

An overview of applications of Field Programmable Gate Array (FGPA)

Hemalatha.N¹, Vineetha M², Shaila I Kolhar³

¹Selection Grade Lecturer, E&C department, Government polytechnic, Immadihalli, Whitefield
Bengaluru- 560066, Karnataka, India

²Selection Grade Lecturer, E&C department, Government Polytechnic Bagepalli, NH-7 Bagepalli –
561207, Karnataka, India

³Selection Grade Lecturer, E&C department, Government Polytechnic Immadihalli,
WhitefieldBengaluru-560066, Karnataka, India

Corresponding authors: hemalatha_parmesh@rediffmail.com
vsarathi17@gmail.com
shailavarun@gmail.com

Abstract:

A broad sense programmed logical device known as a field programmable gate array (FPGA) can be customised by a client after construction to carry out anything from straightforward logical gate activities to intricate algorithms on chips or even artificial intelligence technologies. This allows us to construct anything within a field programmable gate array, out of a straightforward logical gate to a sophisticated systems on chip or even an artificial intelligence system. This study's goal is to offer a comprehensive review of the best field programmable gate array technologies through a relevant research review of papers pertaining to field programmable gate array up to 2015. Accordingly, there are 6 categories that are used to organize the main field programmable gate arrays elements in this work. The first section includes brief introduction about the field programmable gate array, the second section includes overview and history of the field programmable gate array. The third and fourth chapters deeply discussed the designing and programming aspects of the field programmable gate array. The fifth and sixth section contains application part and security aspects of the field programmable gate array. Finally, the review ended with brief summary and motivation.

Keywords: Field programmable gate array; Applications; Security; Designing; Programming.

1. Introduction

The phrase "field-programmable" refers to an integrating circuit (ICs) that can be customised by a client or a programmer after production. This is known as a field-programmable gate array (FPGA). Generally, the hardware description language (HDL) identical to that employed for the application-specific ICs (ASICs) are employed to specify the field-programmable gate array setup. Initially, the design was specified using circuit diagram, however, with the development of electrical designing automotive toolkits, that is becoming less common [1-4]. Field-programmable gate arrays have a network of reconfigurable interconnections that permit logical blocks to be connected to one another as well as a collection of programmable logical blocks. Logical blocks can indeed be programmed to carry out intricate combinatorial operations or behave like straightforward logical gates like AND & XOR. Many field-programmable gate arrays have flash memory that can be simple switch or larger, less complex memory locations. Versatile re-configurable computation as carried out by software applications is made possible by the large number of field-programmable gate arrays that can be reconfigured to establish multiple logical functions [5-8]. Because of their capacity to begin software application developments concurrently to hardware's, facilitate process performance modelling at an extremely beginning phases of developments, and permit different network experiments and design iterations prior finalizing the conceptual framework, field-programmable gate arrays play a massive part in embedded device progress [9-13].

2. History

Programmable read-only memory (PROM) and programmable logical devices (PLD) gave rise to the field-programmable gate array business. PROM and PLD might be coded in groups either in a workshop or at the job site. The EP-300, that incorporated a quartz window inside the box that enables people to flash an ultra - violet light on the dies to delete the erasable programmable read-only memory (EPROMs) cells that retained the system configurations, became the initial re - programmable logical device produced by Altera, that has established in 1984. The XC-2064, created by Xilinx in 1986, was the initial field-programmable gate array to be financially successful.

The XC-2064 introduced programmed interconnectivity among gate and programmed gate, marking the launch of a brand-new marketplace and technologies. The XC-2064 contained 2 - 3 input search databases (ISB) and 64 customizable logical blocks (CLB). Between 1984 to the middle of the nineties, Altera and Xilinx expanded rapidly unaffected by opponents, losing a sizeable chunk of overall share in the market. Actel, which began operating in 1994, supplied around 18 percent of total of the industry. Field-programmable gate arrays saw a time of significant expansion in the nineties, as well as in terms of circuit complexity and manufacturing costs. Field-programmable gate arrays have mostly been utilized in networks and telecommunications at the beginning of the nineties. Field-programmable gate arrays entered the retail, automobile, and professional markets through the decade's conclusion. Because of the efficiency per watt advantages such field-programmable gate arrays provide, businesses including Microsoft has begun to employ them to expedite higher-performance, high computational devices (such as the data centers that run their Google web browser). Field-programmable gate arrays were first used by Microsoft in 2013 to speed up Bing, and they were then introduced to certain other datacenter applications for its Azure cloud computing platform in 2019 [14-21].

