Sigma Journal of Engineering and Natural Sciences
Web page info: https://sigma.yildiz.edu.tr
DOI: 10.14744/sigma.2025.00007

Research Article

Analysis and floor plan of multimedia design for the internet of things

Ipseeta NANDA[1,*], J. MIDHUNCHAKKARAVARTHY[2]

[1]*Department of Computer Science, Lincoln University, Kota Bharu, 1570, Malaysia*
[2]*Faculty of Computer Science and Multimedia, Lincoln University College, Kota Bharu, Malaysia*

ARTICLE INFO

Article history
Received: 08 November 2023
Revised: 15 January 2024
Accepted: 15 February 2024

Keywords:
Audio Communication System Architecture; Design Rule Check; Dynamic Partial Reconfiguration; Internet of Things; Multimedia Communication System Architecture

ABSTRACT

A platform for computation and communication is planned to support real time multimedia communication system architecture (MCSA) model for Internet of Things (IoT) application. In the current scenario, reducing the chip area plays an important role in chip design, where lower power consumption becomes significant consideration. We applied a Dynamic Partial Reconfiguration (DPR) to design a processor peripheral model, achieving area reduction and power efficiency. Furthermore, the Audio Communication System Architecture (ACSA) filter design plays an important role in multimedia systems, demonstrating greater feasibility in hardware implementation. The objective of the paper is to design flexible system for different application areas with the use of partial reconfiguration technique. This allows for enhanced application performance. Utilizing the Design Rule Check (DRC), we achieve high level design on Xilinx platform. In this work, we compared Audio Mutimedia Architecture ACSA design with and without DPR with area and power. Our work has given better performance with respect to area reduction and low power consumption with respected to existing methods.

Cite this article as: Nanda I, Midhunchakkaravarthy J. Analysis and Floor Plan of Multimedia Design for the Internet of Things. Sigma J Eng Nat Sci 2025;43(1):88−95.

INTRODUCTION

Introduction to FPGA adapts on-the-fly configuration. The main characteristic is that the processor designed provides flexibility and performance that can adapt to the hardware application. The Partial Reconfiguration (PR) offers benefits for multiple applications which support industries [1,2]. The modules that are used for applications are pre-synthesized. Reconfigurable hardware is an aspect of FPGA interest in IoT to be feature-proof without any hardware modification [3,4]. FPGA properties are required for interfacing with the Internal Configuration Access Port, Program Memory, hyper-terminal, which interfaces with PC. The ML401 FPGA includes processors, tri-mode Ethernet MACs, and a serial transceiver of 6.5Gb/s. IoT connects the physical objects with an IP address. It interacts, and data are stored in the cloud, or it interacts with another connected device around the world. Machine-to-Machine data gathered from IoT devices can be beneficial in system monitoring and control to a maximum extent. It provides a reconfigurable platform that delivers functional

***Corresponding author.**
*E-mail address: ipseeta.nanda@gmail.com

This paper was recommended for publication in revised form by Editor-in-Chief Ahmet Selim Dalkilic

Published by Yıldız Technical University Press, İstanbul, Turkey

and implementation modules for IoT. There are many applications, for example, audio processing, image processing, high-performance computation, and speech recognition. For digital signal processing these days, FPGA [5] platforms are accessed more often. In the execution of these applications in hardware, utilizing coprocessors is a rigorous characteristic feature. The disadvantage of the system arrangement is that here the coprocessors are complex and are surrounded by existing resources in the appliance. The coprocessors that are present on the chip but are not frequently used. The idleness prevents it from giving the ultimate performance [6,7]. This can be overcome by the utilization of the DPR technique as it provides reconfigurable hardware designs for high-performance processing. FPGA platforms are designed for facilitating parallel systems which provide a basis for adaptive, scalable, and simultaneous computing. The architecture is also admissible for real-time system structure in the area of audio and video advancing [8,9]. The design can provide flexibility and deals with minimum latency and maximum performance [10,11]. The power utilization is measured as one of the key challenges for FPGA System on Chip (SoC) [12]. Another disadvantage is that the video processing and audio processing performance is becoming very rigid nowadays with the use of smart devices. The multimedia application is demanding an increase in the number of functionalities in the device. The SoC provides the best application to the electronic industry in the case of hardware [13]. If the module can be reconfigured without dropping a frame and lower the area, then it can boost the competitive market [14-16].

