ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 49-53 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

DIGITAL IMAGE NOISE FILTERING USING EVOLVABLE HARDWARE

T.Jenny[1], T.Geetamma[2]

[1]M-Tech., Student, Dept.of ECE,GMRIT,Rajam,JNTUK,India,jnyjms@gmail.com

[2]Asst. professor, Dept.of ECE,GMRIT,Rajam,JNTUK,India,tgeetamma@yahoo.co.in

Abstract*:* Images obtained through modern cameras may be infected by a range of noise sources, it may be the photon or a chip electronic noise and also by the distortion such as shading or improper lighting. The system should adapt to these changing environment autonomously. For example, to the changes of illumination or after replacement of a damaged camera, for this automatic design, a new technique is described, i.e. Evolvable Hardware (EHW) which changes hardware structure in order to adapt to the environment in which it is fixed. The communications between the evolutionary algorithm and electronics give rise to the field "evolvable hardware". Evolvable hardware architectures that can evolve autonomously straight on the reconfigurable fabric of FPGAs, using Dynamic partial recofiguration.This DPR creates a very high level of flexibility and it helps to recognize the adaptive hardware algorithms.

Key words: Evolvable hardware, self-adaptive systems, Reconfigurable hardware, Adaptable architecture, Autonomous systems.

INTRODUCTION

Embedded systems engineering is nowadays facing a vast challenge resulting from the ever-increasing demand for the highly adaptable electronic devices, high performance, increasing complexity, flexibility, low power, and reprogram ability. And these products should reach the market at a low time and should achieve high challenging life. These severe requirements, which resist each other, have caused a rapid increase in the underlying complexity of embedded systems. Since among other concerns, the need for systems to adapt to a very different operating condition throughout its lifetimes occur.

Besides, without any human intervention the adaptation have to be done. In the last years the system design is declared by reason of its product strength by using and then mixing of different IP cores. This helps in reducing complexity, low time to market and also offers low power as well as high performance. But for this self adaptation a plenty of technologies and methods should be needed [1].

The partial reconfiguration helps the designer to shrink the size of the design vigorously. In the earlier period, for making changes in a required field, the designer had to stop the entire device. But now by the usage of DPR, the designer needs to robustly change just a portion of the system without stopping the entire system.

Evolvable hardware use the evolutionary algorithm to meet their design requirements, and also it manipulates a population of individuals, where the individual describes how to build a candidate circuit and for every circuit, fitness is assigned, which indicates how well a candidate circuit can satisfy the design conditions. The development of logic circuits has been carried out by a superficial evolvable system which uses $(1 + \lambda)$ evolution strategy as the hub of the evolution [2].EHW was proposed by higuchi in 1993 as the automatic design of hardware using an EA. It is an electronic hardware system that evolves to verify circuit using an FPGA and other reconfigurable modules, and also it arises as a competent solution for automatic digital combination of digital and analog circuits. During the last decade, a special interest has been paying attention on evolving digital systems by straight mapping a Chromosome on the FPGA design bit stream [3].This approach allowed a great degree of suppleness for evolving circuits.

PROPOSED ARCHITECHTURE

Partial reconfiguration is the procedure of altering a part of reconfigurable hardware circuit while the further part of the device is still operating. By using the partial reconfiguration one or several subcomponents can interchange while the FPGA is still running. The partial reconfiguration cannot support all types of FPGA.

Partial reconfigurations can be divided into two parts. They are:

- Dynamic partial reconfiguration
- Static Partial reconfiguration

Dynamic partial Reconfiguration

Since 1980s the FPGA market is growing quickly with wide ranging of applications in different industries. For that the FPGA manufactures keep updating their FPGA with the most advanced stage. In the recent days a new technology is introduced, called Dynamic partial recofiguration.This Dynamic partial Reconfiguration supply a way to adapt the implemented logic in FPGA where still the device is running and new design spaces which are having more benefits are offered through DPR, those benefits can reduce the design time as well as they can save the memory as the reconfiguration files i.e., bit streams [4].The dynamic partial reconfigurations is an

ISSN 2278-3091

advanced technique, that is carried in the device to allow the FPGA for adapting the changes in hardware algorithms, for improving the fault tolerance and for utilizing the resource properly as well as it can also reduces the power expending.

