ISSN NO: 2770-4491
Date of Publication:24-08-2022

Intelligent FPGA Based Design and Implementation of Digital Systems

Asmaa Hameed Rasheed
Al Nahrain University
Baghdad, Iraq
d.asmaa.mas@gmail.com

Abstract— FPGAs as a digital design based platforms, are usually candidates over DSP processors and many other digital hardware platforms for real time systems. FPGAs products for different companies have various number of available resources. So, the numbers of configurable logic blocks and input / output ports are varied for different versions. The wrong choice for the suitable FPGA plant for specific applications leads to do not have full utilities from the used FPGA processor, and may use only less than 10% from its resources that make the designer lose more than 90% of the processor utilities. In this paper, intelligent systems is built to determine the most suitable Xilinx processor depending on its available resources and other specific criteria, such as the operation speed of real time processing. The intelligent system can also choose the suitable partitioning algorithms and segmentation techniques for processing of specific applications with high data density. The suggested intelligent system built on an extended data base of different FPGA processors with comprehensive information about available resources of each version with detail information about frequency, processing speed and many other specific features of each processor. This intelligent system must include a wide experience of different digital technique and adapting algorithms as well as partitioning and segmentation criteria. Thus, an intelligent system that supported by this wide experiences and activities can provide a perfect decision to avoid any lose for processor components and utilities by efficiently determining the suitable FPGA processor for specific applications. As well as by choosing more efficient data processing technique. The suggested intelligent FPGA based system is built using Xilinx System Generator block sets, and Xilinx software of ISE 14.7 cooperated with MATLAB R2013b. The used Xilinx processors in this paper are Spartan, Artix, Kintex, and Virtex processors.

Keywords— FPGA, Xilinx System Generator (XSG), digital system design, intelligent systems.

I. INTRODUCTION

FPGA capacity increases rapidly to tens of millions system gates that makes it the more suitable platform for many DSP hardware design and implementation. It consists of a regular and flexible configurable logic cells and input/output blocks with powerful routing interconnections of these basic elements. All these parts are programmable that can be reprogrammed completely with the specified application. FPGA also provides both single-port and dual-port RAM which are very important for many real time processing.

Many digital signal processing depend mainly on FPGAs in their design and hardware implementation. Filtering processing is one of the main DSP application that can be effectively designed and implemented using the FPGA kit [1], [2], to be the basis for many real time applications.

Various hardware platforms can be depended upon for implementation of real time signal processing. Each platform has its own strengths and weakness, FPGAs offer unique and specialty configurations for various signal processing applications. FPGAs are still the best candidate choice due to their reconfigurable gate structure and MIPS requirements of operation, as well as the fact that FPGAs offer low design cost.

FPGAs are usually used with many high data density DSP applications such as modern digital image processing with real time systems, since FPGAs are good at flexibility, reprogramming and parallelism. The processing speed is acceptable and gives a fast output response compared to other available digital hardware platforms [3].

In spite of all FPGAs utilities, the system with high data density can impact and parallelize its efficiency due to the restricted number of I/O ports available for each FPGA and Xilinx processors [1]. This problem

ISSN NO: 2770-4491
Date of Publication:24-08-2022

can be solved by the assistance of external RAM synchronized with the Xilinx processor. Xilinx System Generator (XSG) can contribute in solving this problem and also submits many other design facilities related to any digital system design.

The choice of a suitable technique to treat the systems of high data densities is in fact a big problem. The most suitable Xilinx product, to be dependant, in any digital system design needs an extended experience of all features and conditions of the available digital hardware. This makes the use of expert system a very essential with any real time applications.

II. SYSTEM GENERATOR

One of the main parts of Simulink library of Matlab engineering package is System Generator tool box. This tool box is known as Xilinx Block Set (XBS). The function of SG, is mapping the Xilinx block elements design to the FPGA internal implemented circuits. It also provides system tools for HDL simulation and implementation.

Xilinx System Generator (XSG), from a professional point of view, is a DSP design tool that submit a block set of the main DSP operations such as convolusion, correlation and so on. This block set also include the main required blocks for most DSP system designs, such as Delay block , digital filters blocks (FIR & IIR), and so many other important blocks of DSP design. By using the modules of XSG library, the System Generator design can be converted automatically to HDL codes which can easily executed with FPGA processors.

