International Journal of Advance Research in Engineering, Science & Technology(IJAREST), ISSN(O):2393-9877, ISSN(P): 2394-2444,
Volume 2, Issue 4, April- 2015

FPGA Technology Based Video Watermarking

Priyanka R. Raval
Computer Science, Post Graduate Department
Institute of Technology, Nirma University
Ahmedabad, India
priyankaraval2003@gmail.com

ABSTRACT— *With the availability of web cameras and high resolution mobile phones video contents are easy to be created. Wide usage of video content demands effective provision of content ownership and copyright protection.*
Effective video processing software should process 24 to 30 frames per second and deal with the complexity of video watermarking algorithm to produce outcome with a reliable speed. As FPGAs have grown in capacity, improved in performance and decreased in cost they have become a viable solution for performing computationally intensive task though this involves intensive research on the hardware implementation of video watermarking algorithm.
This paper is mainly based on implementing video watermarking algorithms using FPGA technology and come up with an effective embedded solution for video ownership.

KEY WORDS: ***FPGA, Video Watermarking, Xilinx Applications, Watermarking.***

INTRODUCTION

Multimedia applications are increasingly becoming popular nowadays. Sharing of video information over networks and internet is increasing day by day. To provide identity and copyrights to these applications is becoming basic need for multimedia users.
Video files being important part of multimedia, video watermarking is becoming a basic need to provide ownership and copyright to video files. Some of the important characteristics of videos that impact watermarking like, 'high spatial correlation between successive frames' and 'embedding the same watermark in all frames' is insecure etc makes video water marking a challenging task.
To overcome the challenges in video watermarking and on the parallel not loosing speed and time boundaries of a system are main motivation for an external device which can perform video watermarking.
FPGA technology is having benefits like reduced inventory costs, easy prototyping etc over other technologies like GPP and ASIC. This project utilizes benefits of FPGA technology and attempts to develop an external embedded solution for video watermarking requirements.

EVALUATION CRITERIA FOR WATERMARK STRATEGY

To evaluate different watermarking strategies, some criteria are defined in the following (Ziener and Teich 2005):

1. Functional correctness: This is the most important criteria. If the watermark process destroys the functional correctness, it is useless to distribute the core.
2. Resources overhead: Many watermark algorithms need some extra resources. Some for the watermark itself, some because of the degradation of the optimization results from the design tools and language. The lower the level of the HDL, the more detail the developer must master (Jain-Yuan-Pari and Qu 2003).
3. Transparency: The watermark procedure should be transparent to the design tools. It should be easy to integrate the watermarking step into the design without altering the common design tools.
4. Verifiability: The watermark should be embedded in such a way that simplifies the verification of the authorship. It should be possible to read out the watermark only with the given product without any further information from the design, which must be ordered from the accused company (Latha-pillai and Sheela 2005).
5. Difficult to remove: The watermark should be resistant against removal. The effort to remove the watermark should be greater than an effort needed to develop a new core or removal of watermark should cause corruptness of the functionality of the core. Watermarks which are embedded into the function of the core are more robust against removal than additive watermarks (Wolf 2005).
6. Strong proof of authorship: The watermark should identify the author with a strong proof. It should be

impossible that other persons can claim the ownership of the core. The watermark procedure must be resistant against tampering.

OVERVIEW OF WATERMARKING ATTACKS

The designer has to think of the attacks to the watermarking techniques to provide the robustness to the design. Ownership Deadlock, Counterfeit Ownership and Forged Ownership are the possible threats to a watermarking design. Based on these threats, the attacks can be categorized as, Ambiguity attack, Removal attack, Copy attack or Key copy attack. These all attacks can break the service provided by the watermark

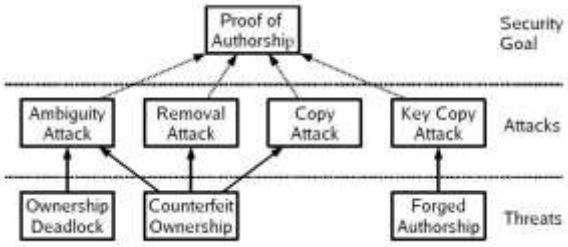

Fig. 1 Overview of attacks

FPGA INTRODUCTION

An FPGA consists of a matrix of logic blocks that are connected by a switching network. The logic blocks and the switching network both are reprogrammable. This allows application specific hardware to be constructed and allows changing the functionality of the system with ease. FPGA is a silicon chip with unconnected logic gates. It is an integrated circuit containing many (64 to over 10,000) (Brown - Jonathan 1996) identical logic cells that can be viewed as standard components. The individual cells are interconnected by a matrix of wires and programmable switches.

