFT algorithm is written as

$$X(k) = \sum_{n=0}^{N\ 1} x(n) W_N^{nk}. \tag{1}$$

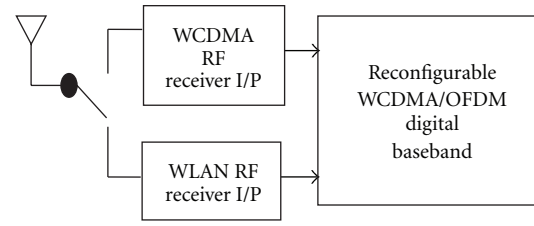


Figure 4: Block diagram of reconfigurable receiver system.

The quantity W_N^{nk} (called Twiddle Factor) is defined as

$$W_N^{nk} = e^{-j2\pi nk/N}. \tag{2}$$

This factor is calculated and put in a table in order to make the computation easier and can run simultaneously. The Twiddle Factor table is depending on the number of points used. During the computation of FFT, the factor does not need to be recalculated, since it can refer to the Twiddle factor table and thus it saves time.

5. Reconfigurable Receiver Architecture

Figure 4 shows the block diagram of reconfigurable receiver system. This receiver system is able to reconfigure itself to the WCDMA or OFDM Wireless LAN (WLAN) standard.

The proposed architecture comprises functional blocks, which in the form of reusable [13], reconfigurable [23–26] functional blocks for use in implementing different algorithms necessary for OFDM and WCDMA standards. One or more reusable functional blocks, can be configured to implement a process including multiplication, addition, subtraction, and accumulation. By accommodating the above-mentioned capabilities, the architecture should be configured to support WCDMA and WLAN OFDM Standards. For example, Fast Fourier Transform (FFT) (basic butterfly function) for WLAN OFDM and rake receiver algorithms. (multiply and accumulate select function) for WCDMA are implemented in the architecture as shown in Figure 5. Figure 6 shows the proposed reconfigurable architecture. The architecture includes the following:

(1) processing elements (PE) and their interconnects,

(2) a controlling block to control the functions of all the blocks,

(3) I/O block configured to receive preprocessed data, deliver processed data out, and determine the required functionality: a configurable I/O block that connects the chip with the outside world;

(4) memory block: configured to store the compiled software instructions and to cache and buffer data.

This architecture allows for transformation of the chip from WCDMA chip to WLAN Wi-Fi chip on demand wherein new algorithms can be accommodated on chip in real time via different control sets.

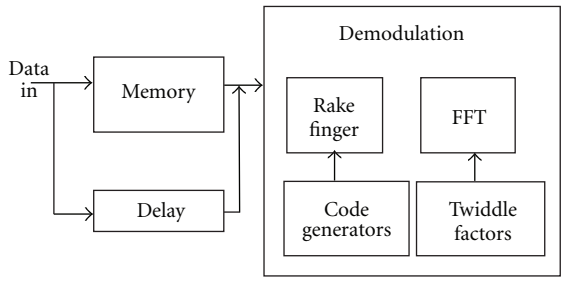


Figure 5: Functional block of reconfigurable receiver algorithms.

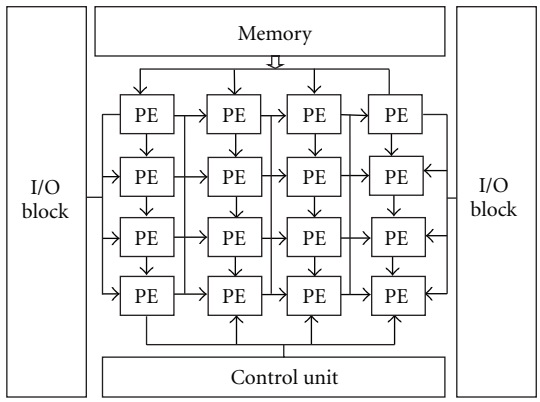


Figure 6: The proposed reconfigurable architecture.