With the aid of XSG the DSP design becomes very easy and not conditionally the designer is an experian with VHDL language. So he do does not need a depth knowledge of FPGA programming, and the only thing that is needed for the DSP designer is his ability to establish a real system design of different digital signal processing applications. but only needs one for a DSP Simulink modeling environment. To generate an FPGA programming file, and then downstream its code the synthesis and routing processes must be down, which are automatically performed. The System Generator integrates the design and FPGAs that combine the RTL and Simulink in an united simulation and implementation environment.

The Xilinx Blockset is a classified groups of Simulink blocks to enable the designer to simulate designs within this environment. This ready blockset are created to familiar Simulink idioms wherever possible.

When any function is not found in the Xilinx Block Set, then this missed function can be divided as an integrated modules. XSG supports a black box block so the RTL is imported to Simulink. Thus the design with System Generator block set, offers the possibility to access predefined functions such as high-speed multipliers. The other important design flexibility is the ability to incorporate user defined VHDL blocks in the processed model.

The Co-simulator of XSG and HDL provide necessary design requirements such as : analyze system diagnose fault, as well as evaluate its performance. This can reduced system processing time with satisfying special specifications of any design.

The test benches can be generated automatically by XSG tool box with its functional block set. Creation of test bench is an important step to ensure the system validation and give initial expectation about the system performance. The step is similar to the simulation part of VHDL package

III. SYSTEM DESIGN USING XSG

The System Generator is one of the simplest FPGA design tools. It represents match ring between the high level digital system design and the FPGA hardware implementations without the need to have any professional experience in the languages details of FPGAs. Designers just need to build the model in Simulink, and more precisely System Generator Block Set, while the VHDL codes can be generated automatically, that leads to implement the model on the FPGA device [4]. The design process of the System Generator can be summarized as follows:

1. Give mathematical definition of the system.
2. Choose suitable Simulink model of the mathematical function of the required system from XSG block set.

ISSN NO: 2770-4491
Date of Publication:24-08-2022

3. Automatically generate HDL codes and project files.
4. Debugge the obtained HDL codes with software package (ISE).
5. Download the generated code to the chosen FPGA kit.

These steps are well illustrated in Fig. 1.

Xilinx block set consists of a system generator (SG) block which is responsible of the system control and a black box block that implements the majority voting circuit in VHDL. Many other function blocks are available in XSG block sets that include different DSP and data processing operations. Gateway IN and gateway OUT are also parts of the Xilinx block set, which are used for type conversion from Matlab data type to Xilinx data type.

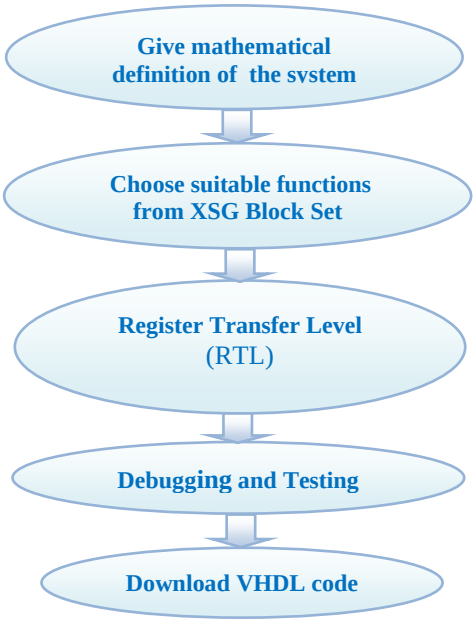

Fig. 1. System design using XSG.

IV. Resources Density

The main resources that are considerable in any FPGA BASED digital system design are the available look-up tables (LUTs), flip-flops, input & output ports (I/O ports) and DSP slices. These resources should be optimized to get full utilization of FPGA processors and they represent the indices for comparative analysis.

In addition to the implementation approach, resource utilization is another key aspect of FPGA-based system control implementation. The system control implementation based on FPGA resource utilization is compared in [11] and [12] for Xilinx FPGA.

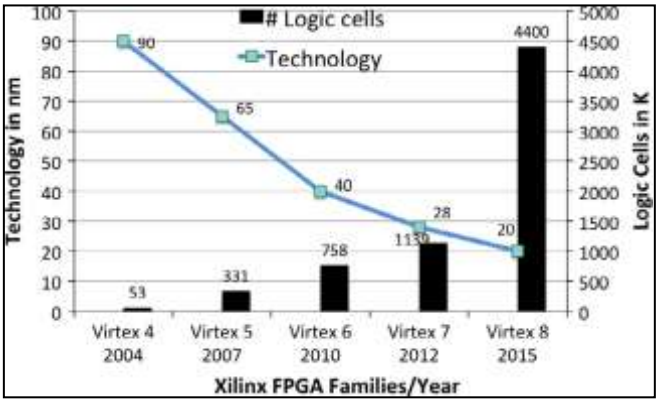

Fig. 2. Available Resources for Xilinx FPGA families.