3. Design

Modern field-programmable gate arrays contain ample random-access memory (RAM) and logical gate components to carry out complicated digitized computation. The efficient and effective input/output velocities and bi-directional information lines used in field-programmable gate array configurations make it difficult to check the timeliness of genuine data during installation and holding times. Field-programmable gate arrays' floorplan helps distribution of resources to adhere to such deadlines. Whatever the logical operation that an ASICs is capable of executing can be implemented using field-programmable gate arrays. Numerous advancements can benefit from the capacity to change functionalities post delivering, incomplete system reconfiguring, and reduced non-recurring development expenses compared to an ASICs designing (despite the significantly greater unit price). Certain field-programmable gate arrays also consist analogue capabilities in additional to their digital capabilities. The most frequent analogue characteristic is a programmed descent rates on every output terminal. This technology helps engineers to configure less costs on light load ports that will normally ding or coupled unacceptably and greater rates on heavy load ports on higher-speed networks that will normally operate excessively slow. In addition, phase-locked loop containing integrated voltage control oscillator, quartz-crystal oscillator, on-chip resistance-capacitance oscillator, and oscillator utilized for higher-speeds serializer-deserializer (SER-DES) transmitter clocks and receiver clock recoveries are frequently employed. Various converters on input port intended for connection to differential signaling pathways are rather frequent. A small number of "mixed signal field-programmable gate arrays" include peripherals analog-to-digital converter (ADC) and digital-to-analog converter (DAC) combined with analog input conditioner modules, enabling it to function like a system-on-a-chips (SoCs). These gadgets blur the distinction among a field-programmable gate array (FPGAs), that also uses its intrinsic programmed interconnecting fabric to carry electronic 1s and 0s, and a field-programmable analogue arrays (FPAAs), which uses that same intrinsic programmed interconnecting fabric to carry analogue values [22-25].

3.1 Logic constructs

Logical blocks, input/output plates, and router pathways make up one of most typical field-programmable gate array design. These blocks may be referred to as customizable logical block (CLB) or logical array block (LAB), based on the provider. Typically, the breadth of each route channel is identical. The breadth among one column or perhaps the elevation of one row with in arrays could both accommodate multiple input/output ports. It is essential to convert a software network it in to a field-programmable gate array with enough capabilities. The amount of CLB/LAB and input/outputs required can be predicted accurately from the model, but even across models that have the similar number of logics, the quantity of route channels necessary can differ significantly. (For instance, a crossbar shift needs to be routed considerably greater than a cyclic array that has the exact number of gates. Field-programmable gate array producers strive to supply only sufficient paths so that because the majority of design features which might accommodate in aspects of lookup table (LUT) and input/outputs may be routed because unutilized routing paths raise the expense (as well as degrade the efficiency) of the component without having provided any advantage. Predictions, including those obtained using Rent's rule, or trials using preexisting models are used to establish it. 2017 saw the development of network-on-chip designs for routing and connectivity [26-29]. The logical blocks typically comprise of a few logical cells. A four-input lookup tables, a full adder (FA), and a D-type switch make up a typical cell. You might divide that into 2 three-input lookup tables. In regular mode via the initial multiplex, that are concatenated it in to a four-input lookup table. These outcomes have been delivered into the adder when it is in arithmetical modes. The secondary mux has the mode selection preprogrammed. Based on how the 3rd mux is programmed, the outcome could either be asynchronized or synchronized. In a bid to

conserve memory, complete or portions of the adder are typically encoded as functionalities in the lookup tables [27-30].

