


An Area Efficient and Low Power Consumption of Run Time Digital System Based on Dynamic Partial Reconfiguration

R. Saravana Ram¹  · A. Gopi Saminathan² ·
S. Arun Prakash³

Received: 13 March 2018 / Accepted: 19 May 2018 / Published online: 30 May 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract Digital signal processing besides multimedia applications needs plenty of data, real-time processing capacity, and high computational power. Thus, adaptable architectures with run-time reconfiguration abilities have gotten expanded consideration. Basically, Reconfiguration computing is going towards advancing the application adaptability at runtime. A reconfigurable structure can be attained by working up the strategy aimed at configuring an array of programmable logic reprogramming. Field Programmable Gate Arrays (FPGAs) is made with the intention of reconfiguring the array system with interconnects as well as the configuration of logic blocks. To implement a high-performance FPGA device and also to enhance, the given paper proposes a proficient design strategy. The proposed strategy count upon the employment of dynamic partial reconfiguration (DPR) to drive from one mode then onto the next utilizing time-multiplexing on the same chip region. Furthermore, reconfigure modules to spare considerable area and enable the low-cost FPGAs usage. In the given work, reconfigurations of the modules accompanied by the memory are finished. The DPR is implemented betwixt these modes to shift from one mode then onto the next. The pro-

✉ R. Saravana Ram
saravanaramkrishnan@gmail.com

A. Gopi Saminathan
agsaminathan@gmail.com

S. Arun Prakash
arunprakashmadurai@gmail.com

- ¹ Department of Electronics and Communication Engineering, Anna University, University College of Engineering Dindigul, Dindigul, Tamilnadu, India
- ² Department of Electronics and Communication Engineering, NPR College of Engineering and Technology, Natham, Dindigul (dist), Tamilnadu, India
- ³ Department of Electrical and Electronics Engineering, Anna University, University College of Engineering Ramanathapuram, Ramanathapuram, Tamilnadu, India

posed method helps in diverse applications with various demands and also attains high performance, power consumption together with throughput. The proposed work gives improved performance with fewer powers and less area utilization which is illustrated by the Experimental outcomes.

Keywords Digital signal processing · Reconfiguration computing · Field programmable gate array · Dynamic partial reconfiguration

1 Introduction

A critical challenge intended for nanoelectronic frameworks is to attain yield together with reliability. As very-large-scale integration (VLSI) technology scales into the nanometer scale, gadgets and also interconnects are liable to progressively common defects and noteworthy parametric varieties [1]. Likewise, in applications which need high processing speeds and also low power consumption, the general purpose processors' sequential behavior gives limitations. A possibility for this issue is the FPGAs usage, which has the gain of the parallel procedure capacity to explore the intrinsic parallelism of handful algorithms [2]. Numerous heterogeneous platforms formed of digital signal processors, application specific integrated circuits (ASIC), and FPGAs have been formed to accomplish adequate processing power along with flexibility. To modify to flexibility and also computational issues, the FPGA DPR's concept can well be engaged [3]. To advance the even more extensive adoption of FPGA gadgets, powers saving systems in FPGA devices have been broadly explored [5].

With the intention of providing an extensive number of computational resources to process numerous parallel streams together with keeping up the power utilization at an acceptable level, reconfigurable architectures are assuming a noteworthy part in embedded frameworks as they give high computation parallelism and throughput [4]. Reconfigurable computing devices encompass immense potential in using application-particular parallelism to do the high-speed computation. As such, FPGA devices are generally utilized in several application domains that require both computing performance and implementation flexibility, for example, digital signal image, besides real-time multimedia processing [5]. Inside such frameworks, heterogeneous computing resources incorporate processors, which may differ inside and across devices and reconfigurable computing resources, say, FPGAs, hardware accelerators, et cetera [6]. Most existing works address on surmised computing which exchanges between computation accuracy and power by dynamic reconfiguring logic centered upon required accuracy [7].

FPGAs are electronic devices without initial functionality or computational model. The procedure expected to expedite applications employing reconfigurable hardware has required so far some mastery in hardware design with the intention of getting maximum advantages as of the hardware platform where the design is being implemented [8]. Contemporary FPGAs show ample of transistors and this drives the chip's static power Consumption to unacceptable levels. Furthermore, the post-fabrication flexibility gave by these devices are executed utilizing countless pre-fabricated routing tracks accompanied by programmable switches, which expend heaps of power every

time they switch [9]. In addition, with the diminishing voltage is utilized to lessen power Consumption, with the expanding cell density, and expanding utilization of VLSI devices the danger from delicate blunders on ground level is always rising [10].

FPGAs must be worked with more productive and substantial interconnect topologies and rationale components to help intricate and larger circuits. The FPGA design enormous challenge is to locate a decent exchange amongst flexibility and also performance concerning power Consumption. Truth be told, these devices incorporate a substantial number of pre-fabricated routing tracks and programmable switches permitting to course different placement solutions [11]. In FPGAs, the values differ as per the design and working conditions, except the tradition which just reliant on the objective device and operating frequency. The Power Consumption of any digital framework relies upon the accompanying central point: logic partition, route and placement, mapping and also resource usage [12]. In FPGA, interconnect necessary to course the signals betwixt these blocks can as well be limited. As the densities of the FPGAs being made keeps on expanding there are raising needs to lessen FPGA compilation times, without diminishing the quality level of the solutions being delivered [13].