ISSN NO: 2770-4491
Date of Publication:24-08-2022

A. Inteligent System

The overall system design will be classified into two main parts. The first part is the expert system and the second one is the XSG design of the implemented system depending on the optimal decision that was given by the expert system. The other important part that is included in the expert system, is the very large and extended database. This database can be updated continuously during the life time of this system, according to the newest production and available solutions for digital signal processing design tools.

The expert system must have a comprehensive knowledge depending on a very wide data base and a big information area of all available FPGA productions and different versions of Xilinx processors. The user must submit the critical conditions and specific design criteria for the demanded system design for any digital signal processing application. One of the most important information about the FPGA kits and different Xilinx processors is the available input output pins.

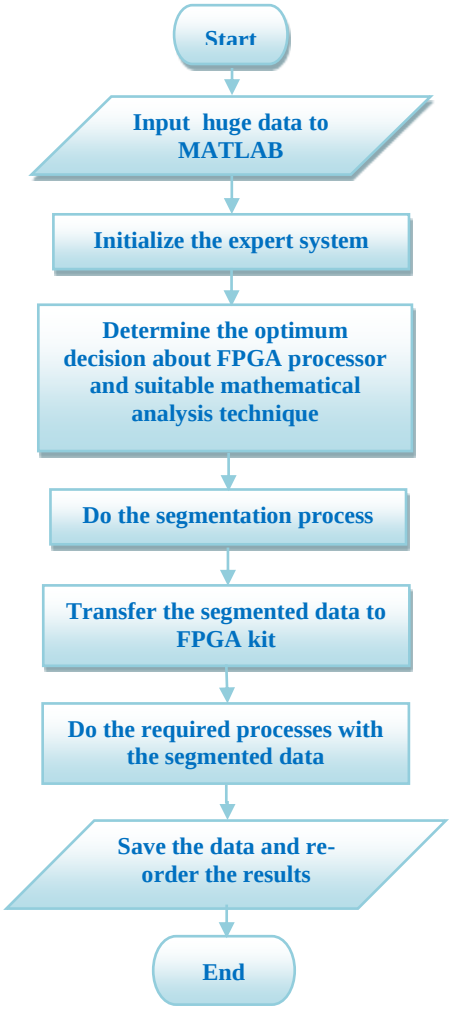

Fig. 3. Flowchart of FPGA-based design with expert system.

The solution of this problem is segmenting these data to a fixed block length that match the FPGA I/O restrictions. These blocks are translated to FPGA individually. XSG supports a black box block so the RTL is imported to Simulink. This is a very useful facility that can be used due to the designer's special requirements. Hence, image filtering algorithm can be done within this block.

Before starting the FPGA software many parameters must be known in order to chose, the more suitable FPGA products, and also to determine the preferable used techniques. The decision must be made by an expert system to make a trade-off between different emergent demands depending on the available data and products

ISSN NO: 2770-4491
Date of Publication:24-08-2022

possibilities. As shown in Fig. 4 the data transition with the data base is bidirectional in order to deliver the available data and to take design specification. Also, this data base can be updated and developed continuously according to the new Xilinx products and processor manufacturing technology.

B. Data Base

In fact the inventor of the Field Programmable Gate Array is Xilinx, and their hardware programmable product can deliver the most efficient and dynamic processors.

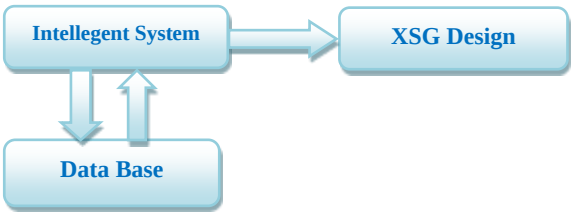

Fig. 4. Expert system architecture.

Xilinx adopted the developments of adaptive and flexible processing platforms that submit super powerful new products depending on a variety of technologies, starting from endpoint and ending at the cloud.

Thus, the candidate hardware platforms that will be dependant in this paper is the Xilinx FPGA processors. So, the data base of the intelligent expert system includes mainly the available Xilinx products with main features of each product. The stored features of each processor include mainly the amount of logic cells, block RAM, and the number of IO pins. The main factor that relate directly to huge data processing is the number of IO pins. Table I presents the most important features of some Xilinx products.