FPGA offers a chance to investigate the difficulties and possibilities that it will provide in real-world applications covered in one of the fascinating areas of research [1,6]. By multiplexing hardware in time, reconfiguration still primarily aims to emulate infinite resources. Performance management, however, is also processed or has its processing unit reduced as necessary. Although electricity wasn't a big issue at the time, researchers found that performance management techniques can also be utilized to control power usage. Utilizing dynamic reconfiguration helps conserve energy. This is done by altering the search regions in accordance with frame-to-frame statistics, which in some situations can reduce computation and hence power consumption. Repurposing unused areas of the reconfigurable device to implement local memory was another method for lowering power-intensive off-chip communications [3,9,15]. The key justification was that the open bitstream format made it possible to reverse engineer the IP-Core, a type of exclusive hardware. After a few months, Xilinx unveiled the Virtex device family, which had features quite identical to those previously mentioned, with the exception of the targeted and safe reconfiguration functionalities. FPGA was developed to guarantee improved design parameters and functional flexibility. Choosing which configurations should stay on the chip and which ones should be replaced when a reconfiguration happens is the difficult part of configuration caching. An improper choice will result in a substantially larger reconfiguration overhead than a correct choice since it will fail to lower the reconfiguration overhead. The limited number of concurrent configurations on the chip and the non-uniform configuration make this choice more difficult. To ensure the best reduction in reconfiguration overhead, configuration aspects such as frequency and latency must be taken into account. In particular, it is sometimes preferable to have setups with high latency rather than those with lower latency that are frequently used. Other times, maintaining high latency configurations while neglecting the frequency factor will force you to transition between other commonly needed configurations since they won't fit in the available space. Switching in this situation results in reconfiguration costs that would not exist if configurations with high latency but low frequency were unloaded. Configuration caching becomes more complex due to the many features of several FPGA programming models, such as the Single Context, Multi-Context, and Partial Run-Time Reconfigurable [5,13] models. Compile Time Reconfiguration (CTR) or Run Time Reconfiguration (RTR) are the two methods through which systems using FPGAs can take advantage of their re-programmability [4]. Never alter the FPGA's settings in a compile-time reconfiguration mechanism while the application is running. RTR systems modify the FPGA configuration while they are operating, either completely or partially [10-12]. Even while the application is running normally, Xilinx Virtex FPGAs offer the option of reconfiguring only a portion of their resources. Avionic, fault-tolerant software, multimedia, and other intelligent systems with restricted resources that need security features currently use this technology. By loading partial bitstreams, usually kept in flash memory, through any open configuration port, Dynamic Partial Reconfiguration (DPR) can modify the FPGA's functionality.

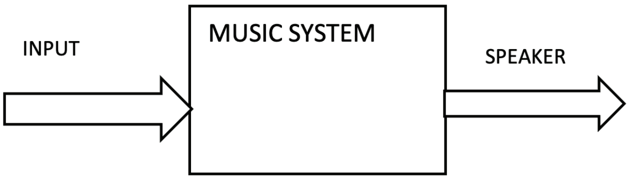


Figure 1. Simple block diagram of ACSA.

ACSA LOW POWER DESIGN

The audio signal processing is taken up as a case study. A complete DPR system is developed in an FPGA [17,18]. The audio codec presented on the FPGA is utilized to digitize the analog audio signal and is fully PC compliant [19-21]. In Figure 2, the ASCA Platform for Real-Time Processing is shown, which consists of processing blocks.

The audio signal of individual channels of Left and Right is then given to the respective filter. Audio signal design involves different filters, and then one filter is chosen based on one at a time [22-24]. The filter design in each channel is mutually exclusive as it is implemented as a dynamic module of the DPR design. In audio signal processing, the benefits of DPR are detected, and it shows a low power saving system having a greater number of PRRs and more RMs associated and is analyzed [25-27].

For logic synthesis design the node representation is implemented using two steps in Sum of Product form (SoP) and Product of Factor form (PoF), Sum of Factor (SoF) [21,23] as shown in eq. 1,2 and 3.

$$x = wy'z + w'yz + acy' \qquad \text{(1) SoP form}$$

$$c[a' + x(e + d)] \qquad \text{(2) PoF form}$$

$$c' + b(e + d) \qquad \text{(3) SoF form}$$

A variable node 'V' is a non-terminal. It is having its argument with index(V) ∈ {1……m}. A constant node 'V' is a terminal node. With value Value(V) ∈ {0,1}. In Node synthesis manipulation optimization algorithm is applied through Boolean network which is very used for floor planning node analysis. It is functional to the optimized network where less amount of logic gates for different purposes. The following are extraction, decomposition, substitution, factoring and collapsing. Incase of decomposition is takes single Boolean function X(f) which is decomposable as shown in eq.4&5.