3.2 Hard blocks

Sophisticated silicon-fixed better level capability is added to the aforementioned functionalities by field-programmable gate array groups. In comparison to creating these basic functionalities using logic cavemen, putting those integrated with in circuit minimizes the space needed and increases their pace. Multiplier, general DSP block, embedded systems, top bandwidth input and output logic, and embedded storages are a few characteristics of such. Higher-speed multi-giga-bit devices including hard input portal cores like CPU components, Ethernet medium access management modules, PCI controller, and external storage control systems can be found in high-end field-programmable gate arrays. Such cores coexist well with programmed network, however since they are made of transistor rather than lookup table, they possess capacity and power usage comparable to ASICs without employing a sizable portion of the fabric's resource, freeing up additional space for the application-specific logics. Highly engineered analogue I/O circuitry, as well as higher-velocity serializers and deserializers, are additional features of multi-giga-bit transmitters that can't be constructed using lookup tables. Based on the field-programmable gate array, higher-level physical layer (PHY) capability such as line coding might or may not be provided beside the serializers and deserializers in hard logics [30-33].

3.3 Softcore

Employing softer processing Internet protocol blocks which are embedded inside the field-programmable gate array logics is an alternative to the use of hard-microprocessors. Prominent multiprocessor includes Nios II, Micro-Blaze, and Mico-32. Reprogrammable computation, also known as re-configurable system, refers to central processing units that can change their configuration to match the job in question. This concept was made possible by the fact that several contemporary field-programmable gate arrays may be configured at "run time." Incremental approach for non-field-programmable gate arrays is also starting to appear [30,31-35].

4. Programming

The client submits a model in a design specification or a hardware description language (HDLs) to specify the behavior of the field-programmable gate array. Larger frameworks can be worked with many more effectively using the hardware description languages format because higher-order structural behavior can be specified instead of having to construct each component manually. Nevertheless, graphical entries may make it simpler to visualize a model as well as its individual components. An electrical circuit design tooling is used to develop a configuration that is innovation-mapped. Place-and-route, which is often carried out through the field-programmable gate array manufacturer's patented place-and-route technology, is then employed to match the configuration to the real field-programmable gate array design. Time assessment, modeling, as well as other methods of confirmation and validation are going to be used by client to verify the mapping, location, and routing outcomes. The field-programmable gate array is reconfigured to use the produced binary code, which is normally created using the user's commercial systems after the designing and verification processes have been finished. The above document is sent from the external storage source, for example an EEPROMs, or a serial interface (J-TAG), to the field-programmable gate array. Verilog hardware description languages and Verilog, together with their derivatives like Systems Verilog, are indeed the two most popular hardware description languages. Nevertheless, there's many efforts to enhance the abstraction layer thru the creation of additional language inside an effort to lessen the difficulty of developing in hardware description languages, which were contrasted to the equivalents of programming language. A field-programmable gate array add-in modules for the Lab-VIEW graphic software package (often known to as "G") is accessible to address and programmed field-programmable gate array technology. Verilog was developed to streamline the method and make hardware description languages more reliable and adaptable. Right now, Verilog is the most widely used. To obscure the specifics of its implementations, Verilog establishes a degree of abstraction. Unlike Verilog hardware description languages, Verilog does have a syntax more akin to C. There seem to be library of prefabricated complicated functions and networks for field-programmable gate arrays that were validated and accelerated to facilitate the construction of complicated processes. Such pre-designed circuit, also known as intellectual property (IP) cores, are offered by field-programmable gate array (FPGA) manufacturers and independent intellectual property providers. These are often published using proprietary licenses and are infrequently freeware. Additional preset networks are accessible via developer communities like Public Cores (usually distributed with freely available and open-source licenses like the GPL, BSD, or comparable license), among several other outlets. These concepts are referred to as open-source hardware [36-40].