5.1. Rake Finger Implementation. In WCDMA receivers, the demodulation is performed in the Rake fingers by correlating the received signal with a spreading code over a period corresponding to the spreading factor. The output of the ith Rake finger can be expressed as

$$O_i(n) = \sum_{i=0}^{L_{sf}-1} C_s(i + nL_{sf})R(i + nL_{sf}), \tag{3}$$

where C_s is the combined spreading and scrambling code and R is the received signal and both are complex numbers [7]. Since the scrambling and spreading codes are always of ± 1, the multiplication and addition of each correlation stage are simplified. So (3) is simplified to

$$(R_r + jR_i)(C_{sr} - jC_{si}) = (R_rC_{sr} + R_iC_{si}) + j(R_iC_{sr} - R_rC_{si}). \tag{4}$$

If the value +1 is represented as logic "0" and the value −1 is represented as logic "1", (4) is simplified as follows:

$$= \begin{cases} R_r + R_i + j(R_i - R_r), & \text{when } C_{sr} = 0,\ C_{si} = 0, \\ R_r - R_i + j(R_i + R_r), & \text{when } C_{sr} = 0,\ C_{si} = 1, \\ -(R_r - R_i) - j(R_i - R_r), & \text{when } C_{sr} = 1,\ C_{si} = 0, \\ -(R_r + R_i) - j(R_i - R_r), & \text{when } C_{sr} = 1,\ C_{si} = 1. \end{cases} \tag{5}$$

Since the code input is binary valued, the complex multiplication in the correlations is simplified to one real addition/subtraction and one imaginary addition/subtraction.

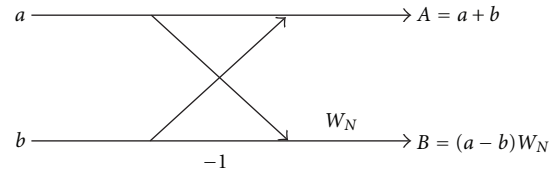


Figure 7: 2 Point butterfly structure.

Selection of addition or subtraction is done with the help of multiplexer. So the total resources required to implement Rake Receiver using (5) are two adders, two subtractors, and one multiplexer.

5.2. FFT Implementation. In OFDM, the demodulation is performed by applying 64-point FFT. The twiddle factor is calculated and put in a table in order to make the computation easier and can run simultaneously. The Twiddle Factor table is depending on the number of points used. During the computation of FFT, this factor does not need to be recalculated since it can refer to the twiddle factor table, and thus it saves time. Figure 7 shows the 2-point Butterfly structure [27] where multiplication is performed with the twiddle factor after subtraction.

Multiplication is certainly the most vital operation in communication processing, and its implementation in an integrated circuit component requires large hardware resources and significantly affects the size, performance, and power consumption of a system [28]. So an efficient way of multiplier reduction in FFT processing is done as follows. Consider the problem of computing the product of two complex numbers R and W

$$\begin{aligned} X = RW &= \left(R_r + jR_i\right)\left(W_r + jW_i\right) \\ &= (R_rW_r - R_iW_i) + j(R_rW_i + R_iW_r). \end{aligned} \tag{6}$$

From (6), the direct architectural implementation requires total of four multiplications and one real subtraction and one imaginary addition to compute the complex product. However, by applying the Strength Reduction Transformation we can reformulate (6) as

$$\begin{aligned} X_r &= (R_r - R_i)W_i + R_r(W_r - W_i), \\ X_i &= (R_r - R_i)W_i + R_i(W_r + W_i). \end{aligned} \tag{7}$$

As can be seen from (7), by using the Strength Reduction Transformation the total number of multiplications is reduced to only three. This however is at the expense of having two additional subtractors and one adder.