Block Diagram

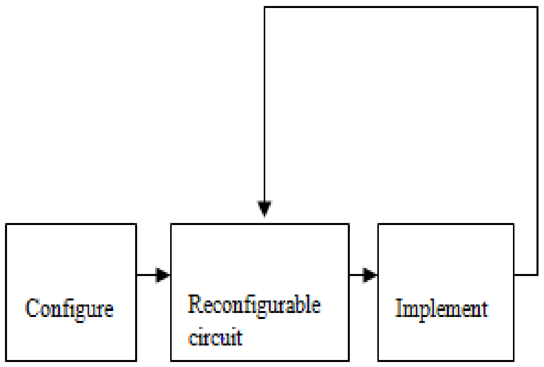


Fig 1: Dynamic partial reconfiguration

The DPR is also well-known as active partial reconfiguration.

The Fig 1 shows the Dynamic partial reconfiguration, as it permits to modify a part of the device while the rest of an FPGA is still working. But in the static partial reconfiguration, the device is not in dynamic during the reconfiguration process.i.e, while the data is sent into the FPGA, the device is stopped up, and will start working after the configuration is accomplished.

Dynamic partial reconfiguration architecture

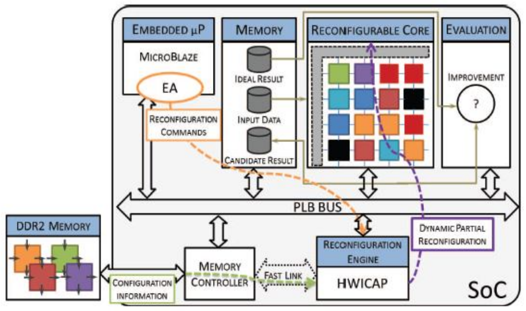


Fig 2: Overview of the DPR system architecture

The work introduced is based on Dynamic partial reconfiguration engine. Contemporary FPGAs with Dynamic partial reconfiguration features allow the implementation of complex, flexible, hardware systems, real adaptive systems. A highly regular and modular architecture integrated with a fast reconfiguration method is introduced in this paper, by allowing the DPR in the evolvable circuit. The evolvable processing IP cores can be inbuilt by providing challenging data, processing capabilities and as well as by the delay overheads.

Different IP cores are located on the system on chip. These different IP cores which are adaptive are connected through a common BUS.

The fig 2 shows the overview of the DPR system architecture. The main two components in the DPR architecture are the reconfigurable core and the reconfiguration engine. The reconfigurable core is reconfigured by the reconfiguration engine during the adaptation process by making use of DPR. The adaptation engine works as an evolutionary algorithm in the DPR.The adaptation in the DPR is achieved by the amalgamation of the reconfiguration engine with the hardware ICAP.The embedded microblaze issues the required reconfiguration instructions to configure the reconfigurable core by the evolutionary algorithm as shown in the figure 2.The reconfigurable core consists of processing elements and the HWICAP(Hardware internal core access port) is linked to DDR2 memory by fast link, this fast link helps the data to drive fast into the DDR2 memory.

Reconfigurable architecture

The RC architecture is a highly regular and parallel 2D mesh type array. The figure 3 shows the reconfigurable architecture. The processing elements which are originated inside the reconfigurable core are of systolic structure, where the processing units are arranged in an array of pipe network where internodes connectivity are fixed and limited to the four closest neighbors (i.e., east, west, north, south) as shown in the figure 3 and for these PEs instructions are given by the microblaze. The processing matrix is generating an array of 4×4 processing element.

The input to the array is a moving window sized 3×3 processing elements. The best path is not yet declared from the window to the input PEs and the output is evolved at the least [5].

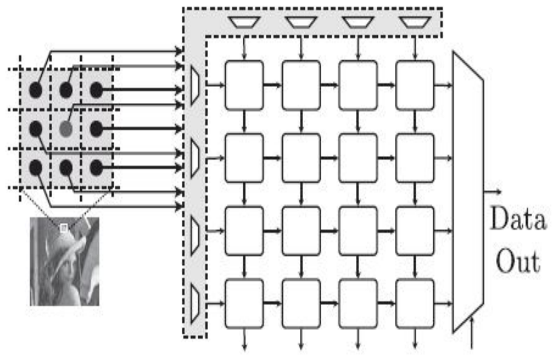


Fig 3: Reconfigurable architecture

Within the RC architecture, each processing element can be dynamically configured to have special functionality and input mapping.

ISSN 2278-3091

PROCESSING ELEMENTS

A processing element is a standard expression used to refer a hardware component that executes a brook of commands. Each processing element contains a data register file and three computation units, such as, an arithmetic/logic unit (ALU), a multiplier and a shifter. Computational directions for these elements comprise both fixed-point and floating-point operations. And each computational instruction can perform in a single clock cycle.