TABLE I. RESOURCES OF XILINX PROCESSORS

Max. Capability	**Spartan-7**	**Artix-7**	**Kintex-7**	**Virtex-7**
Logic Cells	102 k	215 k	478 k	1,955 k
Block RAM	4.2 Mb	13 Mb	34 Mb	68 Mb
DSP Slices	160	740	1,920	3,600
DSP performance	176 GMAC/s	929 GMAC/s	2845 GMAC/s	5335 GMAC/s
Transceiver Speed	----	6.6 Gb/s	12.5 Gb/s	28.05 Gb/s
External memory support	800 Mb/s	1,066 Mb/s	1,866 Mb/s	1,866 Mb/s
I/O pins	400	500	500	1,200

Each Xilinx product version has a series that includes different FPGA processors which characterized by their special feature. Each processer has its family members that differ highly through many processing characterization such as system gates, block RAM, user I/O and many other factors that determine the processor's capability. As an example the Spartan FPGA processor with Xilinx products include a series start at Spartan 1 and continue to Spartan 7, and each one of them includes an extended series, each processor of any series has its family members that specialized by its different features from others. So, as an example one of the Spartan 3 series is Spartan 3E that has five family members as shown in Table II [5].

ISSN NO: 2770-4491
Date of Publication:24-08-2022

TABLE II. SPARTAN 3E FAMILY SERIES.

	System Gates	Logic cells	Bloc RAM	User I/O
XC3S100E	100K	2160	72K	108
XC3S250E	250K	5508	216K	172
XC3S500E	500K	10476	360K	232
XC3S1200E	1200K	19512	504K	304
XC3S1600E	1600K	33192	648K	376

Thus the expert system data base contains so many branched tables for each Xilinx processor series and family members to enable the expert system to choose the most suitable FPGA processor for any specific required applications.

Hence, huge data processing application systems to be designed and implemented using the more suitable Xilinx product need the aid of the expert system which has a large data base about the special features of the Xilinx FPGA processors as well as some important and efficient digital techniques that can be dependant in such situations.

Fig. 5 presents different industrial applications which can be configured and formulated with every Xilinx processors groups depending on their potential efficiency, operation conditions, available component, design resources and processor utilities.

Spartan 6 « 1. Industrial Networking 2. Motor control
Artix 7 1. Motor control 2. I/O Module
Kintex 7 1. computational 2. DSP applications 3. High band width application, video, data processing, and transmission.
Zynq-7000 1. Video surveillance 2. Machine vision 3. Motion control platforms

Fig. 5. Industrial applications for different Xilinx processors.

Fig. 6 illustrates the application bands which prefer to be designed and implemented with specific products more than others. The available resources for different Xilinx product of FPGA processors shown in Table I emphasizes the processors - applications correspondence.

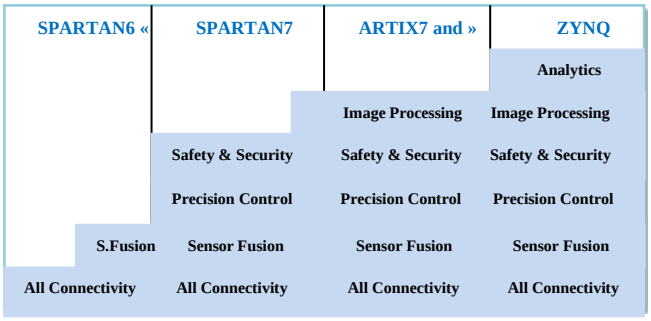

Fig. 6. Applications cost optimized for Xilinx processors.

ISSN NO: 2770-4491
Date of Publication:24-08-2022

V. CASES STUDY

A. Case 1: Median Filter Design

The image was taken here as an example of huge data. So, if the processed image of size 200x200 is taken then the hole input data become 40000 data samples (this number for input data only without any consideration to the output requirements) and this number of input data samples is huge and out of FPGA I/O pins limits. Thus the Xilinx products cannot cover all these data simultaneously which leads to lock the processing scheduling while forcing the processor to stay in its state without proceeding any further. The chosen application in this case is image processing. The problem of huge data is solved by using image segmentation techniques that are designed using Xilinx block sets. Here the black box block is used to take a small window of the processed image which is stored in a RAM , as shown in Fig. 7 and complete the processing on this window before entering the new segment of the processed image.