5.3. Design Procedure and Mapping Tool. Figure 8 shows the proposed tool chain for application mapping which is similar to the conventional ones, however it has been customized for this design. This design typically starts with the application codes written in VHDL. This abstract design is optimized to fit into the FPGAs available logic

through a series of steps. The basic functionality of this flow is to generate configuration bit streams and an executable code for the reconfigurable architecture. Initially DFGs (Data Flow Graph) are extracted manually, however automatic generation is possible for some application. The extracted DFGs possess a large number of operations which necessitates a sophisticated mapping tool to locate DFGs onto the proposed reconfigurable architecture. Applying architectural constraints, the DFGs are mapped onto the architecture through placing DFG nodes on the PE array and routing interconnection as well as assigning input/output nodes to the proper I/O ports. Our placement algorithm uses the well-known conventional method for placing DFG nodes on the PEs. Routing process establishes connections between the source and target PEs of Reconfigurable Array, by means of routing resources. The main objective of this process is to reduce the connection length between PEs. Placement is not limited to only one row of PE array but, the nodes can be distributed over the entire PE array. Placement and routing procedures ought to be iterated until a valid mapping satisfying the architectural constraints are generated. The total number of required PE in the PE array, the number of PEs in each row, the number of rows, the number of input/output ports, and the suitable routing resources are the essential architectural constraints for this design process. Configuration's bit_stream associated with each node of DFGs can be generated after completion of the mapping stage. For logic and layout synthesis the Xilinx ISE toolset can be utilized. The configuration controller (processor embedded in the FPGA) controls the execution of the partitioned basic block by enabling the proper configurations (bit_streams) on the FPGA according to application's data and control flow. Data memories embedded in the FPGA can be used for storing the input and output values. The virtex 5 FPGA device is used as the design platform since this architecture provides more resources and an improved logic count with respect to the previous family. A set of configuration registers defines the state of this configuration logic at a given moment in time. Configurations are the memory where the bit_stream file has direct access. Actual configuration data are first written by the bit_stream into these registers and then copied by the configuration logic on the configuration SRAMs.

5.4. Reconfiguration Overhead. To be practical, RTR systems must insure that the time spent performing reconfiguration is negligible with respect to the time spent performing applications. If the system is reconfigured too frequently, more time is spent reconfiguring than performing applications. The time wasted in performing reconfiguration is called reconfiguration overhead. The reconfiguration overhead in dynamically reconfigurable systems is a major problem that affects the total execution time. The three methods (beyond the scope of this paper) used to minimize the reconfiguration overhead are configuration reuse, configuration prefetch, and configuration replacement. Configuration reuse allows different applications to use the same configured hardware task so that the number of configurations is reduced.

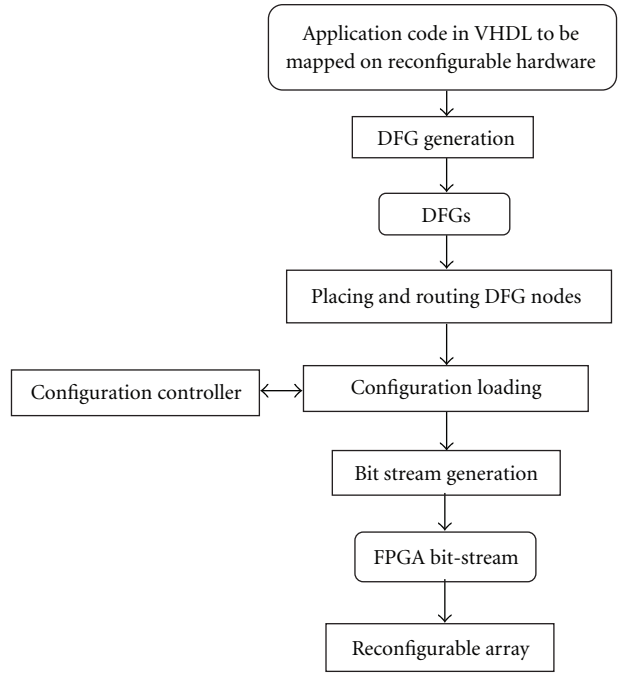


Figure 8: Application mapping tool chain.

Configuration prefetch allows hardware tasks to be configured well before it is required so that the configuration latency is hidden or overlapped with other hardware or software executions. The replacement technique increases the possibilities of reusing those subtasks that are more critical for the system performance.