Field Programmable means that the FPGA's function is defined by a user's program rather than by the manufacturer of the device. Depending on the particular device, the program is either 'burned' in permanently or semi permanently as part of a board assembly process, or is loaded from an external memory each time the device is powered up.

A. FPGA: Basic Architecture

The FPGA has three major configurable elements:

- Configurable logic blocks(CLBs)
- Input-output blocks(IOB)
- Interconnects

Each CLB contains a logic element which is implemented as a lookup table (See Fig. 3). This logic element operates on four one-bit inputs and outputs single data bit. Using CLB any Boolean function of four inputs can be performed.

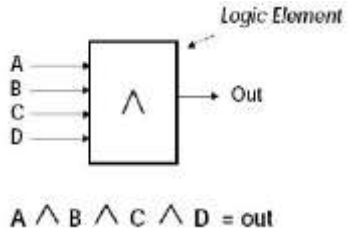

Fig. 3 Basic CLB architecture (Brown - Jonathan 1996)

The Configurable Logic Block is the basic logic unit in an FPGA. Exact numbers of CLBs and features of the CLB vary from device to device. Every CLB consists of a configurable switch matrix with 4 or 6 inputs, some selection circuitry (MUX, etc.), and flip-flops. The switch matrix is highly flexible and can be configured to handle combinatorial logic, shift registers or RAM.
FPGAs provide support for dozens of I/O standards. I/O in FPGAs is grouped in banks (see fig. 2) with each bank independently able to support different I/O standards.

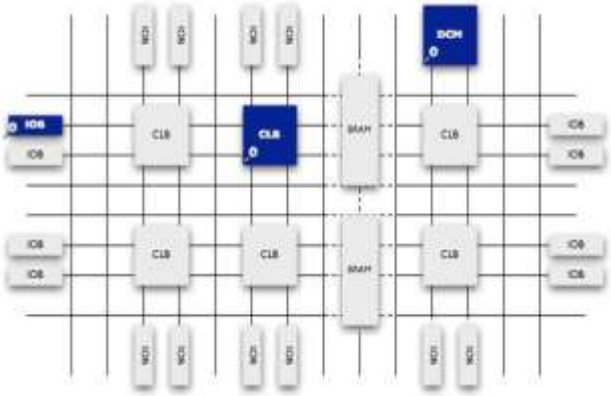

Fig.2 Basic FPGA architecture (Xilins site)

FPGA architecture supports so many kind of interconnects. Like, short wires, general-purpose wire, global interconnects and specialized clock distribution networks. Because wires can introduce a lot of delay, also wiring networks of different length and connectivity need different circuit designs. That is why FPGAs need different types of wires.

In new generation FPGAs, Digital clock management (DCM) is an important feature and is provided by most FPGAs in the industry. The prime concern of DCM is to eliminate the skew and other issues that designers had to face with in designing global signals into FPGAs in the past.

B. Advantages of FPGA Technology over Other Existing Technologies

The role of FPGA in Embedded Systems is gaining importance due to its increasing capabilities and availability of powerful FPGA design software tools. The digital video applications are driving FPGA market and enabling use of FPGA for broad range of applications.
FPGA devices have got advantages over General Purpose Processor and Application Specific Processor to design an embedded system. Table I tries to describe some important points among the three technologies.

Table 1: Advantages of FPGA Technology over GPP and ASIC

GPP	ASIC	FPGA
It can be used in variety of environment	Its usage is specific to	It can be designed specific
Slower, power hungry	Fastest, lower power	Between GPP and ASIC
Such devices are ready to use but general purpose	They take months to be fabricated on manufacturing line	Production slower than GPP because needs to be configured but much more faster than ASIC
Uses much more no of transistors than actually required by the application	Uses lesser transistors	Uses more transistors than ASIC and lesser than GPP

Moreover, below are some application life-cycle specific advantages of FPGAs (Wolf 2005):

- No wait for the final design. The design can be programmed and tested into FPGA immediately.
- FPGA is excellent prototyping vehicle. Because jump from prototype to product is easier.
- They can be used in several different designs reducing inventory costs.
- Performance gains are obtained by bypassing the fetch decode execute overhead of general purpose processors.

Thus, FPGA offers a compromise between the flexibility of general purpose processors and the hardware-based speed of ASICs.

FPGA DESIGN PROCESS

Programming process of FPGAs is having some similarities with the programming process of conventional microprocessors. Many powerful tools exist to program FPGAs. In microprocessors, we can program the memory bits, and in FPGAs even the logical gates are under programmer's control.
Below figure shows you the steps involved in the designing of FPGAs.