The draft structure of the given paper is systemized as takes after. Section 2 surveys the associated work regarding the proposed work. In Sect. 3, exhibited a short depiction of the proposed approach, Sect. 4 analyses the investigational outcome and Sect. 5 the paper is concluded.

2 Literature Survey

Palumbo et al. [14] posited a structure in which data flow specifications was embraced as a beginning position for characteristic power minimization in coarse-grained reconfigurable embedded systems. The suggested stream was formed of two resulting phase: (1) the categorization of the ideal topological framework specification(s) (2) the finding of disjointed logic regions. The last was then employed to execute clock along with power gating methods. The suggested approach expanded a dataflow-centered design framework aimed at coarse-grained reconfigurable designs, the Multi-Dataflow Composer tool, designed inside MPEG-RVC research studies. The significance of the suggested MDC tool's power-aware extension was for the most part identified with the way that, inside the MPEG-RVC structure, power utilization was as yet an open problem and also power assessment or else management approaches were right now under inspection. The solution joined structural along with dynamic procedures implemented systematically at the modeling level. Focusing on an ASIC 90 nm technology along with a 45 nm one, Results were evaluated. The model-centered approach's validity had been demonstrated above the reconfigurable calculating center of a multifunctional coprocessor Aimed at image processing applications. Still, the optimization process is time-consuming and power consumption stands high.

Tu et al. [15] planned a Deep Convolutional Neural Networks (DCNN) acceleration architecture, with reconfigurable calculation patterns intended for various models, termed Deep Neural Architecture (DNA). The model included a data reuse pattern together with a convolution mapping technique. Aimed at huge with diverse layer sizes, to aid a hybrid data reuse pattern, DNA set up again it's data path that decreased

aggregate energy utilization by 5.9–8.4 times above traditional method. Considering different convolution parameters, DNA set up again the calculating resources to help a profoundly scalable convolution mapping strategy, which acquired computing resource employment of 93% on contemporary DCNNs. At last, a layer-centered scheduling system was presented to adjust DNA's power effectiveness and also the performance for various DCNN. DNA was actualized in the area of 16 mm² at 65 nm. It accomplished 194.4 GOPS at 200 MHz and expended just 479 mW on the standard. The framework level power productivity was 152.9 GOPS/W, which beat the best in class designs by means of 1–2 orders. Though it achieved high performance and also power efficiency, the design process was quite complex.

Yu et al. [16] displayed a multicenter embedded processor by means of reconfigurable same instruction multiple process (RSIMP) architecture intended to diminish the power expenditure of instruction memory (IM) therefore lessening the aggregate processor power. It was implemented to the situation that few centers implemented exceptionally matching instruction sequence simultaneously. Then they have reconfigured to a master–slave mode which merely the master center brought guidelines, and appropriated the orders to the slave center(s), whilst the slave centers' IM were closed down to spare power. NoC was utilized to achieve the mode reconfiguration. Slow down generated by mutual resources dispute and also accelerator call in slaves was joined to master's pipeline control. To adjudicate mutual memory access, State machines were composed. Outcomes of the Experiment demonstrated 13.8% and also 21.9% of the framework power diminishment with insignificant hardware overhead besides performance loss in 2Core and also 4Core configurations, individually, for the standards chose as of LTE MIMO framework, RS decoder, H.264/AVC codec along with AES encryption/decryption, contrasted with that of the baseline platform.

Liu et al. [17] introduced a system level simulator, termed subsequent to SimRPU, aimed at the RPU (reconfigurable processing unit), that was the main computing engine. The suggested simulator comprises of software compiler, simulation kernel, framework profiler giving performance, area beside power data aimed at the preferred architectures, besides a framework debugger supporting modification together with scrutinizing of the RPUs' internal state. Object-oriented hierarchical accompanied by parameterized architecture modeling strategies were presented to serve the necessities for a quick and extensive assessment. To augment the profiled performance data's accuracy, Cycle-precise simulation mechanisms were produced. Contrasted with the normal register transfer level centered simulation scheme, the suggested simulator attained an average speed of 18.5× with just 3.5% diminishment on performance evaluation accuracy. A reconfigurable processor focused on high-definition multimedia decoding applications was applied with Taiwan Semiconductor Manufacturing Company 65-nm process utilizing the recommended exploration and also design flow. Calculated outcomes established that the applied architecture had clear favorable circumstances regarding performance and also power utilization than the reference plans in multimedia decoding applications. Still, SimRPU needs longer simulation time than traditional schemes.