5.5. Debugging the Reconfigurable Array. After mapping the application in reconfigurable array, this FPGA design can be debugged using simulators. However, we verified this design directly, by downloading them into an FPGA and then testing them in a system. This testing provides a stronger form of functional verification than simulation. Simulations for debugging purposes were carried out with ModelSim SE v6.5 from Mentor Graphics. The Xilinx 9.2i tool set was used to compile the VHDL code and then the design was placed and routed.

6. Processing Element

Figure 9 shows the Processing Element (PE) and its resources required for the implementation of FFT in WLAN OFDM and Figure 10 shows the Processing Element (PE) and its resources required for the implementation of Rake finger in WCDMA. It is shown that the two adders and subtractors (red coloured) are shared by both the standards.

So the proposed reconfigurable architecture consists of processing units, their computational elements are shared by both the Rake Receiver operation of WCDMA and FFT operation of OFDM. The processing units perform the multiply accumulate operation in the Rake mode as described in Section 5.1 and butterfly operations in the FFT mode as described in Section 5.2. The computational

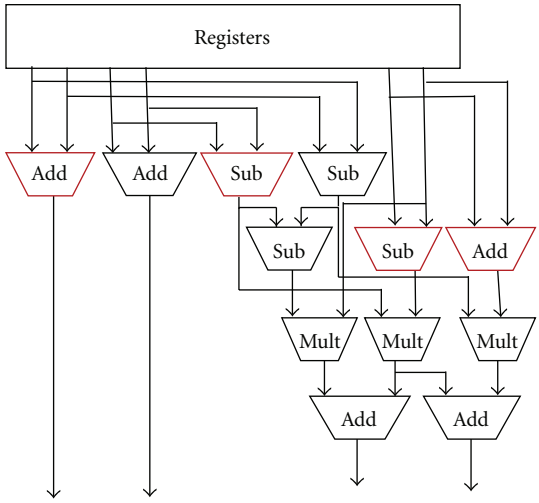


Figure 9: PE and its resources of WLAN OFDM.

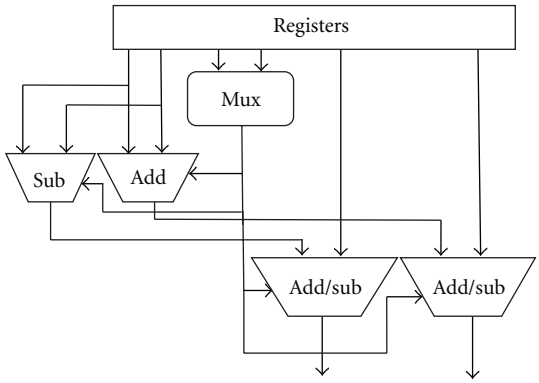


Figure 10: PE and its resources of WCDMA.

resources required by the proposed architecture are 5 adders, 4 subtractors, and 3 multipliers and multiplexers.

7. Results and Discussions

The proposed architecture described in Sections 5 and 6 was simulated using ModelSimSE v6.5, coded in VHDL and mapped [29, 30] onto a Virtex 5 FPGA device (xc5v1x30ff324) with speed grade (−3) using the tool Xilinx ISE 9.2i and synthesized. The metrics extracted from Xilinx ISE after synthesis and implementation are the followings: (1) the number of used Slice LUTs (Look Up Tables) and (2) gate count. Table 1 shows the implementation results of our optimized Strength Reduction Transformation Technique of FFT algorithm. These results are obtained using the design scheme explained in Section 6. The comparison results as shown in Figure 11 shows that our proposed Strength Reduction Transformation Technique of FFT algorithm has significant improvements in terms of area saving compared to the standard implementation. On average, our optimized approach shows an improvement about 15.93% in terms of number of gates used and 19.27% in terms of number of Slice LUTs used, as illustrated in Table 4.

Table 1: Resource utilization of conventional FFT and FFT with strength reduction transformation technique.

Resources utilized	Conventional FFT	FFT with strength reduction transformation technique
Number of slice LUTs	1588 out of 19200	1282 out of 19200
Gate count	14796	12440

Table 2: Resource utilization of conventional rake finger and multiplier-less rake finger.