A field-programmable gate array app developer would replicate the model at various phases across the construction phase according to a normal design flow. Testing benches are originally created to replicate the systems and track outcomes in order to mimic the RTL definition in Verilog hardware description languages or Verilog. The

configuration is transformed into a gate-level specification and simulations are conducted to ensure the synthesis process went well once the synthesizing engines have translated the idea to a template. Dissemination delay may then be incorporated into the design once it has been set across the field-programmable gate array, where at time the simulations can be performed instead with those values re-annotated into the template [37,40].

Software developers are now using Open Computer Language) to benefit from the performance and energy savings that field-programmable gate arrays offer. By employing Open Computer Language structures, developers can execute scripts in the C language that targets field-programmable gate array effects as Open Computer Languages kernels. Consider higher-order synthesizing and C to hardware description languages for more details. The majority of field-programmable gate arrays are reprogrammed using an SRAM-dependent method. Although these field-programmable gate arrays need additional boot mechanisms, they are in-system configurable and reprogrammable. Flash storage or EEPROM systems, for instance, frequently automatically download into inbuilt SRAM that regulates sequencing and logical. SRAM technology relies on CMOS [38-41].

5. Applications

Whatever challenge that is able to solved computationally can be solved using a field-programmable gate array. The reality that field-programmable gate arrays have the ability to incorporate smooth microcontrollers like the Xilinx Micro-Blaze or Altera Nios II trivially proves the above. Due to their own parallelism and the appropriate number of entryways those who are using for specific methods, they have the added benefit of being considerably quicker for certain application domains. Field-programmable gate arrays were initially developed as CPLDs' rivals for implementing sticky reasoning for printed boards. Field-programmable gate arrays began to perform more tasks as their own size, power, and pace continued to increase to the moment whereby most of them have presently advertised as complete systems on chips (SoC). Software packages that had previously only used digital signal processor hardware (DSPH) began to use field-programmable gate arrays, especially after the addition of devoted multiplications to field-programmable gate array frameworks in the 90s [42,43].

The popularity of such computers has increased as a result of the advancement of field-programmable gate arrays, for whom the design enables the creation of troubleshooting tools that are optimized for complicated jobs like three-dimensional MRI image segmentations, three-dimensional discrete wavelet transforms, and computed tomography image restoration. The advanced remedies could indeed encounter the strict requirements related with healthcare microscopy also while being inexpensive, being interactively reconfigurable, and performing focused parallel computing simultaneous tasks.

Hardware speed seems to be another tendency in the utilization of field-programmable gate arrays, in which the field-programmable gate array has the ability to speed up specific portions of a method and split some of the arithmetic operations among it and a conventional central processing unit. In 2013, the web search Bing made headlines for implementing field-programmable gate array velocity in its algorithm. 2015 has seen a rise in the use of field-programmable gate arrays as Intelligence activators, which include Microsoft's infamous "Project Catapult" and for speeding up artificial neural systems for applications involving machine learning [44].

Field-programmable gate arrays have typically been used in specialized lateral applications that require small capacity utilization. For such reduced application domains, a configurable chip's higher hardware price per unit is much more cost-effective than that of the time and money required to create an ASICs. As of 2016, more implementations are now feasible due to shifting pricing and performance kinetics. Giga-byte Technique exploits an i-random access memory card using a Xilinx field-programmable gate array, and although mass-producing a specially made processor would have been more cost-effective. The field-programmable gate array was selected because it would be quickly brought to market and because the official release was limited to 1000 items, which made it the best option. With the aid of this gadget, random access memory can be used as a hard disk drive [45-47].

6. Security

In regards to hardware protection, field-programmable gate arrays include both benefits and drawbacks especially in comparison to ASIC or protected micro controllers. Due to the flexibility of field-programmable gate arrays, the threat of malevolent manufacturing adjustments is reduced. The layout data stream has been earlier revealed when the field-programmable gate array loaded it from external storage for a number of these devices (typically on every power-on). A variety of security features, including bit - stream authentication and encryption are already provided to developers by all significant field-programmable gate array supplier. For bit-streams maintained in an externally flash storage, Altera and Xilinx, for instance, provide Encryption algorithm (up to 256-bit) [48].