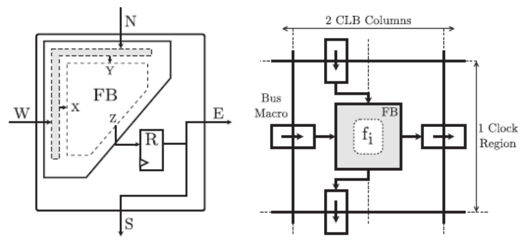
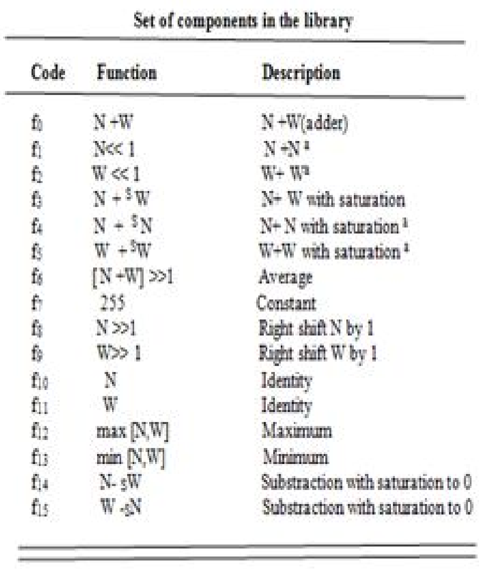


Fig 4: (a) Conceptual view (b) Reconfiguration view

Figure 4(a) shows the conceptual point of view, where each block consists of a functional block as well as a register and each combination are integrated and stored as an autonomous module in the PE library [6].The Figure 4(b) shows the reconfiguration point of view. Here the adaptation is achieved by directly configuring the necessary PE in each location of the array by taking the required data that is stored inside the library as shown in the table 1.The reconfiguration time is kept low because low mutation rates in the evolutionary algorithm generate a small amount of PEs to reconfigure.

TABLE 1



Set of components in the library

Code	Function	Description
f_0	N +W	N +W(adder)
f_1	N<< 1	N +N [a]
f_2	W << 1	W+ W[a]
f_3	N + ^{S}W	N+ W with saturation
f_4	N + ^{S}N	N+ N with saturation [a]
f_5	W + ^{S}W	W+W with saturation [a]
f_6	[N +W] >>1	Average
f_7	255	Constant
f_8	N >>1	Right shift N by 1
f_9	W>> 1	Right shift W by 1
f_{10}	N	Identity
f_{11}	W	Identity
f_{12}	max [N,W]	Maximum
f_{13}	min [N,W]	Minimum
f_{14}	N- $_S$W	Substraction with saturation to 0
f_{15}	W -$_S$N	Substraction with saturation to 0

MODIFIED ARCHITECHTURE

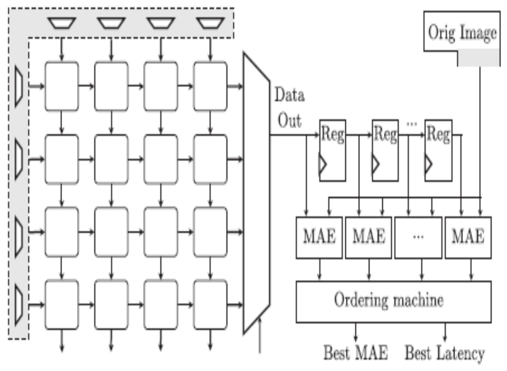


Fig 5: Modified architecture for calculating circuit latency

The figure 5 shows the modified architecture for calculating circuit latency, as explained in Dynamic partial reconfiguration architecture where the processing elements are originated inside the RC , they have the fixed internodes connectivity, so that it can define the static circuit latency. But when a PE is reconfigured, then the internodes connectivity also changes according to the input. Besides, the output multiplexer controlled by the EA makes it impossible to predict the selected output PE. So for that reason there is no other logical way to determine the circuit latency. To achieve the circuit latency a modified architecture is designed as shown in the fig 5.

The new architecture consists of a number of MAE calculations so that we can calculate the best circuit latency. The input to the filtered pixels is sampled in a special clock cycle by inserting a simple shift register. The fitness of an individual offspring is tested at a once.