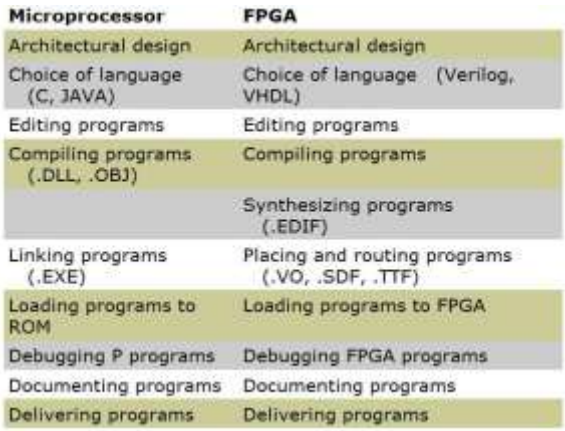

Fig. 4 Design process for microprocessor and FPGAs (Klingman 2004)

Looking to the figure one can compare various steps of FPGA programming with the corresponding stages of program development in a microprocessor. Though the first level picture of FPGA Programming looks similar to the microprocessor programming, the actual programming process is quite different.

The output of Verilog code compilation is RTL netlist. When input to a synthesizer, the Verilog is converted into a gate-level netlist. It is capable of being mapped into FPGA hardware. This gate-level Verilog can be compiled and simulated. So we can debug at the actual gate level.
The simulation of the RTL Verilog is called functional simulation, while the simulation of the synthesizer Verilog output is called gate-level simulation.

In gate-level simulation, synthesizers can optimize FPGA netlists. Area optimization is possible during gate-level simulation which will attempt to use the fewest number of gates (silicon area) on an FPGA at the expense of execution speed. Delay optimization

attempts to maximize the execution speed, even if more FPGA area is required. That is why the functional code written in Verilog at the RTL level may have different implementations.

DESIGNING TOOLS AND TECHNOLOGY BEING USED

A. Hardware Utilized

In this Experiment, Xilinx Spartan 6 - SP605 board is used for having FPGA functionalities. Xilinx provide Xilinx ISE 13.2 Design Suite to operate with SP605 board. This tool set supports HDL, VHDL and Verilog languages.

B. Language Utilized

C, VHDL: VHDL is the basic proprietary hardware design language. It is originally come as simulation languages. Below figure shows a snapshot of Xilinx SP605 Spartan 6 board (Wain-Bush-Guest-Deegan-Kozin and Kitchen 2006).

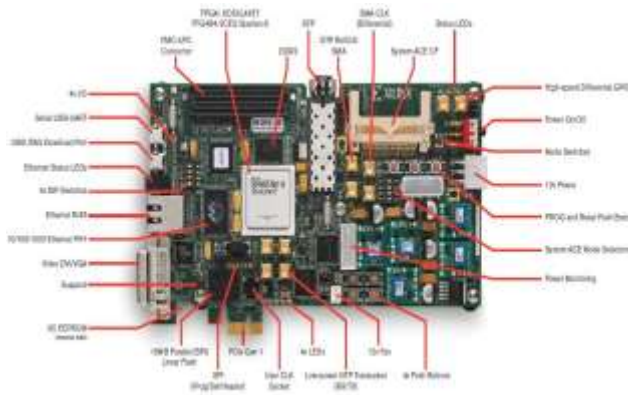

Fig. 5 Spartan-6 FPGA SP605 Board Features (Xilinx Site)

Moreover, the board (Spartan SP605) is being programmed through Xilinx Platform studio (XPS) and Xilinx Software Development kit provided under Xilinx EDK package. Programming in Xilinx Software Development kit is in C language while Xilinx ISE (Integrated Software Environment) supports VHDL. So, C and VHDL is the chosen programming language for this experiment work.

Apart from Xilinx ISE tool, Matlab Simulink tool is also utilized to design the hardware block diagrams of the FPGA system.

IMPLEMENTATION DETAIL

A frequency domain algorithm referred from IEEE paper - 'A DCT Domain Visible Watermarking Technique for Images (Mohanty - Ramakrishnan and Kankanhalli, 1977) is used as an example in this experiment to
Showcase that after evolution, the FPGA technology is quite suitable for video watermarking. This is one of the most referred algorithms for visible image watermarking. Also, this algorithm satisfies the security criteria for visible video watermarking discussed in this paper. Looking to these aspects, the algorithm has been selected as an example for video watermarking experiment.

Below is a block diagram showing overall design of the experiment.