The function [21][23]

$$x = ab'c' + ab'd + a'cd' + bcd' \qquad (4)$$

It consists of 12 literals. Incase of decomposition x be represented as

$$\begin{aligned} x &= ab'(c' + d) + (a' + b)cd' \\ &= ab'(c' + d) + (ab')'(c' + d)' \\ &= fy + (fy)' \end{aligned} \qquad (5)$$

Where f=ab' and y=c'+d

Hence here the decomposition reduces x the operation of y to 8 literals.

The modeling and implementation of reconfigurable hardware design for real-time video applications using external DPR. In video real-time signal processing, different categories of noises are present, for which filters are required. As explained earlier, to eliminate noise, one filter is essential at a time, for which DPR is used as an analysis. By using DPR, the filter which is essential for the removal of noise is configured. In this case, mean and median for video signals are constructed. The Mean and Median filters are used for reducing the noises found in image files. Noise is acquired in images either during creation, allocation, or also on display instance, which needs to be corrected by using different types of filters. The pixels often found with errors that are found to be distinctly dissimilar from their neighbors. The noise is reduced based on using filters that are only for incoming data. Every time one filter is utilized for a particular noise present in the incoming image file. The hardware virtualization is accessible by DPR [27-30]. The filters that are required are configured in the slot during runtime. It is designed for video signals. The external DPR is utilized to reconfigure the filter. The video filtering method for noise reduction can be fulfilled by the mathematical analysis of the real-time intensity of each pixel using a decoder. The mean filtering is used to replace each pixel value in an image. It has the effect of eliminating pixel values. In the digital processing of an image, it can be in the region of is a way that is generally in use to improve video images that are ruined because of noise. It computes the arithmetic mean of pixel intensity value. For every pixel, the location is captured, images from a scene which is to be used. In the case of median filtering, a single pixel is substituted by its median value of every pixel value confined in the neighborhood, counting itself. It requires the type of categorization in pixel values, region either in progressing or lowering order, and picks up the median estimation of the array. It can also be applied as a dynamic module of DPR design. The module-based method of DPR is particularly used. The system consists of a single PRR and two RMs for using mean and median filters. The blocks are developed by XILINX ISE. The DPR system is built via the Xilinx PlanAhead tool. An expected approach is shown in Figure 1.

WORKING ACSA IN REAL TIME USING XILINX PLANAHEAD

The designed system (MCSA) consists of four filters per channel for an audio system. In this case study design, four digital filters are used: a low-pass filter (LPF), a high-pass filter (HPF), a band-pass filter (BPF), and a band-stop filter (BSF) per channel for low-power design analysis. The system consists of four filters per channel for an audio system. The details of the filter cut-off frequencies are enlisted in Table 1. Individual filters are utilized for the two channels, Left and Right, of the audio input signals. All the required blocks are developed using hardware languages such as VHDL/Verilog and are compiled on XILINX ISE. Here, a DPR system is built using the Early Access PlanAhead tool by Xilinx. The result using PlanAhead is shown in Figure 2, in which all the four filters are designed as one filter slot.

The video signal processing includes the implementation of mean and median filters. The filters are designed dynamically. A significant amount of area utilization is necessarily dynamically configured. The MB DPR, where PR is certainly selected, consists of a single Partial Reconfiguration Region (PRR) and two reconfigurable

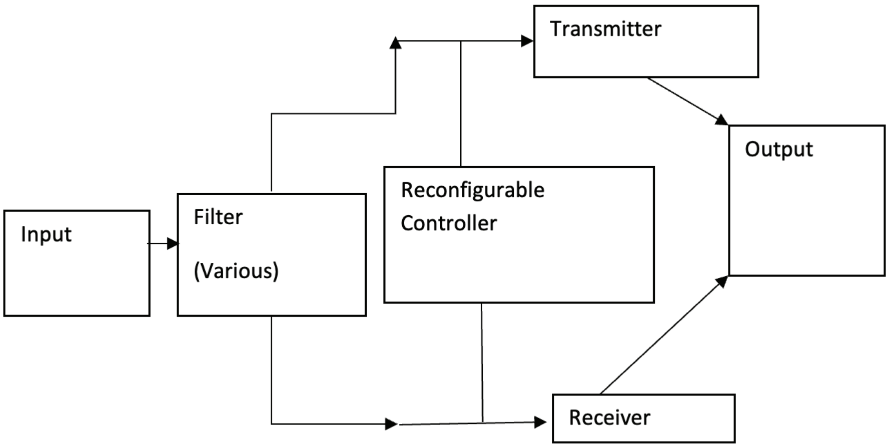


Figure 2. MSCA representation for real time processing.