Resources utilized	Conventional rake finger	Rake finger (multiplier-less)
Number of slice LUTs	1426 out of 19200	128 out of 19200
Gate count	13898	1328

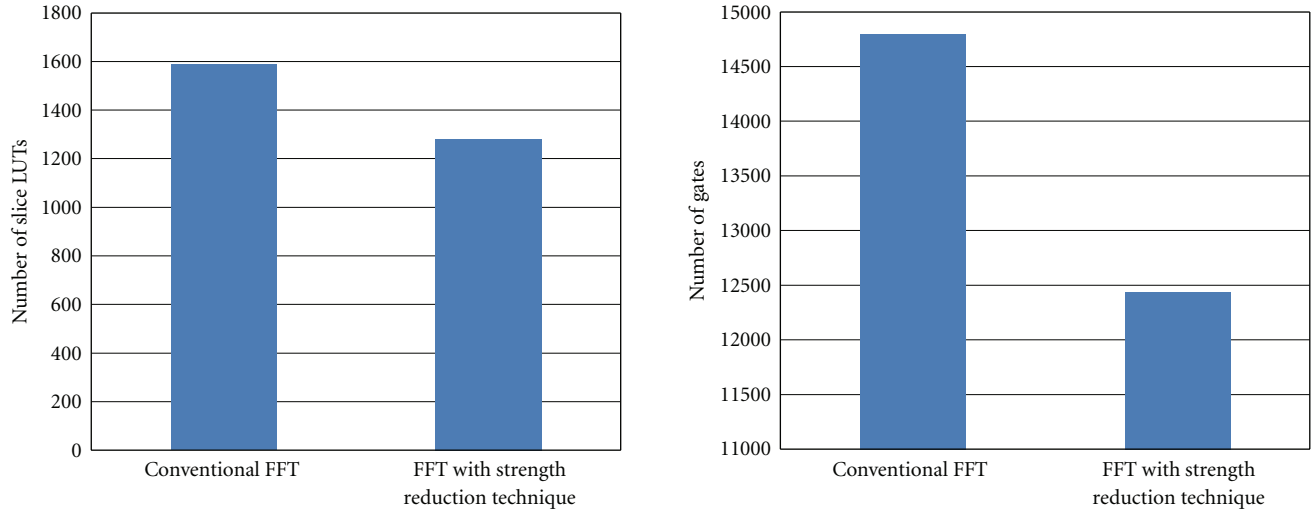
Table 3: Resource utilization of reconfigurable architecture without and with resource sharing.

Resources utilized	Reconfigurable architecture without resource sharing	Reconfigurable architecture with resource sharing
Number of slice LUTs	3014 out of 19200	1290 out of 19200
Gate count	28694	12540

Table 2 and Figure 12 illustrate the comparison of the results of our optimized multiplier-less Rake finger implementation and the Conventional Rake finger. From this result it is clear that our proposed approach outperformed the conventional Rake finger in terms of the number of Slices LUTs used and the number of gates utilized. On average, our optimized approach shows an improvement of about 91.03% and 90.45% in terms of number of Slice LUTs used and the number of ates utilized, as illustrated in Table 4.

Finally the implementation results of the proposed reconfigurable architecture with Resource sharing are compared with Reconfigurable Architecture without resource sharing techniques. Table 3 and Figure 13 show the comparison results of the proposed Architecture (Reconfigurable Architecture with Resource sharing) and the Reconfigurable Architecture without Resource sharing in terms of the number of Slice LUTs used and the number of gates utilized. The average reductions for the parameters are 57.2% and 56.3%, respectively, as shown in Table 4. From this result it is clear that our proposed Resource sharing Reconfigurable Architecture is better than the standard Reconfigurable Architecture.