Amin et al. [18] posited a method to handle managing resources in an extensive heterogeneous wireless network which upheld reconfigurable devices. The framework exemplified internetworking ideas requiring autonomous wireless networks to coordi-

nate so as to give a united network to clients. They posited a multi-attribute scheduling algorithm executed via a central global resource controller (GRC) which handled the resources of handful distinctive self-sufficient wireless systems. The traits deliberated by means of the multi-attribute optimization function comprise of framework spectral efficiency, battery lifespan of every client (otherwise general energy utilization), and immediate and also long-term fairness intended for every client in the framework. They utilized the analytical hierarchy process (AHP) to figure the relative import of every characteristic, which took interview responses as of wireless network suppliers as input, furthermore, created weight assignments aimed at every feature in the optimization issue. Through Matlab/CPLEX centered simulations, they demonstrated an expansion in a multi-attribute framework usefulness gauge of till 57% aimed at the algorithm contrasted with other broadly examined resource allotment algorithms comprises Max-Sum Rate, Proportional Fair, Max–Min Fair and also Min power.

Purohit et al. [19] displayed the design along with the assessment of two processing components for reconfigurable processing. They additionally displayed a circuit-level execution of the data paths in static along with dynamic design styles to scrutinizing the different performance-power tradeoffs included. At the point when actualized in IBM 90-nm CMOS process, the 8-b data paths accomplished working frequencies extending above 1 GHz aimed at static along with dynamic usage, with every data path behind single-cycle computational capability. Additionally introduced a single-precision floating point processing element (FPPE) utilizing a 24-b variation of the suggested data paths. It worked at 1 GHz frequency with 6.5-mW normal power Consumption was established by the complete usage of the dynamic FPPE. Correlation with contending architectures demonstrated that the FPPE gave two orders of magnitude elevated throughput. Besides, to assess its possibility as a soft-processing solution, they likewise map the floating point unit on the Virtex 4 beside 5 devices furthermore watched that the unit needs below one percent of the aggregate rationale slices whilst using just about four percent of the DSP blocks accessible. At the point when looked at against famous FPGA-centered floating point units, the plan on Virtex 5 demonstrated considerably lower resource use, whilst attaining comparable peak operating frequency. Several of the parameters utilized in this work are centered upon assumptions. It is one of the disadvantage of this work.

Prabhu et al. [20] presented an FPGA based scalable fixed point QRD architecture centered on given Rotation algorithm. This suggested QRD core utilized an effectual pipelined and unfolded 2D MAC based systolic array architecture with DPR (dynamic partial reconfiguration) ability. While functioning at a frequency 49% larger than the prevailing boundary cell architectures, an improved LUT centered Newton–Raphson method was suggested for determining square root and also inverse square root which assisted in decreasing the area by seventy-one percentage and also latency by fifty percent. The scalability of the QRD core was accomplished using DPR which results in the decline in dynamic power and area utilization as contrasted with a static implementation. The evaluation outcomes showed a decline in power, area, and latency as contrasted with CORDIC based architectures. The proposed scalable QRD core was utilized for executing a high-performance adaptive equalizer (QRD-RLS Algorithm) utilized in mobile receiver's and the assessment was conducted by transferring BPSK symbols in its training mode.

Vipin et al. [21] presented an fully automated framework (CoPR) for implementing Partial Reconfiguration centered adaptive hardware systems on Zynq family of hybrid FPGAs. The designer specified a group of valid configurations encompassing hardware modules. CoPR automated partitioning of modules into regions, floor planning on the FPGA fabric, as well as the generation of partial bitstreams. The runtime framework offered an abstracted view of system configuration through an API that allowed the designer to focus on adaptation software without considering details of the underlying hardware. They have shown an example cognitive radio application to demonstrate the effectiveness of CoPR.

3 Proposed Methodology

Lately, by utilizing a reconfigurable platform with custom-made hardware or software, a few applications are created. In this elevated area, by reutilizing the configuration, the redesigning process can well be evaded. FPGA stands one amongst the essential system utilized in reconfiguration to enhances performance and to diminish the variables that to encompass a superior accuracy. Here, proposed a DPR of a configurable multi-mode memory controller (MMMC) and modules to fulfill various applications with various memory necessities and to enhance framework bus usage. Figure 1 depicts the proposed DPR based multimode memory controller. The proposed DPR of a configurable multi-mode memory controller (MMMC), also called Internal Configuration Access Port (ICAP) and modules have been shown in Fig. 1. The proposed architecture counts upon DPR. With DPR, it is possible to execute diverse circuits that are not essential meanwhile, and that doesn't work simultaneously, on the same FPGA zone.

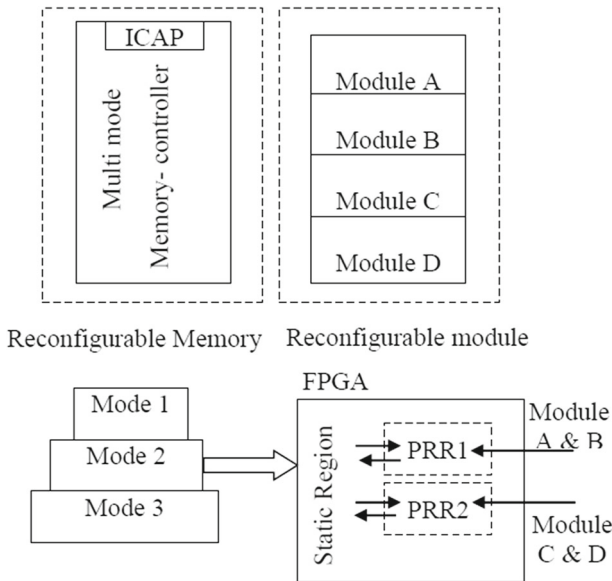


Fig. 1 Proposed DPR

This area is by and large called the reconfigurable region (RR). At whatever point the designer needs to change the applied circuit, a sum of time is anticipated to modify the configuration memory at run-time moreover this is identified as the reconfiguration time. The subsystem that plays out the reconfiguration is termed as the reconfiguration supervisor furthermore is primarily executed in software.