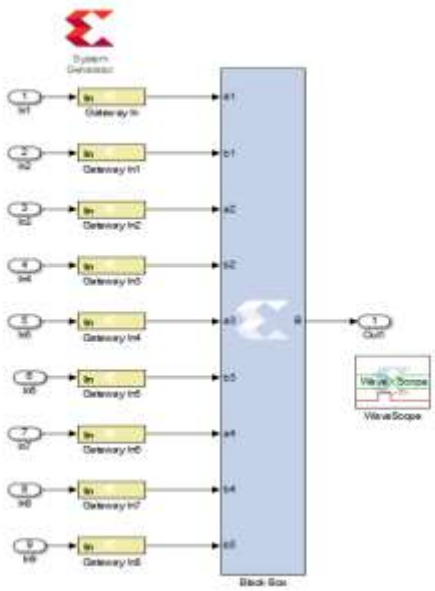

Fig. 7. Implementation of Median filter algorithm by writing the complete VHDL code inside one Black box [1].

Another median filer design by using XSG hardware design for the sorting circuit is shown in Fig. 8. The black box in this design acts as a digital comparator that accept two inputs only. The segmented data window enter simultaneously to all comparators at the same time.

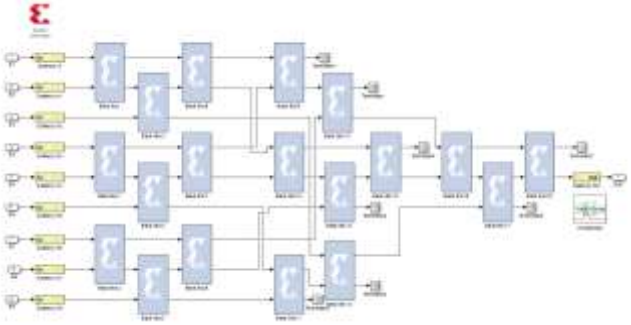

Fig. 8. Systolic Median filter design using XSG [6].

The decision of Expert System is shown in Table III, and the hardware implementation results of the median filter design with different Xilinx products is summarized in Table IV.

TABLE III. EXPERT SYSTEM DECISION

ISSN NO: 2770-4491
Date of Publication:24-08-2022

Recommended	Technique	Suitable Processor
Expert decision	Segmentation	Spartan

	Spartan 3	Spartan 6	Virtex 6
No. of Occupied Slices	344 (5%)	185 (1%)	46 (< 1%)
No. of Slice LUTs	656 (5%)	381 (3%)	728 (1%)
No. of Bonded IOB	81 (47%)	81 (14%)	81 (6%)

TABLE IV. RESOURCES OF MEDIAN FILTER DESIGN

B. Case 2: Turbo Coder

The basic turbo codes should have two parallel convolutional codes with one interleaver that is called a turbo interleaver. For high speed, the parallel codes can be processed simultaneously. A simple structure of turbo code is shown in Figure 9. The first encoder applies first parity bits sequence, and its interleaved sequence is encoded to second parity bits sequence by the second encoder [7].

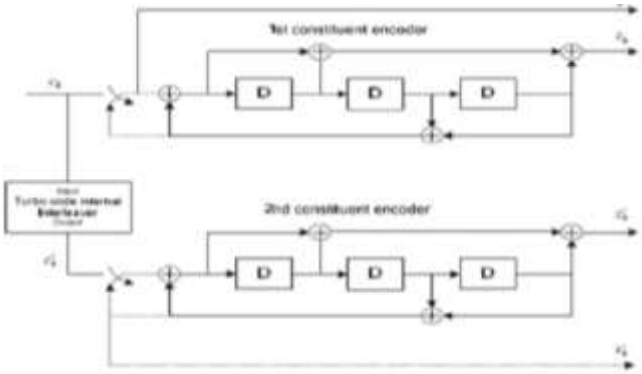

Fig. 9. Block diagram of Turbo Coder. (the figure is changed)

The output of the expert system ensure that Artix processor is the most suitable for FPGA based turbo encoder design among other Xilinx products.

TABLE V. EXPERT SYSTEM DECISION

Recommended	Technique	Processor
Expert decision	Serial entering	Artix

The amount of used resources with various FPGA processors, to implement the turbo encoder, is summarized in Table VI.

TABLE VI. RESOURCES OF TURBO CODER DESIGN

Resources	Artix-7	Spartan-7	Kintex-7	Virtex-7
Number of Slice LUTs.	**25** (1%)	**23** (1%)	**27** (1%)	**29** (<1%)

ISSN NO: 2770-4491
Date of Publication:24-08-2022

Number of Slice Registers	**52** (1%)	**47 (1%)**	**53 (1%)**	**42** (0%)
Number of Bonded IOB	**83** (13%)	**83** (20%)	**83** (16%)	**83** (6%)

The turbo decoder needs more design components and processor resources, therefore Artix and Spartan processors are excluded. Table VII presents the required resources for decoder design with higher level Xilinx processors.