Field-programmable gate arrays, like the Pro-Asic 3 or XP2 programmed devices from Microsemi, which keep their setup from within in non-volatile flash storage, do not reveal the stream of bits and don't require encryption. In contrast, specific incident upsetting security for aircraft applications is provided by flash storage for a lookup tables. Consumers could use write-once, anti-fuse field programmable gate arrays from suppliers like Micro-semi if they want a greater assurance of contaminate impedance [49-51].

7. Summary and future scope

Field programmable gate arrays have a wide variety of uses in contemporary computer systems, as was already mentioned. The findings include development, trending's, and co-relations for the leading application areas on field programmable gate arrays. In this approach, emerging applications and emerging technology trends are demonstrated. Similarly, to the way other types of apps, like cancer detection methods, indoors geolocation utilizing arrival time or even more effective big data processing techniques, could drive programmers to find new and more effective ways to address real-world issues.

This article provides a thorough foundation on field programmable gate arrays-based applications for academics, graduates, consumers, business, publications, and funded organizations. The summary of field programmable gate arrays-based expedited applications presented here enables researcher students and business to increase the effectiveness of technologies that have already been put into use. Additionally, the deployment of field programmable gate arrays for the speeding of novel types of applications is made possible by the implementation of expedited techniques and methodologies. The final limitation of our endeavor was the two core databases for scientific articles (Scopus and web of science). Additional researches could look at these industrial applications, assessing and contrasting their patterns with those of other resources like Google Analytics and others.

References:

1. Trimmerger, S. *Field-Programmable Gate Array Technology*; Springer US: New York, NY, USA, 2012.
2. Hamouda, M.; Blanchette, H.F.; Al-Haddad, K.; Fnaiech, F. An Efficient DSP-FPGA-Based Real-Time Implementation Method of SVM Algorithms for an Indirect Matrix Converter. *IEEE Trans. Ind. Electron.* 2011, 58, 5024–5031. doi:10.1109/TIE.2011.2159952.
3. Pozniak, K.; Czarski, T.; Romaniuk, R. Functional analysis of DSP blocks in FPGA chips for application in TESLA LLRF system. In *Proceedings of the 12th IEEE-SPIE Symposium on Photonics and Web Engineering*, Warsaw Univ Technol Resort, Wilga, Poland, 21–25 May 2003.
4. Huang, J.B.; Xie, Z.W.; Liu, H.; Sun, K.; Liu, Y.C.; Jiang, Z.N. DSP/FPGA-based controller architecture for flexible joint robot with enhanced impedance performance. *J. Intell. Robot. Syst.* 2008, 53, 247–261. doi:10.1007/s10846-008-9240-7.
5. Diaz, J.; Ros, E.; Pelayo, F.; Ortigosa, E.; Mota, S. FPGA-based real-time optical-flow system. *IEEE Trans. Circuits Syst. Video Technol.* 2006, 16, 274–279. doi:10.1109/TCSVT.2005.861947.
6. Kalomiros, J.A.; Lygouras, J. Design and evaluation of a hardware/software FPGA-based system for fast image processing. *Microprocess. Microsyst.* 2008, 32, 95–106. doi:10.1016/j.micpro.2007.09.001.
7. Hegarty, J.; Brunhaver, J.; DeVito, Z.; Ragan-Kelley, J.; Cohen, N.; Bell, S.; Vasilyev, A.; Horowitz, M.; Hanrahan, P. *Darkroom: Compiling High-Level Image Processing Code into Hardware Pipelines*. *ACM Trans. Graph.* 2014, 33. doi:10.1145/2601097.2601174.
8. Rajagopalan, S.; Amirtharajan, R.; Upadhyay, H.; Rayappan, J. Survey and analysis of hardware cryptographic and steganographic systems on FPGA. *J. Appl. Sci.* 2012, 12, 201–210. doi:10.3923/jas.2012.201.210.
9. Kean, T. Cryptographic rights management of FPGA intellectual property cores. In *Proceedings of the FPGA 2002: Tenth ACM International Symposium on Field-Programmable Gate Arrays*, Monterey, CA, USA, 24–26 February 2002.
10. Chakraborty, D.; Raha, P.; Bhattacharya, A.; Dutta, R. Speed optimization of a FPGA based modified Viterbi Decoder. In *Proceedings of the 3rd International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, 4–6 January 2013.
11. Tsai, C.C.; Huang, H.C.; Chan, C.K. Parallel Elite Genetic Algorithm and Its Application to Global Path Planning for Autonomous Robot Navigation. *IEEE Trans. Ind. Electron.* 2011, 58, 4813–4821. doi:10.1109/TIE.2011.2109332.
12. Jarvinen, K.; Skytta, J. On parallelization of high-speed processors for elliptic curve cryptography. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* 2008, 16, 1162–1175. doi:10.1109/TVLSI.2008.2000728.
13. Keymeulen, D.; Zebulum, R.; Jin, Y.; Stoica, A. Fault-tolerant evolvable hardware using field-programmable transistor arrays. *IEEE Trans. Reliab.* 2000, 49, 305–316. doi:10.1109/24.914547.
14. Cheatham, J.A.; Emmert, J.M.; Baumgart, S. A survey of fault tolerant methodologies for FPGAs. *ACM Trans. Des. Autom. Electron. Syst.* 2006, 11, 501–533. doi:10.1145/1142155.1142167.
15. Emmert, J.M.; Stroud, C.E.; Abramovici, M. Online fault tolerance for FPGA logic blocks. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* 2007, 15, 216–226. doi:10.1109/TVLSI.2007.891102.