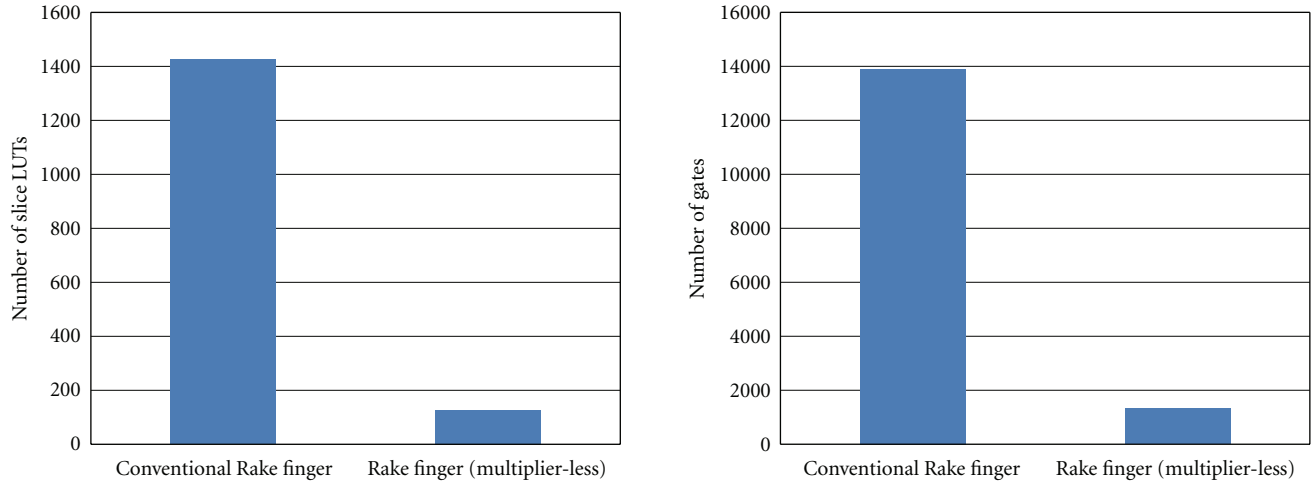
From the results presented earlier it seems that there is a significant reduction in large number of computational resources which forms the proposed architecture which is more efficient than the conventional architecture. These efficient realizations require fewer FPGA resources, so not



(a) Comparison of the Slice LUTs used

(b) Comparison of the number of gates utilized

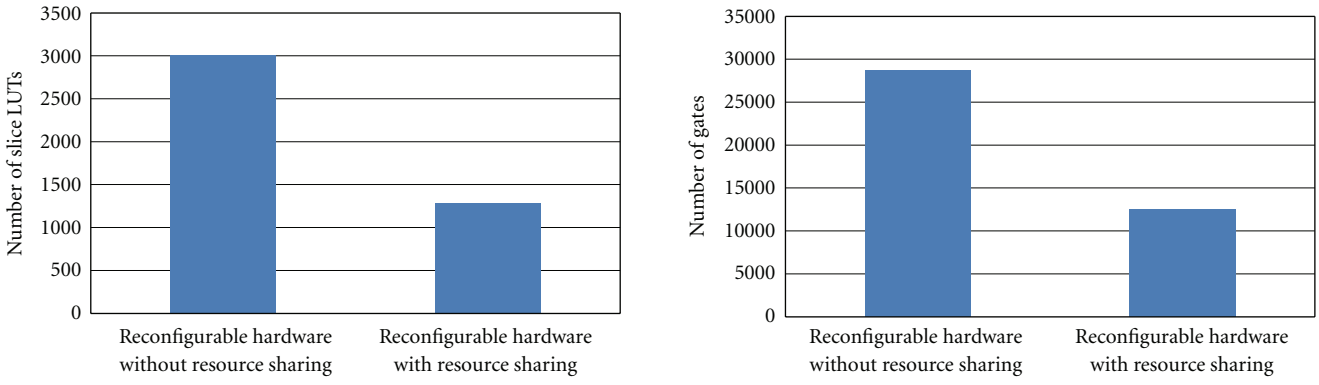
Figure 11: Comparison results of resources utilized by conventional FFT and FFT with strength reduction transformation technique.



(a) Comparison of the Slice LUTs used

(b) Comparison of the number of gates utilized

Figure 12: Comparison of the percentage of resources utilized by conventional rake finger and rake finger (multiplier-less).



(a) Comparison of the Slice LUTs used

(b) Comparison of the number of gates utilized

Figure 13: Comparison of the percentage of resources utilized by reconfigurable architecture without resource sharing and with resource sharing.