TABLE VII. RESOURCES OF TURBO DECODER DESIGN

	Virtex-4	**Virtex-5**	**Kintex-7**
No. of Slice LUTs	4.857(39%)	4.197(21%)	6.131 (3%)
No. of Slice Registers	4.875(39%)	4.911(21%)	153 (1%)
BIOs	11 (5%)	11 (4%)	11 (2%)

Special case can occur when implementing the turbo decoder with FPGA processors, that the required resources may exceed the availability of the recommended processor [8]. Table VIII shows the required design components for the turbo decoder using the Virtex 7 Xilinx processor.

TABLE VIII. RESOURCES OUT OF THE PROCESSOR AVAILABILITY [8]

Resources	**Used**	**Available**	**Utilization**
No. of LUTs	139258	303600	45%
No. of Bonded IOBs	1200	700	**184%**
No. of DSP Slices	456	2800	16%

Approximately all the over range cases, the design requirements of which exceed the available resources of a specific processors, relate to the IOBs resource which is limited for all processors versions. I/O cannot be made free in number or over a specified allowed limit with any processor manufacturing technology.

C. Case 3:Adaptive Noise Cancelling System

Adaptive noise cancellation (ANC) is used with non-stationary environment. The heart of Adaptive noise canceller is the adaptive filter that cancels the interference noise from the desired signal by continuously varying its weights. Different optimized algorithms can be used to adapt filter performance. The LMS Algorithm is usually used to update the weights of the adaptive filter. The main reason behind choosing the LMS algorithm is due to its simplicity and high performance efficiency [9]. A simple block diagram of an adaptive noise cancelling system is shown in Fig. 10.

ISSN NO: 2770-4491
Date of Publication:24-08-2022

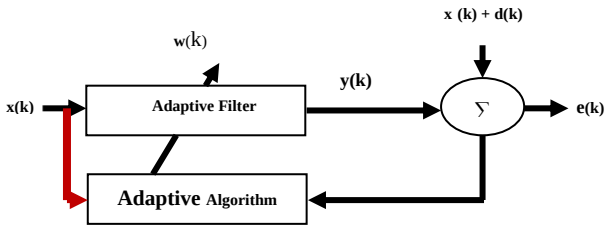

Fig. 10. Adaptive noise cancelling system.

In this case study the LMS algorithm was implemented using different FPGA processors to determine the difference between the used resources for each processor and then it can decide the most suitable one.

The expert system gives the decision shown in Table IX, for FPGA based implementation of the ANC system.

TABLE IX. EXPERT SYSTEM DECISION

Recommended	**Technique**	**Suitable Processor**
Expert decision	Partitioning	Spartan

The resources used with the FPGA design for the LMS algorithm and ANC system with different processors was shown in Table X.

It is clear from data in Table X that the Spartan processor is the preferable among the others due to utilization ratios which indicate efficient utilization of the processor resources.

TABLE X. RESOURCES OF ANC SYSTEM

Resources	**Spartan 3E**	**Virtex II**	**Virtex 4**
Number of Occupied Slice	2010 (43%)	2395 (22%)	2160 (14%)
Number of Slice Flip Flops	1913 (20%)	2942 (13%)	1913 (6%)
Number of 4 input LUTs	3266 (35%)	2172 (10%)	3265 (10%)
Number of bounded IOBs	169 (72%)	169 (43%)	169 (37%)

VI. DISCUSSION AND ANALYSIS

As we mentioned earlier one of the main problems in FPGA based designs is the limited number of input and output pins because it represents the basis for achieving the design, otherwise what is the benefit of the design if I cannot enter the data and then extract the results to work with and implement them.

ISSN NO: 2770-4491
Date of Publication:24-08-2022

The best way to study and analyze the results is by presenting them with charts that save a lot of researcher's effort. In this section the results of the three case studies were plotted in the following figures.

Figure 11, Figure 12, Figure 13, and Figure 14 are for median filter design, turbo encoder design, turbo decoder design, and adaptive noise canceller system design respectively. These graphs are plotted for one of the design factors which is IOBs, that represents the number of input and output pins. Here the ratios of the needed IOBs with respect to the total available are taken. The dependent on the ratios is very important in expert system design. Thus, when the percentage is high, this indicates that the processor used is very suitable to implement the required design.