segment modules. A full DPR system is designed around the ML402, which consists of a Virtex SX35 FPGA. The system is developed, designed, and compiled using languages such as VHDL/Verilog on Xilinx ISE. By using the Xilinx PlanAhead tool, a DPR system is implemented. In this design, a module-based approach is selected, and a large amount of chip area is configured; it consists of two PR Regions (PRR) and four RMs: LPF, HPF, BPF, BSF, associated with each PRR. The case study is analyzed with respect to chip area, resource usage, and reconfiguration time.

A total DPR system is designed with an ML402 board that contains an SX35 FPGA. The audio codec available on the board is used to digitize the analog audio signal. The audio source is the line out of any music-playing system, while the output device is a speaker, as shown in Figure 1. The preliminary configuration and peripheral DPR are carried out through the JTAG interface and also with the help of the Impact tool. Here, ICAP is used under software control.

In this case, the author uses definite regions of the FPGA [30-35] to reconfigure the novel function at runtime, and other areas in the FPGA act as static for a particular time period. The user interacts with Partial Reconfigurable Regions (PRRs) and reconfigurable modules (RMs). By means of HyperTerminal, it controls the operation accordingly. It reconfigures approximately specific areas of the FPGA with a dynamic innovative operation at active runtime [36]. The remaining areas stay static during this operation time. The difficulties in runtime can be summarized by a tool called PLANAHEAD, which was presented by XILINX. It is able to implement throughout runtime reconfigurable systems for all VIRTEX FPGAs. PLANAHEAD is the first graphical environment for partial reconfiguration (PR). It aims to reduce the board space, alter a design in the field, and lower the power consumption. The DPR is recognized as active dynamic partial reconfiguration. It modifies a slice of the device, and the rest of the FPGA is running. It means that partial data is emphasized on the FPGA. The remaining area of the device is still operating (on the fly).

The innovative data is executed and configured on the FPGA. PlanAhead's key determination is to alter the way circuits planned in ISE are laid out on the FPGA [37]. It also provides timing and placement analysis to progress in the case of circuit function. By means of this tool, applications that group the circuits and modules provide the benefit of each and every single module term in its own area. The mission of PR becomes much easier. As the first one area is being programmed and that won't affect any other units, it is identified as floor planning. PlanAhead has a facility of suitable set of design rule checks (DRC) that can be utilized to improve the designs and offer suggestions to the designer [1]. The PR balances the requirement desired by the complex design. Indeed, the DPR makes the hardware very flexible. It also allows the user to change the internal structure of the FPGA dynamically [24,25], without having to turn off. New FPGA chips with PR capability such as Virtex II, Virtex II Pro, Virtex-4, Zynq7000 are existing. These new devices comprise two subsets of resources: system resources and operational resources. The system sources are constituted by a processor and also by internal peripherals.