3.1 General Structure of Partial Reconfiguration

DPR is a component of modern FPGAs that permits the adjustment of a working FPGA. Partial bitstreams can well be stacked into the FPGA to reconfigure those areas, without bargaining the functionality of different elements of the device. A framework that utilizes DPR can be thoughtfully separated into a static element as well as a dynamic element. The static part is configured just the once at boot-time with a complete bitstream. The dynamic section can well be reconfigured numerous times amid run-time with various partial bitstreams and it might comprise of some independent reconfigurable areas. At the point when a bitstream stands stacked into the FPGA, it is put away in a static RAM called configuration memory. The element of this memory characterizes the hardware configuration actualized in the FPGA. In this manner, by composing a partial bitstream into the reconfiguration memory, the hardware framework applied in the FPGA stands powerfully modified.

Figure 2 demonstrates the partial reconfiguration's design in the proposed design. DPR is done to guarantee the affinity betwixt different processors and also memory banks. The data distribution happens amongst different processors is retrieved with memory besides latency accessibility. High bandwidth interconnection is acquired in the local communication accompanied by low latency. The fit for implementing it in various processors in association with the effectiveness of data exchange betwixt threads is made sure by this. Inside a transducer, via interconnections in the framework, the moving of data is mapped betwixt the threads.

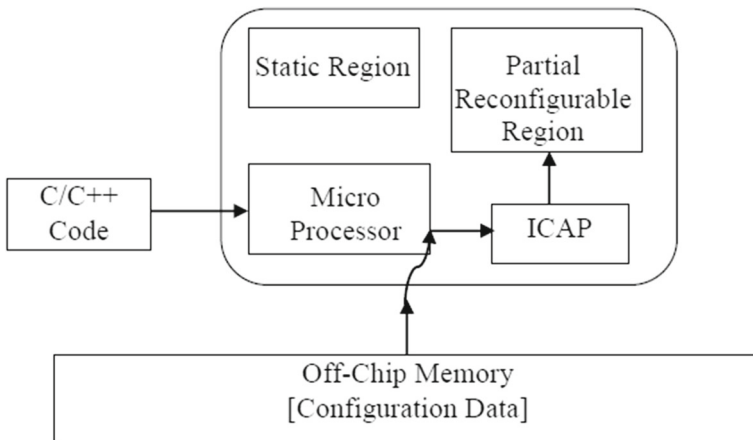


Fig. 2 Design of partial reconfiguration

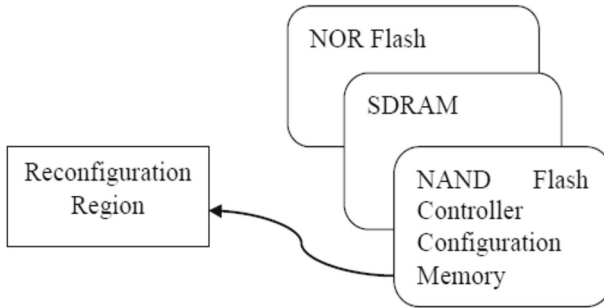


Fig. 3 Multi mode memory controller

3.2 Multi-Mode Memory Controller

Applications require an expansive, quick memory. Be that as it may, in the today's VLSI fabrication technology, as the memory span builds so does its access time. Actually, the main monetarily realistic approach to giving a hallucination of substantial, quick memory is by sorting out it as a hierarchy of locations: little, quick memories nearer to the processors besides bigger, however, slower memories more distant from processors. Figure 3 reveals the proposed MMMC centered on DPR.

The principle idea of the proposed DPR-based multimode memory controller is to present three methods of operations: SDRAM (Synchronous Dynamic Random Access Memory) mode, NOR (Not OR) flash mode, NAND (Not AND) flash mode that doesn't function concurrently. The principle advantage is diminishing region and aggregate count of pins as different protocols can share pins that have same functions, for instance, data bus, address bus, et cetera. In this way, it will be financially savvy and the area overhead from single-mode is acceptable. The DPR strategy permits reconfiguring a piece of the FPGA zone with various functionalities at runtime. DPR permits the improvement of FPGA region productivity and the lessening of power Consumption. DPR is implemented between these memory controller protocols to shift from protocol to another.

3.2.1 Framework of DPR Design Flow

The RR configuration memory comprises of SRAM memory cells which manage the elements of the look-up tables together with the state of the routing switches. To apply a circuit in the RR, a configuration should be produced that encloses the binary values that should be composed in the RR's memory cells. Figure 4 exhibits the DPR design flow's structure.