Table 4: Comparison of results of the conventional architecture and the proposed method targeting Virtex 5 FPGA.

Proposed method	Gate reduction	Slice LUTs saving
FFT with strength reduction transformation technique	15.93%	19.27%
Rake finger (multiplier-less)	90.45%	91.03%
Reconfigurable hardware with resource sharing	56.3%	57.2%

only reduce the area, but reduce the total power dissipation also.

8. Conclusion

An architecture which can reconfigure itself to wireless LAN OFDM and WCDMA standards, was presented in this paper. While configuring these two standards, it was also presented to implement FFT operation for OFDM and Rake Receiver functioning for WCDMA efficiently. To lower the number of multipliers in FFT and eliminate the multipliers in Rake Receiver, we adopted Strength Reduction Transformation technique and multiplier-less technique. The proposed architecture was simulated using ModelSimSE v6.5 and mapped onto a Xilinx Virtex 5 FPGA device and synthesis report was generated. Substantial improvements in terms of number of Slice LUTs used and the number of gates utilized were achieved. Comparison results showed that the proposed architecture can reduce large number of FPGA resources, enhance efficiency of the hardware architecture, and significantly reduce area and power consumption. Moreover, the proposed architecture can be improved to reconfigure to various other advanced wireless standards.

References

[1] R. Hirschfeld, W. kellerer, C. Prehofer, and H. Berndt, "An integrated system concept for future generation mobile communication," in *Proceedings of the Eurescom Summit on Powerful Networks for Profitable Services (EURESCOM '02)*, Germany, October 2002.

[2] L. M. Gavrilovska and V. M. Atanasovski, "Interoperability in future wireless communications. Systems: a roadmap to 4G," *Microwave Review*, vol. 13, no. 1, 2007.

[3] V. Gazis, N. Houses, A. Alonistioti, and L. Merakos, "Generic system architecture for 4G mobile communications," in *Proceedings of the 57th IEEE Semiannual Vehicular Technology Conference (VTC '03)*, pp. 1512–1516, April 2003.

[4] S. Hauck and A. Detlon, *Reconfigurable Computing the theory and Practice of FPGA-Based Computation*, Elsevier, New York, NY, USA, 2008.

[5] M. Gokhale and P. S. Graham, *Reconfigurable Computing Accelerating Computation with Field-Programmable Gate Arrays*, Springer, London, UK, 2005.

[6] R. Tessier and W. Burleson, "Reconfigurable computing for digital signal processing: a survey," *Journal of VLSI Signal Processing*, vol. 28, no. 1-2, pp. 7–27, 2001.

[7] J. S. Lee and D. S. Ha, "FleXilicon: a reconfigurable architecture for multimedia and wireless communications," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '06)*, pp. 4375–4378, May 2006.

[8] J. Kim, D. S. Ha, and J. H. Reed, "A new reconfigurable modem architecture for 3g multi-standard wireless communication systems," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '05)*, pp. 1051–1054, May 2005.

[9] A. S. Y. Poon, "An energy-efficient reconfigurable baseband processor for wireless communications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 3, pp. 319–327, 2007.

[10] R. David, D. Chillet, S. Pillement, and O. Sentieys, *A Compilation Framework for a Dynamically Reconfigurable Architecture*, vol. 2438, Springer, London, UK, 2002.

[11] L. Harju and J. Nurmi, "A programmable baseband receiver platform for WCDMA/OFDM mobile terminals," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '05)*, pp. 33–38, New Orleans, La,USA, March 2005.

[12] G. K. Rauwerda, P. M. Heysters, and G. J.M. Smit, "Mapping wireless communication algorithms onto a reconfigurable architecture," *Journal of Supercomputing*, vol. 30, no. 3, pp. 263–282, 2004.

[13] Y. C. Liang, S. Naveen, S. K. Pilakkat, and A. K. Marath, "Reconfigurable signal processing and hardware architecture for broadband wireless communications," *EURASIP Journal on Wireless Communications and Networking*, vol. 2005, no. 3, pp. 323–332, 2005.