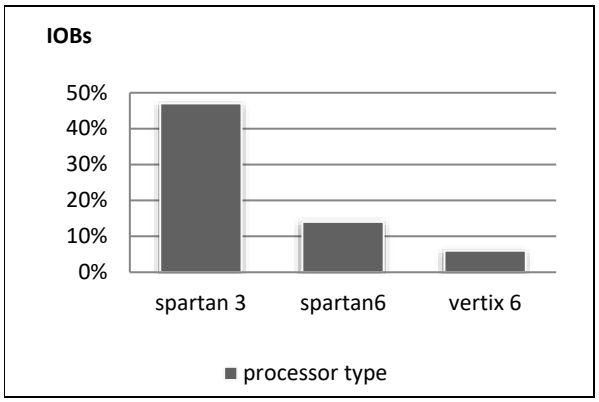

Fig. 11. Percentage of used IOBs for median filter design with various FPGA processors.

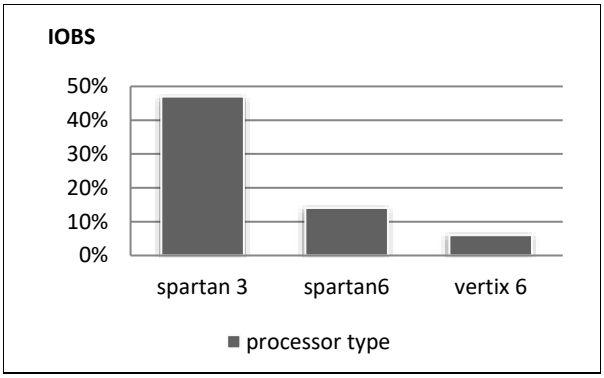

Fig. 12. Percentage of used IOBs for turbo encoder design with various FPGA processors.

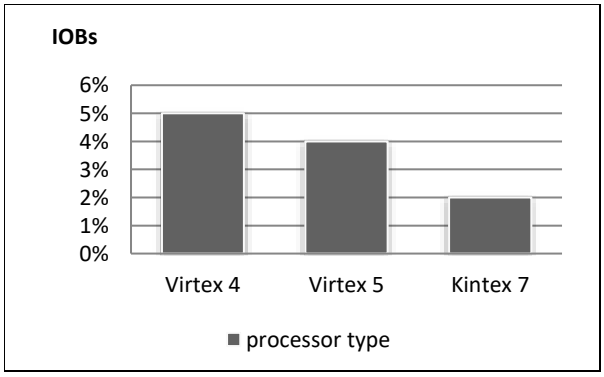

Fig. 13. Percentage of used IOBs for turbo decoder design with various FPGA processors.

ISSN NO: 2770-4491
Date of Publication:24-08-2022

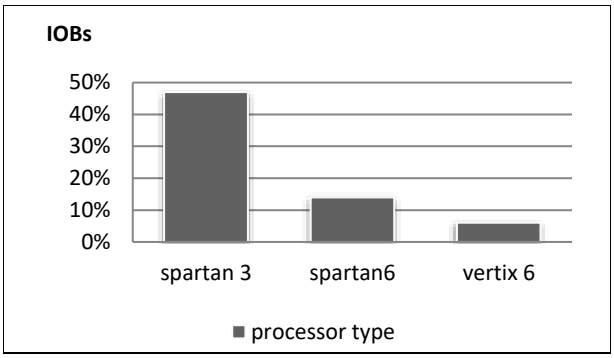

Fig. 14. Percentage of used IOBs for ANC design with various FPGA processors.

From Figure 11, it can be seen that the Spartan family is preferable for 3*3 median filter design, precisely Spartan 3. Spartan 6 is far away to be a suitable processor used to spatial filter design of order three since the required input and output pins is somewhat few in number. If other filter designs are needed and if the filter order is high then the modern versions of Spartan products may be required.

Figure 12 shows that the preferred processor for encoder design is Spartan , while the decoder needs at least Vertix processor as it is clearly viewed in Figure 12. This is due to known axiom that states the turbo decoder design is more and more complex than its encoder design. An overview to the chart of Figure 13 show us that the Xilinx processor generations seem to be approximately the same for turbo encoder design from point of view that take only IOBs in its considerations, since most of the newest Xilinx products have ranges of input and output pins that are close to each others.

Once again Spartan processor is better than Virtex processor for adaptive noise canceller (ANC) design to implement the least mean square (LMS) algorithm, and Spartan 3E is more suitable than Spartan 6 as shown in Figure 14.