16. Li, F.; Lin, Y.; He, L.; Cong, J. Low-power FPGA using pre-defined dual-Vdd/dual-Vt fabrics. In Proceedings of the ACM/SIGDA Twelfth ACM International Symposium on Field-Programmable Gate Arrays—FPGA 2004, Monterey, CA, USA, 22–24 February 2004.
17. Zhou, Y.; Thekkel, S.; Bhunia, S. Low Power FPGA Design Using Hybrid CMOS-NEMS Approach. In Proceedings of the 12th International Symposium on Low Power Electronics and Design, Portland, OR, USA, 27–29 August 2007.
18. Lee, D.; Luk, W.; Villasenor, J.; Cheung, P. A Gaussian noise generator for hardware-based simulations. *IEEE Trans. Comput.* 2004, 53, 1523–1534. doi:10.1109/TC.2004.106.
19. Tan, Z.; Waterman, A.; Avizienis, R.; Lee, Y.; Cook, H.; Patterson, D.; Asanovic, K. RAMP Gold: An FPGA-based Architecture Simulator for Multiprocessors. In Proceedings of the 47th Design Automation Conference (DAC), Anaheim, CA, USA, 13–18 June 2010.
20. Hasanzadeh, A.; Edrington, C.S.; Stroupe, N.; Bevis, T. Real-Time Emulation of a High-Speed Microturbine Permanent-Magnet Synchronous Generator Using Multiplatform Hardware-in-the-Loop Realization. *IEEE Trans. Ind. Electron.* 2014, 61, 3109–3118. doi:10.1109/TIE.2013.2279128.
21. Buccella, C.; Cecati, C.; Latafat, H. Digital Control of Power Converters-A Survey. *IEEE Trans. Ind. Inform.* 2012, 8, 437–447. doi:10.1109/TII.2012.2192280.
22. Bayam, K.A.; Ors, B. Differential power analysis resistant hardware implementation of the RSA cryptosystem. *Turk. J. Electr. Eng. Comput. Sci.* 2010, 18. doi:10.3906/elk-0904-4.
23. Monmasson, E.; Cirstea, M.N. FPGA design methodology for industrial control systems - A review. *IEEE Trans. Ind. Electron.* 2007, 54, 1824–1842. doi:10.1109/TIE.2007.898281.
24. Sun, F.; Wang, H.; Fu, F.; Li, X. Survey of FPGA low power design. In Proceedings of the 2010 International Conference on Intelligent Control and Information Processing, Dalian, China, 13–15 August 2010. doi:10.1109/ICICIP.2010.5565246.
25. Hadiyoso, S.; Astuti, R.P.; Hidayat, I. Design of an FPGA-Based OFDM-STBC Transceiver for WiMAX 802.16e Standard. In Proceedings of the 2nd International Conference on Information and Communication Technology (ICoICT), Bandung, Indonesia, 28–30 May 2014.
26. Antichi, G.; Di Pietro, A.; Giordano, S.; Procissi, G.; Ficara, D. Design and Development of an OpenFlow Compliant Smart Gigabit Switch. In Proceedings of the 54th Annual IEEE Global Telecommunications Conference (GLOBECOM), Houston, TX, USA, 5–9 December 2011.