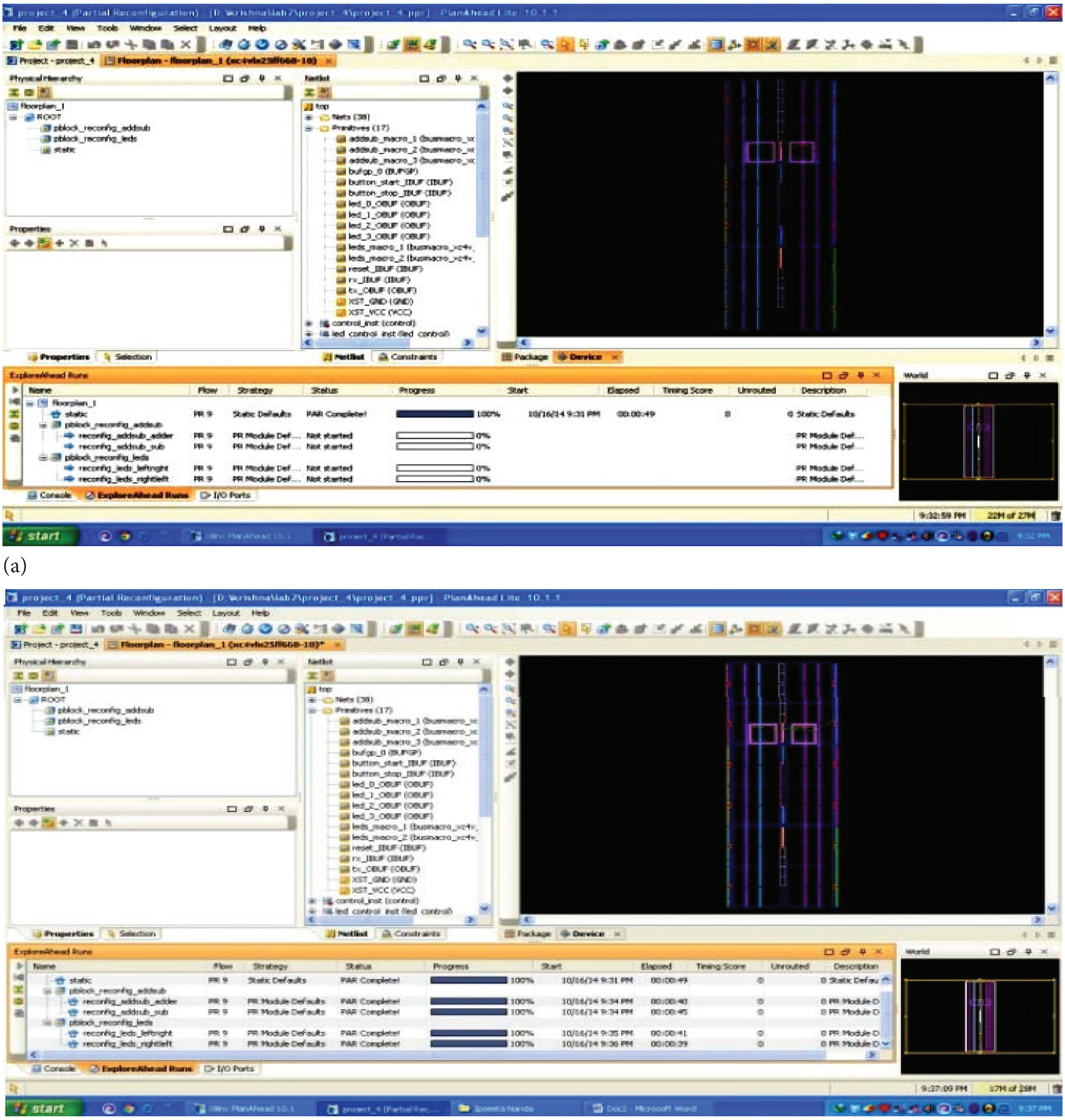
The floor planning retrieves the constraint entry and Design Rule Checks (DRCs) [30] with the help of Xilinx PlanAhead [16]. The place-route tools are used to produce the necessary bitstreams. The full and partial design images, and each configuration characterizes a whole FPGA design. The tool used is Xilinx PlanAhead, which is capable of reducing the complexity at runtime. The reconfigurable system is built on the Virtex field programmable array. PR is dynamically configured and is modified by downloading partial and full bitstreams. The module is pre-synthesized, and the netlist file is stored in the synth directory. Bus macro plays an important role in the floorplan due to which communication is possible. The ucf, bus macro, and additional net files are carried by the data directory. The region partition of the FPGA is added with a newly function added dynamically on the fly, and the remaining files remain static.

Table 1. Resource Utilization in audio processing without DPR

Resource	# Available Resources	# Used	% Utilized
LUT	30720	3214	10
Flip flops	30720	1963	8
Slice	15360	2575	14
DSP blocks	192	85	42

Table 2. Resource utilization in audio processing with external DPR

Resource	# Available	# Used	% Usage
LUT	30720	882	3
Flip flops	30720	627	2
Slice	15360	541	4
DSP blocks	192	20	18



(a)

ExploreAhead Runs

Name	Flow	Strategy	Status	Progress	Start	Elapsed	Timing Score	Unrouted	Description
floorplan_1									
static	PR 9	Static Defaults	PAR Complete!	100%	10/16/14 9:31 PM	00:00:49		0	0 Static Defaults
pblock_reconfig_addsub									
reconfig_addsub_adder	PR 9	PR Module Def...	Not started	0%					PR Module Def...
reconfig_addsub_sub	PR 9	PR Module Def...	Not started	0%					PR Module Def...
pblock_reconfig_leds									
reconfig_leds_leftright	PR 9	PR Module Def...	Not started	0%					PR Module Def...
reconfig_leds_rightleft	PR 9	PR Module Def...	Not started	0%					PR Module Def...