DPR design flow strategy structure involves a collection of steps, which are important to execute the proposed multi-mode memory controller's applications utilizing DPR as demonstrated in the below steps.

- (a) During the early stage, in Hardware Description Language (HDL), the static modules accompanied by the partial reconfiguration modules (PRM) are depicted.
- (b) The PRMs stands synthesized to produce the relating netlist for every module.

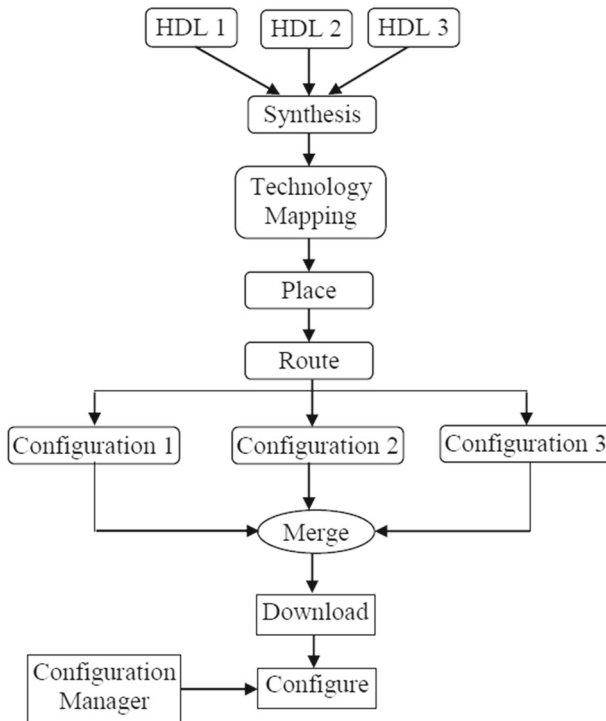


Fig. 4 Framework of DPR design flow

- (c) Carry out placement, routing, and generation of the full and partial reconfiguration bitstream.
- (d) Merges the full bitstream to create a last downloadable bitstream.
- (e) The last downloadable bitstream is duplicated onto the compact flash card and also the card is attached to the FPGA on the way to raise the design on the subsequent power cycle.
- (f) To switch between the distinctive circuits, the reconfiguration director composes the RR with the proper bitstream configuration.

3.3 Reconfigurable Modules

Aimed at boosting the FPGA device's performance, Improved IDPR is proposed. Aimed at fast logic functions, Reconfiguration is done to program the logical circuit; in addition, even during runtime, it does the execution. At once the structure execution of the function permits a solitary full-chip configuration to load. With slow proficiency, the reconfiguration procedure of the chip occurs at runtime. In transducer, performed the function with time base timers [Fixed Interval Timer (FIT) together with Decrementer (DEC)].

3.3.1 In Timer Status Register (TSR)

The status bit will be set, moreover at the time the register value becomes zero, the actions are activated, and after that, the process of decrementing the register will be ended. Additionally, through programming, the Auto-Reload register, can reload the values. As of the time base bit selection, the periodic time generating is obtained in FIT. The data as of the Embedded Parallel Filters have processed the transducer function.

To access data from channels, the technique of pipeline or anchoring process is conducted. Through the predatory pipeline, the different phase of processing is disdain to every stage of the transducer. Centered upon the extra registers embedded in it, every single stage is isolated. The contentions of utilizing chained processing or pipeline are proposed in this design. Concerning data in addition to control are addressed by clenching appropriate circuitry structures. On-chip memory is utilized for productive partition. For holding noteworthy tasks, local memory is utilized for the capacity. Additionally, it identifies the register files which neglect to fill the files limit and the files near processors. At the point when strings executing the tasks are arranged to the tasks register which is fortified in the transducer.

Static reconfiguration stands as a compile-time reconfiguration while DPR is a runtime course. A segment of an FPGA is recomposed by their potentiality. Reconfiguration of rationale framework includes the parameters of hardware employed; furthermore, they are executed to accomplish proficiency process in real time environment deprived of data loss. Statically reconfigurable is possible in numerous prevailing reconfiguration frameworks. Whilst configuring the framework, it is necessary to resume new function; however, in proposed framework reconfiguration is done by reutilizing the design with new function besides making a new design. In on-chip memory, the memory dwelling the data transfer betwixt the processors as well as in the center and past besides the local storage is additionally accessible. In local memory, the storage bounds are implanted to plan systems to guarantee the communication concurrency in addition to computation by pre-fetching and double buffering. Concerning memory access, any remote latency can well be effectively depressed.

3.3.2 Adaptation Mechanism

In this means, by utilizing the parameter tuning, the lookup table's functions are adjusted. To give effective process, the reconfiguration flow of rationale is done by altering the execution flow. By the enhanced flow design, the bit stream size of the reconfiguration, as well as the flow's execution time, is diminished. In the given model, the circuits' logical blocks are set together. The proposed transducer circuit happens with PR link is reveals in Fig. 5.