27. Li, Y.F.; Chuang, L.L. Controller design for music playing robot - applied to the anthropomorphic piano robot. In Proceedings of the IEEE 10th International Conference on Power Electronics and Drive Systems (PEDS), Kitakyushu, Japan, 22–25 April 2013.
28. Li, Y.F. FPGA-based module design for PM linear motor control-applied to music playing robot. In Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE), Taipei, Taiwan, 28–31 May 2013.
29. Li, Y.F. FPGA-Based Distributed Control Module Design for Music Playing Robot - Applied to the Anthropomorphic Piano Robot Control. *J. Chin. Soc. Mech. Eng.* 2013, 34, 143–150.
30. Costillo, L.; Ramos, J.; Ibanez, J.; Aparicio, B.; Herranz, M.; Garcia, A. New control system for the 1.5m and 0.9m telescopes at Sierra Nevada Observatory. In Proceedings of the Advanced Software and Control for Astronomy, Orlando, FL, USA, 24–26 May 2006. doi:10.1117/12.671308.
31. Cherif, S.; Trabelsi, C.; Meftali, S.; Dekeyser, J.L. High level design of adaptive distributed controller for partial dynamic reconfiguration in FPGA. In Proceedings of the 2011 Conference on Design and Architectures for Signal and Image Processing, DASIP 2011, Tampere, Finland, 2–4 November 2011. doi:10.1109/DASIP.2011.6136896.
32. Trabelsi, C.; Meftali, S.; Dekeyser, J.L. Distributed control for reconfigurable FPGA systems: a high-level design approach. In Proceedings of the 7th International Workshop on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), York, UK, 9–11 July 2012.
33. Trabelsi, C.; Meftali, S.; ben Atitallah, R.; Dekeyser, J.L. Model-Driven design flow for distributed control in reconfigurable FPGA systems. In Proceedings of the 8th Conference on Design and Architectures for Signal and Image Processing (DASIP), Madrid, Spain, 8–10 October 2014.
34. Chakraborty, D.; Raha, P.; Bhattacharya, A.; Dutta, R. Speed optimization of a FPGA based modified Viterbi Decoder. In Proceedings of the 3rd International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 4–6 January 2013.
35. Vestias, M.; Neto, H.; Sarmento, H. Design of high-speed viterbi decoders on virtex-6 FPGAs. In Proceedings of the 15th Euromicro Conference on Digital System Design, DSD 2012, Cesme, Izmir, 5–8 September 2012. doi:10.1109/DSD.2012.42.