(b)

ExploreAhead Runs

Name	Flow	Strategy	Status	Progress	Start	Elapsed	Timing Score	Unrouted	Description
static	PR 9	Static Defaults	PAR Complete!	100%	10/16/14 9:31 PM	00:00:49		0	0 Static Defau
pblock_reconfig_addsub									
reconfig_addsub_adder	PR 9	PR Module Defaults	PAR Complete!	100%	10/16/14 9:34 PM	00:00:40		0	0 PR Module D
reconfig_addsub_sub	PR 9	PR Module Defaults	PAR Complete!	100%	10/16/14 9:34 PM	00:00:45		0	0 PR Module D
pblock_reconfig_leds									
reconfig_leds_leftright	PR 9	PR Module Defaults	PAR Complete!	100%	10/16/14 9:35 PM	00:00:41		0	0 PR Module D
reconfig_leds_rightleft	PR 9	PR Module Defaults	PAR Complete!	100%	10/16/14 9:36 PM	00:00:39		0	0 PR Module D

Figure 3. (a) (b) Real Time result of reconfigurable processor filter design using Xilinx PlanAhead.

Table 3. Comparison table with state of art

Aspect	Current research	State of art
Technique	Dynamic partial reconfiguration (DPR)	Traditional FPGA configurations
Application	Multimedia communication system architecture (MCSA)	General Purpose FPGA applications
Power efficiency	Demonstrates significant power savings	Limited focus on power optimization
Area utilization	Optimized chip area using DPR	Traditional FPGA designs
Tools used	Xilinx planahead, virtex FPGA	Standard FPGA designs
Use cases	Real-Time audio processing, video signal processing	General FPGA applications
Flexibility	Adaptable, reconfigurable for various scenarios	Limited adaptability
Analysis	Detailed resource utilization tables	Generalized FPGA application descriptions

DISCUSSION AND AREA ANALYSIS OF ACSA WITH DPR AND WITHOUT DPR FOR LOW POWER DESIGN

Area Analysis of resource utilization for real-time audio processing without DPR targeting a Virtex-4, and resource utilization for real-time audio processing with DPR, is shown in Tables 1 and 2, respectively. As a replacement for four filters, one filter slot in DPR implementation saves three times. From the table, the number of input LUTs is 3214, the number of Flip Flops (FF) is 1963, the number of Slices is 2575, and the number of DSP blocks is 85 in real-time audio processing without DPR. In the case of real-time audio processing with DPR, resource utilization is calculated as the number of LUTs being 882, the number of FFs being 627, the number of Slices being 541, and the number of DSP blocks being 20.

This research introduces a real-time multimedia communication system architecture for IoT using dynamic partial reconfiguration, optimizing power and chip area. The comparison of state of art with current research is discussed in table 3 with various aspects. If a comparison is made between two designs, then it is found that the resource utilization of real-time audio processing with external DPR targeting a Virtex-4 FPGA is more advantageous than without DPR. The extra number of LUTs used is (3214-882) = 2332, the extra number of slices is (2575-541) = 2034, the extra number of FFs used is (1963-627) = 1336, and the extra number of DSP used is (85-20) = 65. The comparison is shown in Figure 4. These days, the size of electronic devices is getting minimized, because of which chip area is an important parameter. The chip area is related to power consumption, as the power consumption is reduced if the chip area is less. So, according to the table provided, the design with external DPR provides less power consumption.

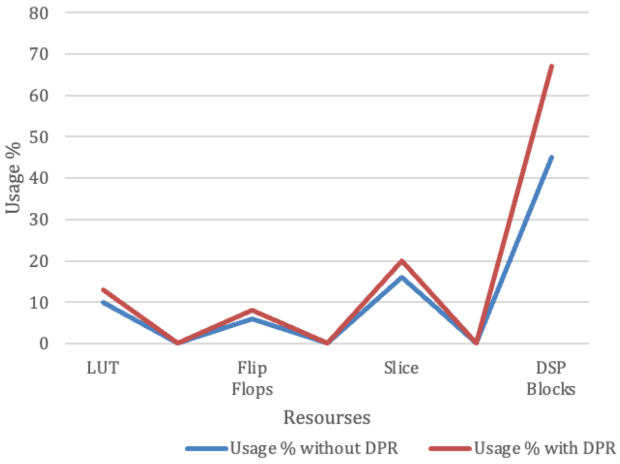


Figure 4. Resource utilization in audio processing.