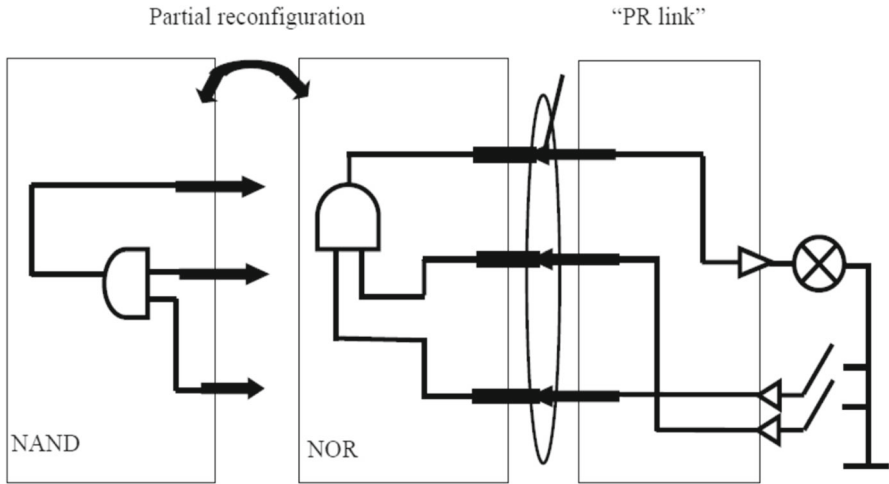


Fig. 5 Partial reconfiguration link

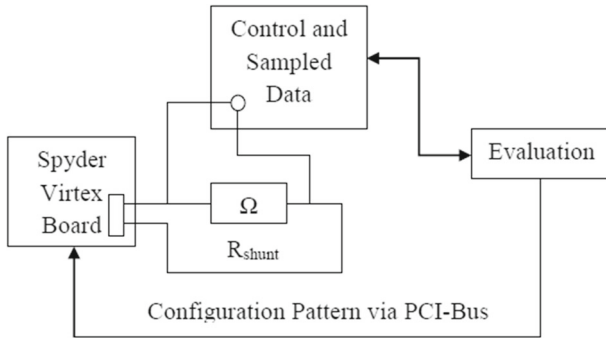


Fig. 6 Evaluation of power consumption

3.3.3 Power Dissipation

In FPGA, it appears to be more intricate for reconfiguration strategy. In this framework model, the energy efficiency is deliberate within the circuit execution. The reconfiguration’s performance happens grounded on the energy level. To boost the process speed level, the pressure of energy level in the reconfigurable frameworks is settled in the proposed circuit and it gives the least power. The power Consumption intended for the framework is gotten from the outcomes. The energy efficiency problems aren’t settled in past commercial architectures. Figure 6 outlines the power consumption’s assessment amid reconfiguration.

Table 1 Comparison in terms of Area, Power Consumption, Throughput and Latency

Metrics	Module reconfiguration (existing)	Joint memory and module reconfiguration (proposed)
Area (mm ²)	45	30
Power consumption (mW)	14	8
Throughput	39.42	41.25
Latency (μ s)	79	65

4 Result and Discussion

In the given section, the outcomes are acquired by utilizing Cadence. With the purpose of testing the performance, reused the Cadence simulations, furthermore, actualized another fuzzy system. By utilizing the range information, Design approval of testing happens. So as to decide the range, at particular interims, synchronized clocks along with ping are utilized.

4.1 Comparative Analysis

The proposed DPR of both memory and modules is contrasted with the prevailing DPR of modules regarding the area, latency, throughput, power consumption, and total clock cycles besides iterations. The metrics, for example, area, power consumption, latency together with throughput were compared in Table 1.

4.1.1 Area

In general, the area is characterized as the extent or else the measurement of a surface.

4.1.2 Power Consumption

Power consumption refers to energy per unit time, supplied to operate something (home appliances, digital circuits etc.).

4.1.3 Latency

It stands as the delay from the system input to the desired outcome. Latency extensively has an effect on how usable and also enjoyable electronic, mechanical devices in addition to communications are.

4.1.4 Throughput

It is characterized as the maximum rate upon which something be able to be processed.

Figure 7 analyses the existing together with the proposed work's performance regarding the formerly mentioned performance metrics. Centered upon the area, the