VII. CONCLUSIONS.

FPGA based design and implementation of different digital systems have been presented in this paper, for various real time applications that depend on Xilinx processors in their implementations. The intelligent system can determine the most suitable technique for data processing of high density, depending on expert knowledge, based on an extended data base of digital techniques and algorithms for data segmentation and partitioning. FPGA based digital system design using different Xilinx processors and products, also requires experience about these processors and their special features and capabilities while determining the available resources for each processor. So, the expert system with its unique and smart data base, can give a most suitable decision to choose the efficient processing technique for a specific application. The second task of the expert system is to determine the most suitable Xilinx processor for which its available resources and design components match the application requirements without any loss for the processor components and abilities. Different case studies have been tested that cover one dimensional as well as multidimensional real time applications. The case studies chosen, that tested for design and implementation with different FPGA based processors are a spatial filter for image processing , turbo coder, and an adaptive noise cancelling system with speech signals. The expert system shows that Spartan 3 Xilinx processor is the most suitable for median filter design. This decision confirms the hardware implementation testing, that shows the highest exploitation of processor resources appears with the Spartan 3 which takes a percentage of resources utilization of 5%, 5%, and 47% for Occupied Slices, LUTs and Bonded BIOs respectively. For turbo encoder design all processors seem to be the same, although the expert system shows that the most suitable processor for this application is the Artix. For turbo decoder the situation is different, since its hardware design requires a large

ISSN NO: 2770-4491
Date of Publication:24-08-2022

amount of logic components. Thus, Artix and Spartan are excluded as candidates. Virtex 4 is the more preferable processor for the this decoder design.

A very important note must be taken into consideration for the FPGAs designer of real time digital systems, namely that the design utilization may exceed the available resources of specific processors and if the upgraded version of the FPGA's processors is dependant, then the processor utilization efficiency will be lost. Another important note, that the percentage utilization of bonded I/Os is always has the highest value among the others, and this is exactly the problem facing huge data systems.

As a final conclusion, professional expert system can provide the perfect decision to choose the optimal processor for a specific application to get full benefits of the processor's available components and design resources. This enables the chosen processor to operate with approximately its full utilities. As well as the, expert system can also determine the most suitable technique, which should be relied upon, to safe the processing from any disturbance while ensuring the achievement of the essential required demands.

REFERENCES

[1] A. H. Rasheed, "FPGA Design and Implementation of Huge Data Systems", IJCA, Volume 174 – No.7, September 2017

[2] R. J. Petersen and B. L. Hutchings, "An assessment of the suitability of FPGA-based systems for use in digital signal processing," *Conference on* Future Generation Computer Systems, vol. 22, pp. 67–79, Jan 2008.

[3] Ownby,M., Mahmoud,W.H., "A design methodology for implementing DSP with Xilinx System Generator for Matlab," IEEE International Symposium on System Theory, pp.404-408, 2003

[4] N. Vani, N. Usha rani "FPGA implementation of image filtering applications using XGS", International Conference on Navigational Systems & Signal Processing Applications, December 13th &14th, 2013.

[5] Xilinx, Sparten 3 FPGA Family: Complete Data Sheet. www.xilinx.com.

[6] A. H. Rasheed, "FPGA-based Optimized Systolic Design for Median Filtering Algorithms", IJAER, Vol. 12, No. 24 (2017) pp. 16100-16113.

[7] Vadim Belov, Sergey Mosin, " FPGA Implementation of LTE Turbo Decoder Using MAX- log MAP Algorithm", in 2017 IEEE Embedded Computing.

[8] G. Santosh, S. Rajaram , "Design And Implementation Of Turbo Coder For LTE on FPGA", IJESS, Vol. 4 : Iss. 1 , Article 7.

[9] S. Gaur, S. Sharma, V.K Gupta, "Implementation of ANC System Using Xilinx System Generator (Co-hardware Simulation using Vertex 6 FPGA Kit)" 2015 IJEDR, Volume 3, Issue 2, ISSN: 2321-9939.

[10] Xilinx, Using Block RAM in Spartan-3 FPGAs, Dual-Port Block Memory Core v6.3,. www.xilinx.com.

[11] B. Stellato, T. Geyer, and P. J. Goulart, "High-Speed Finite Control Set Model Predictive Control for Power Electronics," *IEEE Trans. Power Electron.*, vol. 32, pp. 4007-4020, 2017.

[12] P. M. Sanchez, O. Machado, E. J. B. Peña, F. J. Rodriguez, and F. J. Meca, "FPGA-based implementation of a predictive current controller for power converters," *IEEE Trans. Ind. Informat.*, vol. 9, no. 3, pp. 1312-1321, Aug. 2013.