CONCLUSION

The work in this paper is carried out with the aim of application development, focusing on using different types of DPR in various applications and scenarios. Here, the area is analyzed, and it is observed that implementation based on module-based DPR shows benefits in resource utilization. Also, for a system having a higher number of PRs and more RMs associated with it, experimental results indicate that the system enables more savings in resources by effectively exploiting hardware virtualization offered by DPR. Out of five applications, four applications are implemented on Virtex-4. The Virtex-4 supports both Internal and External DPR. The sizes of static and dynamic bitstreams differ in both filter designs. The reconfiguration is handled by an external PC, by downloading partial bitstreams through JTAG. Two applications are implemented using Internal DPR, where reconfiguration is handled internally using ICAP. Future research can explore advanced optimization strategies for Dynamic Partial Reconfiguration (DPR), assess DPR's suitability in real-time applications, investigate security considerations, evaluate newer FPGA architectures, and delve into dynamic resource allocation. These areas promise to enhance DPR efficiency, security, and applicability across evolving technologies.

AUTHORSHIP CONTRIBUTIONS

Authors equally contributed to this work.

DATA AVAILABILITY STATEMENT

The authors confirm that the data that supports the findings of this study are available within the article. Raw data that support the finding of this study are available from the corresponding author, upon reasonable request.

CONFLICT OF INTEREST

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

ETHICS

There are no ethical issues with the publication of this manuscript.

REFERENCES

[1] Nanda I, Adhikari N. Application and performance of FPGA using partial reconfiguration with Xilinx PlanAhead. In 2017 IEEE Transportation Electrification Conference (ITEC-India) (pp. 1-4). IEEE, 2017. [CrossRef]

[2] Siva Prasad B, Mallikarjuna Rao P, Madhav BTP. CPW fed T-stub and U-slot reconfigurable antenna for Wi-Fi and WLAN communication applications. J Adv Res Dyn Control Syst 2017;9(Suppl 14):2104–2116.

[3] Nanda I, Ali SM. Implementation and execution of math partial reconfiguration region and LED dealing with Xilinx PlanAhead. Int J Adv Electron Electr Eng 2015;4:1–3. [CrossRef]

[4] Balaji S, Khan H, Janga Reddy M, Gurunadha Babu M. Authentication frameworks for enhancing security in biometric systems. Int J Mech Eng Technol 2017;8:1073–1080.

[5] Kao C. Benefits of partial reconfiguration. Xcell J 2005;55:65–67.

[6] Murali Krishna B, Madhumati GL, Khan H. Dynamically evolvable hardware-software co-design based crypto system through partial reconfiguration. J Theoretic Appl Inform Technol 2017;95:2159–2169.

[7] Nanda I, Pujari S, Panda CS. Implementation of math PRR and LED processing using Xilinx PlanAhead. In 2015 International Conference on Computing Communication Control and Automation (pp. 955-958). IEEE, 2015. [CrossRef]

[8] Peesapati R, Anumandla KK, Sabat SL. Performance evaluation of floating point Differential Evolution hardware accelerator on FPGA. IEEE Region 10 Annual International Conference, Proceedings/TENCON, pp. (3173–3178), 2017. [CrossRef]

[9] Hübner M, Becker J. Exploiting dynamic and partial reconfiguration for FPGAs: toolflow, architecture and system integration. In Proceedings of the 19th annual symposium on Integrated circuits and systems design (pp. 1–4), 2016. [CrossRef]

[10] Saleem Akram P, Madhav BTP, Jeevana Sravya G, Sudhakar V, Lakshmi Sirisha G, Mounika C, et al. Design and analysis of square shaped serrated patch antenna for ultra-wideband applications with single rejection band. Int J Eng Technol 2018;7:525–529. [CrossRef]

[11] Bhandari SU, Subbaraman S, Pujari S, Mahajan R. Internal dynamic partial reconfiguration for real time signal processing on FPGA. Indian J Sci Technol 2010;3:365–368. [CrossRef]

[12] Madhav BTP, Sai Dheeraj G, Raghavarapu SS. Design of a CPW-fed monopole antenna for ultrawide band based iot and medical applications. Int J Pharm Res 2018;10:74–79. [CrossRef]

[13] Lopez-Buedo S, Garrido J, Boemo EI. Dynamically inserting, operating, and eliminating thermal sensors of FPGA-based systems. IEEE Trans Compon Packag Technol 2010;25:561–566. [CrossRef]

[14] Madhav BTP, Venkateswara Rao T, Tirunagari A. Design of 4-element printed array antenna for ultra-wideband applications. Int J Microw Opt Technol 2018;13:8–17.

[15] Eto E. (2007). Difference-based partial reconfiguration. XAPP290 (v2. 0) December, 3.