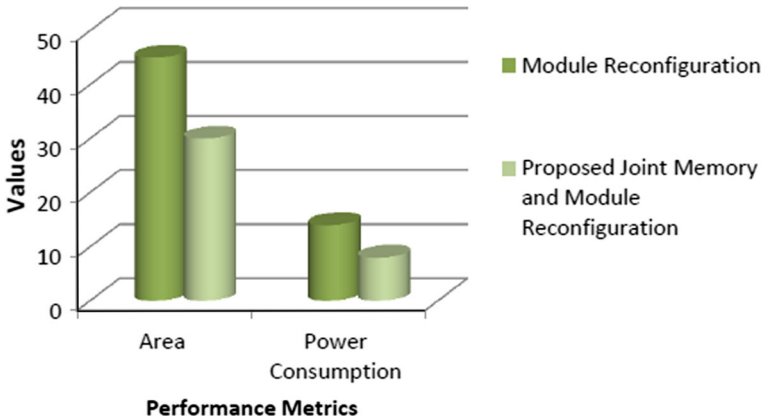


Fig. 7 Performance analysis of area and power consumption

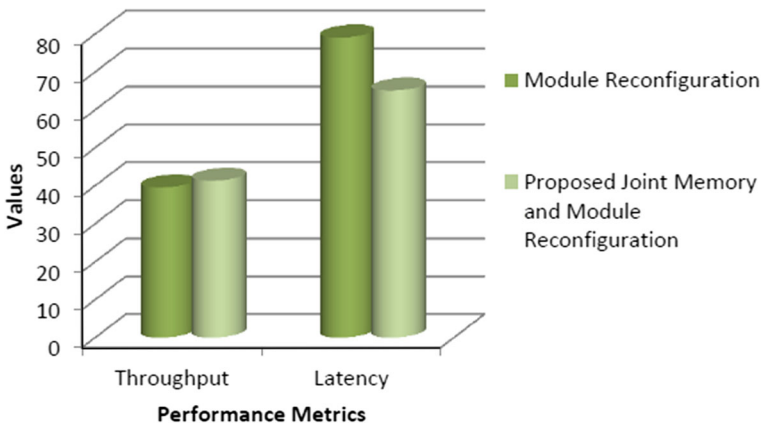


Fig. 8 Performance analysis of throughput and latency

existing technique has 45 mm² though the proposed work has 30 mm². Deliberating the power consumption, the proposed approach displays enhanced performance in which the power utilization is diminished by 6 mW. Accordingly, the proposed approach reveals the more prominent performance.

Figure 8 analyses the existing together with the proposed strategy’s performance concerning the formerly mentioned performance metrics. On thinking about throughput, the proposed work indicates relatively higher throughput when contrasted with the prevailing one. Regarding, Latency, the proposed work is 14 μs lesser than the existing approach. Henceforth, the proposed one demonstrates the predominant performance.

Table 2 exhibits the comparison of performance concerning total clock cycles along with iterations.

Table 2 Comparison in terms of total clock cycles and iterations

Metrics	Module reconfiguration (existing)	Joint memory and module reconfiguration (proposed)
Total clock cycles	1298	1153
Iterations	2	2

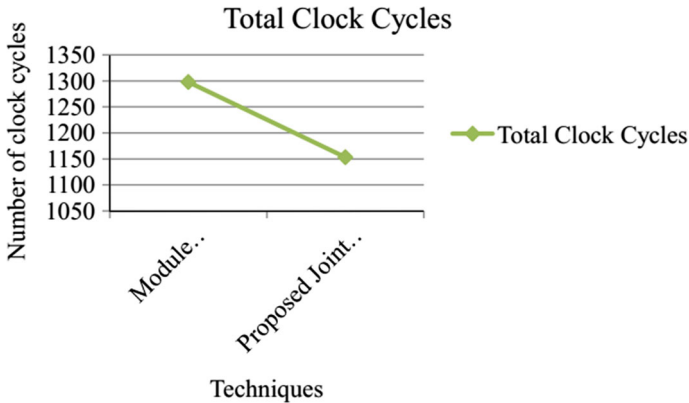
**Fig. 9** Comparison of total number of clock cycles

Figure 9 depicts the comparison of the total number of clock cycles. It as well be inferred from the table that the iterations for both the proposed and the existing are same. Moreover, concerning aggregate clock cycles, the proposed strategy has lesser number of clock cycles when contrasted with the prevailing one. The proposed joint memory and module reconfiguration has 1153 cycles while the existing requires 1298 clock cycles. Taking a gander at iterations, the existing together with the proposed work has broken even with the number of iterations.

5 Conclusion

In the conventional ASICs reconfiguration of the framework is designed in an attempt to achieve the standard of offering an FPGA device of higher performance. Here, proposed a proficient design strategy used to apply high-performance multimode memory controllers (MMMC) which fit diverse applications with various demands. The proposed design strategy relies upon the utilization of DPR to drive starting with one mode then onto the next utilizing time-multiplexing on the same chip region and furthermore reconfigure modules to spare significant region and empower the utilization of low-cost FPGAs. As far as the area, latency, throughput along with power consumption the enhanced powerfully reconfigurable frameworks are designed and then implemented. The proposed joint memory and module reconfiguration module required only 8 mW power while the existing required 14 mW power. While concerning area proposed reconfigured framework required area about 30 mm² while

existing framework required 45 mm². The proposed run-time FPGA device reconfigures the framework so as to give a superior performance in power consumption, area, and throughput. For future work, improved techniques can be implemented for safe operations and also improve the throughput and latency of dynamically reconfigurable systems at runtime.