36. Shu, Z.; Guo, Y.; Lian, J. Steady-state and dynamic study of active power filter with efficient FPGA-based control algorithm. *IEEE Trans. Ind. Electron.* 2008, 55, 1527–1536. doi:10.1109/TIE.2008.917151.
37. Mellit, A.; Kalogirou, S.A. Artificial intelligence techniques for photovoltaic applications: A review. *Prog. Energy Combust. Sci.* 2008, 34, 574–632. doi:10.1016/j.peccs.2008.01.001.
38. Punitha, K.; Devaraj, D.; Sakthivel, S. Artificial neural network based modified incremental conductance algorithm for maximum power point tracking in photovoltaic system under partial shading conditions. *Energy* 2013, 62, 330–340. doi:10.1016/j.energy.2013.08.022.
39. Juang, C.; Chen, J. Water bath temperature control by a recurrent fuzzy controller and its FPGA implementation. *IEEE Trans. Ind. Electron.* 2006, 53, 941–949. doi:10.1109/TIE.2006.874260.
40. Uchida, T. Hardware-based TCP processor for Gigabit Ethernet. In *Proceedings of the IEEE Nuclear Science Symposium/Medical Imaging Conference, Honolulu, HI, USA, 26 October–3 November 2007.*
41. Bomel, P.; Crenne, J.; Ye, L.; Diguet, J.P.; Gogniat, G. Ultra-Fast Downloading of Partial Bitstreams through Ethernet. In *Proceedings of the 22nd International Conference on Architecture of Computing Systems, Delft Univ Technol, Delft, The Netherlands, 10–13 March 2009.*
42. Kung, Y.S.; Fung, R.F.; Tai, T.Y. Realization of a Motion Control IC for X-Y Table Based on Novel FPGA Technology. *IEEE Trans. Ind. Electron.* 2009, 56, 43–53. doi:10.1109/TIE.2008.2005667.
43. Del Campo, I.; Echanobe, J.; Basterretxea, K.; Bosque, G. Scalable architecture for high-speed multidimensional fuzzy inference systems. *J. Circuits Syst. Comput.* 2011, 20, 375–400. doi:10.1142/S0218126611007359.
44. Qin, W.; Zhou, J.; Li, C. Research of Continuous Variable Camshaft Timing system based on fuzzy-PID control method. In *Proceedings of the 2011 2nd International Conference on Mechanic Automation and Control Engineering, Hohhot, China, 15–17 July 2011.*
45. Hwu, K.L.; Yau, Y.T. Performance Enhancement of Boost Converter Based on PID Controller Plus Linear-to-Nonlinear Translator. *IEEE Trans. Power Electron.* 2010, 25, 1351–1361. doi:10.1109/TPEL.2009.2036727.
46. Chander, S.; Agarwal, P.; Gupta, I. FPGA-based PID controller for DC-DC converter. In *Proceedings of the 2010 Joint International Conference on Power Electronics, Drives and Energy Systems & 2010 Power India, New Delhi, India, 20–23 December 2010.*
47. Sivaramakrishna, M.; Upadhyay, C.; Nagaraj, C.; Madhusoodanan, K. Development of pid controller algorithm over FPGA for motor control in failed fuel location module in Indian fast reactors. In *Proceedings of the 2011 3rd International Conference on Electronics Computer Technology, ICECT 2011, Kanyakumari, India, 8–10 April 2011.*
48. Bai, J.; Wu, L.; Yun, N.; Liu, Y.; Zhang, X. A 10Gbps In-line Network Security Processor with a 32-bit Embedded CPU. In *Proceedings of the 22nd Wireless and Optical Communications Conference (WOCC), Chongqing Univ Posts & Telecommunicat, Chongqing, China, 16–18 May 2013.*
49. Niu, Y.; Wu, L.; Wang, L.; Zhang, X.; Xu, J. A configurable IPsec processor for high performance in-line security network processor. In *Proceedings of the 2011 7th International Conference on Computational Intelligence and Security, CIS 2011, Sanya, China, 3–4 December 2011.* doi:10.1109/CIS.2011.154.
50. Wang, A.; Li, Z.; Yang, X.; Feng, B. A new security problem of USB: Monitoring cable attack and countermeasures. In *Proceedings of the 2012 International Conference on Information Technology and Software Engineering, ITSE 2012, Beijing, 8–10 December 2012.* doi:10.1007/978-3-642-34522-7_15.
51. Yang, X.; Li, Z.; Wang, A.; Zhang, Y. System design of cryptographic security USB device controller IP core. *Huazhong Keji Daxue Xuebao (Ziran Kexue Ban)/J. Huazhong Univ. Sci. Technol. (Nat. Sci. Ed.)* 2010, 38, 59–62.