[16] Srikanth Reddy P, Satyanarayana P, Sai Krishna G, Divya K. Hardware implementation of variable digital filter using constant coefficient multiplier for SDR applications. Adv Intell Syst Comput 2018;668:495–502. [CrossRef]

[17] Two Flows for Partial Reconfiguration: Module Based or Difference Based Xilinx,, XAPP290 (v1.2) September 9, 2004.

[18] Partial Reconfiguration Design with Planahead 9.2 by Brian Jackson Ver1.1, August 2007.

[19] Cheerla S, Venkata Ratnam D, Teja Sri KS, Sahithi PS, Sowdamini G. Neural network based indoor localization using Wi-Fi received signal strength. J Adv Res Dynam Control Syst 2018;10:374–379.

[20] Media Presentation. Using the PlanAhead 9.2 for Partial reconfiguration flow Xilinx, Brain Jackson.

[21] Timing S, Device X. Xilinx ISE 8 software manuals and help-PDF collection, 2015.

[22] Nanda I, Adhikari N. Application and performance of FPGA using partial reconfiguration with Xilinx PlanAhead. In 2017 IEEE Transportation Electrification Conference (ITEC-India) (pp. 1-4). IEEE, 2017. [CrossRef]

[23] XPS HWICAP (v1.00.a), DS586, Xilinx, November 05, 2007.

[24] Nanda I, Adhikari N. Study and design of reconfigurable processor peripherals using Xilinx tools. J Electron Des Technol 2018;9:17–23.

[25] Biradar V, Reddy BM, Anumandla KK. DM3730 Processor Hardware Debugging on Linux Platform. 2018 4th International Conference for Convergence in Technology, I2CT 2018. [CrossRef]

[26] Kishore KH, Prasad BKV, Teja YMS, Akhila D, Sai KN, Kumar PS. Design and comparative analysis of inexact speculative adder and multiplier. Int J Eng Technol 2018;7(Suppl 8):413–418. [CrossRef]

[27] Zalke JB, Pandey SK. Dynamic partial reconfigurable embedded system to achieve hardware flexibility using 8051 based RTOS on Xilinx FPGA. In 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies (pp. 684-686). IEEE, 2009. [CrossRef]

[28] Chaitanya NK, Varadarajan S. Load distribution using multipath-routing in wired packet networks: A comparative study. Perspect Sci 2016;8:234–236. [CrossRef]

[29] Fernando X, Lăzăroiu G. Spectrum sensing, clustering algorithms, and energy-harvesting technology for cognitive-radio-based internet-of-things networks. Sensors 2023;23:779. [CrossRef]

[30] Kulakli A, Arikan CL. Research trends of the ınternet of things in relation to business model innovation: results from co-word and content analyses. Future Internet 2023;15:81. [CrossRef]

[31] Wang Y, Su Z, Guo S, Dai M, Luan TH, Liu Y. A survey on digital twins: architecture, enabling technologies, security and privacy, and future prospects. IEEE Access 2023;10:1–27. [CrossRef]

[32] Azzaz MS, Kaibou R, Madani B, Arzazi M, Dahmane K. FPGA Implementation of Real-Time Video Watermarking Prototype Using PL-PS Embedded Processor Over Network Communication. In 2023 International Conference on Networking and Advanced Systems (ICNAS) (pp. 1-6). IEEE, 2023. [CrossRef]

[33] Faizan M, Intzes I, Cretu I, Meng H. Implementation of deep learning models on an SoC-FPGA device for real-time music genre classification. Technologies 2023;11:91. [CrossRef]

[34] Mani V, Ghonge MM, Chaitanya NK, Pal O, Sharma M, Mohan S, Ahmadian A. A new blockchain and fog computing model for blood pressure medical sensor data storage. Comput Electric Eng 2022;102:108202. [CrossRef]

[35] Popoff M, Michon R, Risset T, Cochard P, Letz S, Orlarey Y, et al. Audio DSP to FPGA Compilation: The Syfala Toolchain Approach (Doctoral dissertation). Lyon: Univ Lyon, INSA Lyon, Inria, CITI, Grame, Emeraude; 2023. [CrossRef]

[36] Al-Dulaimi MAA, Wahhab HA, Amer AA. Design and implementation of communication Digital FIR filter for audio signals on the FPGA platform. J Commun 2023;18:89–96. [CrossRef]

[37] Ding B, Huang J, Wang J, Xu Q, Chen S, Kang Y. Task modules partitioning, scheduling and floorplanning for partially dynamically reconfigurable systems with heterogeneous resources. ACM Trans Design Autom Electron Syst 2023;28:1–26. [CrossRef]