References

1. Liu, B., Wang, B.: Reconfiguration-based VLSI design for security. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **5**(1), 98–108 (2015)
2. Contreras, L., Cruz, S., Mauricio, J., Motta, S.T., Llanos, C.H.: FPGA implementation of the EKF algorithm for localization in mobile robotics using a unified hardware module approach. In: 2015 International Conference on ReConFigurable Computing and FPGAs (ReConFig), pp. 1–6. IEEE (2015)
3. Nafkha, A., Louet, Y.: Accurate measurement of power consumption overhead during FPGA dynamic partial reconfiguration. In: 2016 International Symposium on Wireless Communication Systems (ISWCS), pp. 586–591. IEEE (2016)
4. Nouri, S., Hussain, W., Nurmi, J.: Design and evaluation of correlation accelerator in IEEE-802.11 a/g receiver using a template-based coarse-grained reconfigurable array. In: Nordic Circuits and Systems Conference (NORCAS): NORCHIP & International Symposium on System-on-Chip (SoC), 2015, pp. 1–6. IEEE (2015)
5. Alawad, M., Lin, M.: Energy-efficient imprecise reconfigurable computing through probabilistic domain transformation. In: Circuits and Systems Conference (DCAS), 2014 IEEE Dallas, pp. 1–4. IEEE (2014)
6. Nam, N., Lysecky, R.: Latency, power, and security optimization in distributed reconfigurable embedded systems. In: 2016 IEEE International Parallel and Distributed Processing Symposium Workshops, pp. 124–131. IEEE (2016)
7. Chen, Y.-G., Wen, W.-Y., Wang, Y.-T., Lee, Y.-L., Chang, S.-C.: A novel low-cost dynamic logic reconfigurable structure strategy for low power optimization. In Design Automation Conference (ASP-DAC), 21st Asia and South Pacific, pp. 250–255. IEEE (2016)
8. Gazzano, J.D.: Integrating FPGAs: a dynamically reconfigurable FPGA-based grid for high performance computing. In: International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAEEES), pp. 1–4. IEEE (2016)
9. Meidanis, D., Georgopoulos, K., Papaefstathiou, I.: FPGA power consumption measurements and estimations under different implementation parameters. In: 2011 International Conference on Field-Programmable Technology (FPT), pp. 1–6, IEEE (2011)
10. Frédéric, A., Thomas, E., Vashishtha, V.: Power consumption improvement with residue code for fault tolerance on SRAM FPGA. In 2011 Conference on Design and Architectures for Signal and Image Processing (DASIP), pp. 1–6. IEEE (2011)
11. Chtourou, S., Marrakchi, Z., Abid, M., Mehrez, H.: Power consumption analysis for mesh based FPGA. In: 2014 9th IEEE International Conference on Design & Technology of Integrated Systems In Nanoscale Era (DTIS), pp. 1–5, IEEE (2014)
12. Lawal, N., Lateef, F., Usman, M.: Power consumption measurement & configuration time of FPGA. In: Power Generation System and Renewable Energy Technologies (PGSRET), 2015, pp. 1–5, IEEE. (2015)
13. Collier, R., Fobel, C., Pattison, R., Grewal, G., Areibi, S., Jamieson, P.: Advancing genetic algorithm approaches to field programmable gate array placement with enhanced recombination operators. *Evol. Intel.* **7**(4), 183–200 (2014)
14. Palumbo, F., Fanni, T., Sau, C., Meloni, P.: Power-awareness in coarse-grained reconfigurable multi-functional architectures: a dataflow based strategy. *Journal of Signal Processing Systems* **87**(1), 81–106 (2017)
15. Fengbin, T., Yin, S., Ouyang, P., Tang, S., Liu, L., Wei, S.: Deep convolutional neural network architecture with reconfigurable computation patterns. *IEEE Trans. Very Large Scale Integr. VLSI Syst.* **25**(8), 2220–2233 (2017)

16. Zheng, Yu., Zhiyi, Yu., Xueqiu, Yu., Liu, N., Zeng, X.: Low-power multicore processor design with reconfigurable same-instruction multiple process. *IEEE Trans. Circuits Syst. II Express Briefs* **61**(6), 423–427 (2014)
17. Liu, L., Wang, D., Yin, S., Chen, Y., Zhu, M., Wei, S.: SimRPU: a simulation environment for reconfigurable architecture exploration. *IEEE Trans. Very Large Scale Integr. VLSI Syst.* **22**(12), 2635–2648 (2014)
18. Amin, R., Martin, J., Deaton, J., DaSilva, L.A., Hussien, A., Eltawil, A.: Balancing spectral efficiency, energy consumption, and fairness in future heterogeneous wireless systems with reconfigurable devices. *IEEE J. Sel. Areas Commun.* **31**(5), 969–980 (2013)
19. Purohit, S.S., Chalamalasetti, S.R., Margala, M., Vanderbauwhede, W.A.: Design and evaluation of high-performance processing elements for reconfigurable systems. *IEEE Trans. Very Large Scale Integr. VLSI Syst.* **21**(10), 1915–1927 (2013)
20. Prabhu, G.R., Johnson, B., Rani, J.S.: FPGA based scalable fixed point QRD core using dynamic partial reconfiguration. In: 2015 28th International Conference on VLSI Design (VLSID), pp. 345–350, IEEE (2015)
21. Vipin, K., Fahmy, S.A.: Mapping adaptive hardware systems with partial reconfiguration using CoPR for Zynq. In: 2015 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), pp. 1–8, IEEE (2015)