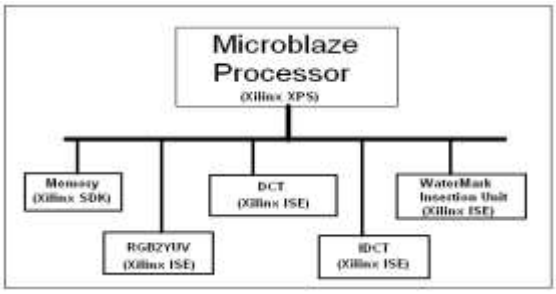

Fig. 6 Block diagram showing Overall Experiment

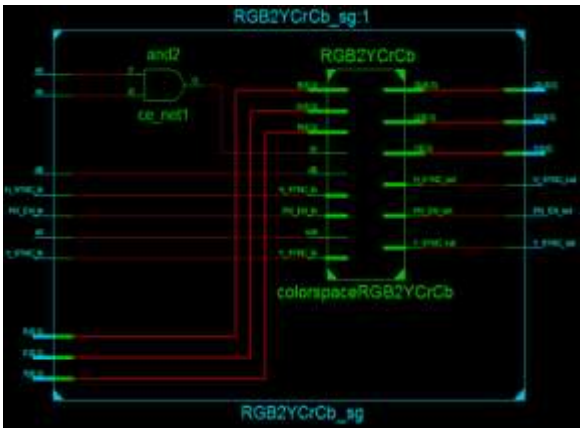

Fig. 7 RTL View of Component RGB to YUV Conversion

Major components of the algorithm like, RGB to YUV conversion, 2D DCT, Watermark insertion Unit and frame buffer manager are implemented on the hardware also. Below is the resource utilization table for the implementation work.

Table 2: Resource utilization summary

Components	Logic elements	Registers
RGB to YUV	1264	0
Watermark buffer	4277	4040
Watermark Insertion	30	0
Frame buffer	8077	4250
2D DCT	1542	157

Development of supporting components and optimization of implemented components is underway.

CONCLUSIONS

The paper presented an embedded solution for Video ownership using FPGA technology. The experiment is done for MPEG-4 compression. The experiment refers to an algorithm 'DCT domain WM algorithm for images' (Mohanty - Ramakrishnan and Kankanhalli, 1977) and extends implementation for video watermarking using FPGA technology. Current resaurce utilization and time analysis says that FPGA solutions are viable for video watermarking. Further development is going on to extend the real-time performance.

FUTURE WORK

1. The components utilizing less number of logic elements can be pipelined to achieve more performance.
2. RTL level subsystem can be optimized to improve resource utilization and minimize execution time.
3. Alternative hardware architectures using on-board memory and pipelining can also be explored for experiment.

ACKNOWLEDGEMENTS

The Author is Thankful to Dr. S.N.Pradhan for providing guidance for this topic. The Author is thankful to Nirma University, Ahmedabad, India, for providing research environment and Technical instruments for the experiments.

REFERENCES

1. Bannur, J and Varma A (1985). "A VLSI implementation of a square root algorithm', In IEEE Symposium on Comp. Arithmetic, pages 159-165. IEEE Computer Society Press, Washington D.C.
2. Brown, S and Jonathan R (1996). "Architecture of FPGAs and CPLDs: A Tutorial", IEEE Design & Test of Computers, Department of Electrical and Computer Engineering. University of Toronto.
3. Jain, AK, Yuan, L, Pari, PR, Qu, G (2003). "Zero Overhead Watermarking Technique for FPGA Designs", Department of Electrical and Computer Engineering, University of Maryland, College Park, GLSVLSI03, Washington, DC, USA.
4. Klingman, Ed(2004). "FPGA programming step by step", Embedded System Programming (Mar 2004), ArticleID-18201956.
5. Lach, J, William, H, Smith, M, Potkonjak, M (1996). "Robust FPGA Intellectual Property Protection Through Multiple Small Watermarks", International Conference on Multimedia Computing and Systems, UCLA EE Department, 56-125B Engineering IV, Los Angeles, CA 90095, 310-794-1630.
6. Latha, MM, pillai, GK, Sheela, KA (2005). "Watermarking based Content Security and Multimedia Indexing in Digital Libraries", International conference on Enabling Technologies for Smart Appliances(ETSA) IEEE, Department of Electronics and communication Engineering JNT University, Hyderabad, Andhra Pradesh, India.
7. Mohanty, SP, Ramakrishnan KR and Kankanhalli (2000) MS. "A DCT Domain Visible Watermarking Technique for Images", IEEE, 0-7803-6536-4/00/$10.00 (c) 2000.
8. Wain, R, Bush, I, Guest, M, Deegan, M, Kozin, I and Kitchen, C (2006). "An overview of FPGAs and FPGA programming; Initial experiences at Daresbury", Computational Science and Engineering Department, CCLRC Daresbury Laboratory, Warrington, Cheshire, WA4 4AD, UK, Version 2.0.
9. Wolf W(2005). "FPGA-Based System Design". Ziener, D and Teich J (2005). "Evaluation of Watermarking methods for FPGA-based IP-cores", Hardware/Software Co-Design, Department of Computer Science, University of Erlangen-Nuremberg, Germany, Co-Design-Report 01-2005, March, http://www12.informatik.unierlangen.de. Xilinx Site: http://www.xilinx.com/company