EVOLUTIONARY ALGORITHM

Evolvable hardware (EHW) techniques were introduced by H.de Garis. The evolvable hardware are carried out by the evolutionary alogorithms.These algorithms can reconfigure themselves without any human intervention, but the main disadvantage is to choose the correct evolutionary algorithm for the EHW. Such as the fitness function, chromosome representation, population size and genetic operators. Here the adaptation is motivated by the (1+**λ**) evolutionary strategy, here 1 is the parent and **λ** is the progeny.

The population is chosen by a random way and by using the selection operator the individuals are selected for the best fittest and each individual is calculated, then the best fittest is selected, by the mutation Operator the genes are mutated and the new population are created as shown in the figure 6.

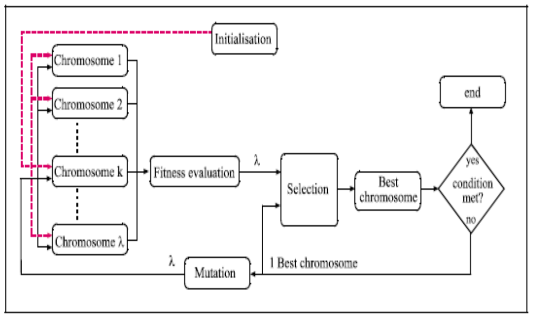


Fig 6: (1+**λ**) evolutionary strategy

The figure 6 shows the evolution strategy .Here the fitness function is calculated by the mean absolute error (MAE).there is three steps to find out the fitness function of a processing matrix. They are, firstly the processing matrix is reconfigured with an applicant circuit, next the obtained image is filtered and then finally the fitness value is calculated.

$$\text{MAE} = \frac{1}{RC}\sum_{c=0}^{R-1}\sum_{c=0}^{C-1}⟦[\text{I}(\text{r,c}) - \text{K}(\text{r,c})]⟧ \qquad (1)$$

As shown in the equation (1), the R represents the row and C represents the column. The I represent the original image as well as the K represent the filtered image.

WAVELET DOMAIN FILTERING

The colour images are converted into a wavelet domain transformation for sinking the noise as well as it is also reduced by introducing the mean and median filters. Wavelet transforms are multi resolution decompositions that can be used to analyze signals and images. They describe a signal by the power at each scale and position. Edges can be located very effectively in the wavelet transform domain. By this method the noise in the images are reduced more than 80%, and most of the noise are reduced at the edges.

Mean filter

The mean filter is a simple sliding window spatial filter that replaces the center value in the window with the average (mean) of all the pixel values in the window. The window, or kernel, is usually square but can be any shape. The image can be getting smoother than before and even the effect of the dead pixels will be reducing gradually. But a lot of data will be lost in the image than before as well as the edges also be blurred.

Median filter

The median filter is also a sliding window spatial filter, but it replaces the center value in the window with the median of all the pixel values in the window. As for the Mean filter, the kernel is usually square but can be any shape. Same as to the mean filter the dead pixels reduces gradually and the image is less blurred as well as the brighter background gets sharper than before

RESULTS

Noise image

Distributed image

Dithered image

Recovery image

CONCLUSION

Hence we have observed that by using the dynamic partial reconfiguration the self adaptive and self reconfiguration are achieved autonomously on the fabric of FPGA with the help of evolvable hardware. And also by using the wavelet domain filtering the colour images are converted into wavelet domain transformation as well as the noise in the image is also reduced by using the filters like Mean and Median filters.

REFERENCES

[1] Ruben Salvador, Andres otero,Javier Mora,Eduardode la Torre, Teresa Riesgo and Lukas "Self-Reconfigurable Evolvable Hardware System for Adaptive Image Processing" TRANSACTIONS ON COMPUTERS ,VOL.62, NO.8, AUGUST 2013.

[2] Emanuele stomeo, Tatiana kalganova, and cyrille Lambert,"Analysis of Genotype Size for an Evolvable Hardware System", international journal of electrical robotics, electronics and communication engineering vol: 1, 2007.

[3] Andres Upegui and Eduardo Sanchez "Evolving hardware by dynamically reconfiguring Xilinx FPGA", Lecture Notes in computer science VOL.3637, 2005, PP 56-65

[4] Wang Lie, Wu Feng-Yan"dynamic partial reconfiguration in FPGA"Third international symposium on intelligent information technology application.

[5] Lukas sekanaina"Image filter design with evolvable hardware"Evo workshops LNCS 2279, PP.255-266, 2002.

[6] Manisha, P.Khorgade, Shweta Hajare, P.K Dakhole "Structural level designining of processing elements using VHDL", International Journal of Soft Computing and Engineering.Issue-2, May 2014.