[14] J. Becker and M. Glesner, "A parallel dynamically reconfigurable architecture designed for flexible application-tailored hardware/software systems in future mobile communication," *Journal of Supercomputing*, vol. 19, no. 1, pp. 105–127, 2001.

[15] Y. Guo, D. McCain, J. R. Cavallaro, and A. Takach, "Rapid industrial prototyping and SoC design of 3G/4G wireless systems using an HLS methodology," *EURASIP Journal on Embedded Systems*, vol. 2006, Article ID 14952, pp. 1–25, 2006.

[16] R. Sivasubramanian and A. Varadhan, "An efficient implementation of IS-95A CDMA transceivers through FPGA," *International Journal on Digital Signal Processing*, vol. 6, no. 1, pp. 23–30, 2006.

[17] A. Alsolaim, J. Becker, M. Glesner, and J. Starzyk, "Architecture and Application of a dynamically reconfigurable hardware array for future mobile communication systems," in *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines*, August 2002.

[18] J. S. Park, H.-J. Jung, and V. K. Prasasnna, "Efficient FPGA-based implementation of the MIMO-OFDM physical layer," in *Proceedings of the World Congress in Computer Science Computer Engineering and Applied Computing (WORLDCMP '06)*, World Academy of Science, Las Vegas, Nev, USA, June 2006.

[19] J. Helmschmidt, E. Schuler, P. Rao, S. Rossi, S. di Matteo, and R. Bonitz, "Reconfigurable signal processing in wireless terminals," in *Proceedings of the Conference on Design, Automation and Test in Europe (DATE '03)*, March 2003.

[20] C. Ebeling, C. Fisher, G. Xing, M. Shen, and H. Liu, "Implementing an OFDM receiver on the RaPiD reconfigurable architecture," *IEEE Transactions on Computers*, vol. 53, no. 11, pp. 1436–1448, 2004.

[21] M. J. Myjak and J. G. Delgado-Frias, "A medium-grain reconfigurable architecture for DSP: VLSI design, benchmark mapping, and performance," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 1, pp. 14–23, 2008.

[22] B. Zhou, Y. Peng, and D. Hwang, "Pipeline FFT architectures optimized for FPGAs," *International Journal of Reconfigurable Computing*, vol. 2009, Article ID 219140, 9 pages, 2009.

[23] S. Hauck, T. W. Fry, M. M. Hosler, and J. P. Kao, "The chimaera reconfigurable functional unit," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 2, pp. 206–217, 2004.

[24] H. Parizi, A. Niktash, A. Kamalizad, and N. Bagherzadeh, "A reconfigurable architecture for wireless communication systems," in *Proceedings of the 3rd International Conference on Information Technology: New Generations (ITNG '06)*, pp. 250–254, April 2006.

[25] R. Hartenstein, "Coarse grain reconfigurable architectures," in *Proceedings of the Conference on Asia South Pacific Design Automation*, pp. 564–570, January 2001.

[26] Y. Qu, K. Tiensyrjä, J. P. Soininen, and J. Nurmi, "Design flow instantiation for run-time reconfigurable systems: a case study," *EURASIP Journal on Embedded Systems*, vol. 2008, no. 1, Article ID 856756, 2008.

[27] P. Heysters, G. Smit, and E. Molenkamp, "A flexible and energy-efficient coarse-grained reconfigurable architecture for mobile systems," *Journal of Supercomputing*, vol. 26, no. 3, pp. 283–308, 2003.

[28] H. Hinkelmann, P. Zipf, J. Li, G. Liu, and M. Glesner, "On the design of reconfigurable multipliers for integer and Galois field multiplication," *Microprocessors and Microsystems*, vol. 33, no. 1, pp. 2–12, 2009.

[29] F. Hannig, H. Dutta, and J. Teich, "Regular mapping for coarse-grained reconfigurable architectures," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. V-57–V-60, May 2004.

[30] M. D. Galanis, G. Dimitroulakos, and C. E. Goutis, "Speedups and energy reductions from mapping DSP applications on an embedded reconfigurable system," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 12, pp. 1362–1366, 2007.