Automatic HLS based Low-Cost IP Watermarking

Bertrand Le Gal, Lilian Bossuet

Automatic HLS based Low-Cost IP Watermarking

Bertrand Le Gal
IMS Laboratory - UMR CNRS 5218,
ENSEIRB-MATMECA, University of Bordeaux, France
bertrand.legal@ims-bordeaux.fr

Lilian Bossuet
Hubert Curien Laboratory, UMR CNRS 5516,
Telecom Saint Etienne, University of Lyon, France
lilian.bossuet@univ-st-etienne.fr

***Abstract*—Currently, the Intellectual Properties (IP) and their reuse are common, however the use of IP is raising design security issues i.e. counterfeiter, reverse engineering. Watermarking is one of the efficient methods to detect an unauthorized IP use and a counterfeiter. In this context, many interesting works have been proposed. However, a few of them combine the watermarking process with the synthesis one. This article presents a new automatic and low cost watermarking solution. The design watermark is implanted in a high-level synthesis process. Some implementation results with Xilinx Virtex-5 FPGA assure the proposed solution low overhead compared to existing solution.**

I. INTRODUCTION

To handle the increased system complexity, companies reuse more and more IP cores. In consequence of the IP core business rise, the theft of IP is increasing [1]. The trade group founded by Ciso, HP, Nortel and 3COM (Alliance for Gray Market and Counterfeit Abatement [2]) estimates that the legitimate electronic companies loose almost $100 billion revenue per year due to counterfeiting. Counterfeiting prevention requires the technological solution in addition to the legal process i.e. patents.

In order to identify an IP theft, a novel watermarking solution (design methodology, which originally answers to the IP security required by actual design reuse market) is presented in this article. To reduce the watermarking overhead (area, delay, power consumption and design time), a mark is automatically inserted in the design. It is done during the behavioral synthesis process by using a High Level Synthesis (HLS) tool [3, 4].

The paper is structured as follows. Section II describes a state of the art of the watermarking solutions. Section IV details the new proposition of watermarking. Section V presents the main parts of the modified HLS automatic flow. Finally section VI gives experimental results with signal processing benchmarks on Xilinx Virtex-5 FPGA target.

II. RELATED WORKS

According to [5] the goals of IP protection are: (1) to enable IP providers to protect their IPs against unauthorized use, (2) to protect all types of design data used to produce and deliver IPs, (3) to detect an unauthorized use of IPs, (4) to trace an unauthorized use of IPs.

IP unauthorized use detection involves the ability to determine that an unauthorized use has occurred and then to trace the source of theft. To answer the detection issue, IP providers introduce shadow digital signature.

Digital watermarking is an indirect protection scheme which demonstrates the ownership of an IP. The concept of active watermarking consists of a digital signature insertion into an IP. Many approaches have been developed depending on the watermarking abstraction level:

- **Application level** digital signature hide examples can be found in [6–9]. A Digital Signal Processing (DSP) watermarking is introduced in [6].
- Authors in [7] target an **algorithmic level** watermarking in the design flow. Both approaches [6] and [7] are based on the idea of slightly changing the gain of filters, without affecting the system behavior. Two different watermarking techniques at behavioral level are introduced in [8] and [9]. Both algorithms are based on adding new input/output sequences in the design finite state machine (FSM) representation.
- Digital signature hiding at **logical synthesis** level can be found in [10, 11, 12, 13]. In [10] Hong presents the watermarking combinational logic synthesis solutions. The watermarking behavioral synthesis techniques for IP protection are described in [11].
- Mostly the **post-synthesis** watermarking introduces constraints [14] and [15]. An example is the addition of extra hardware, like buffer [16] or dedicated embedded tester [17].

The pre-synthesis watermarking techniques are application dependant, and their over-cost is not measurable. The post-synthesis techniques are time consuming and design/device dependant. The in-synthesis watermarking techniques introduce power/area/timing overhead. Generalizing the watermarking usage for IP identification is important. However, it requires enough generic watermarking techniques with low overhead cost. Such techniques must be implemented in automatic design flows due to *time to market* constraint. It is for a fast component tagging in a designer friendly process. For these reasons, a new in-synthesis watermarking scheme is devised in the following section.

III. ARCHITECTURAL SYNTHESIS CONCEPTS

Our methodology targets custom hardware architectures, dedicated to the computational intensive applications for signal and video processing. In most usual video and signal processing applications, full pipeline architecture is quite inefficient. In fact, such architectures are too area and power consuming. Usually, there is a trade-off between the high-performance and the low-cost (area, power consumption) architectures [18] and [19]. An efficient architecture shares the hardware resources (operators and registers) during an application execution. This resource sharing makes some input and output slots free and ready to tag.

The circuit inputs and outputs permit it to receive and send data from/to the system. Figure 1 presents the IO behavior for a FIR filter application. This IP component receives data from the system (*Xn*), performs computations and then provides valid result to the system (*Yn*). The output time slots between two successive high levels of data valid signal are unused. These free output slots are grey-colored in Figure 1.

The proposed idea is to employ the empty output slots for the design watermarking. The circuit watermark is composed of a set of mathematical relations. These relations are based on circuit input values, initial values and internal results. Each mathematical relation is a sub-mark. The sub-marks are read like an output value during free output slots (when data valid is inactive). The watermark is invisible for an IPs integrator. It is because these sub-marks results look like the dynamic transient output values. Consequently, the watermark is invisible from static analysis.

The proposed method is well adapted for general-purpose applications like digital signal, image or video processing etc. Nevertheless, this method is not appropriate for data-security application such as data encryption, data integrity or data authentication. In such cases, the IP watermark can cause a dramatic data security failure

The next section details this novel watermarking technique, which uses output dynamic sub-marks to watermark the IP.

IV. NEW IPP PROPOSITION BY WATERMARKING

This novel technique is based on the free output slots behavior. These output slots can be modified by introducing custom design singularities (that are IP internal values). The proposed method is based on the assertion that the hardware IP has free output slots.

Depending on the required protection level and the watermark cost allowed, we propose two watermarking solutions with different area costs: random low-cost watermark and cost-less watermark.

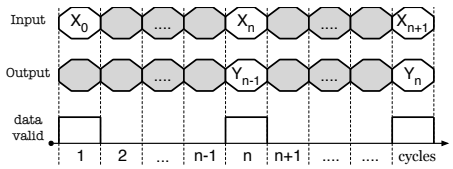

Figure 1. Example of an IO behavior.

Random low cost watermark is characterized by a set of randomly chosen architecture internal values. A special data-path generates each sub-mark by transferring the chosen internal value (during the choosen clock cycle) to a free output slot. The watermarking area cost is due to the data path modification (output multiplexer resizing and FSM controller modifications). The data path area overcost could be very low. It is particularly true for FPGA implementation. In fact, while using FPGA, the reconfigurable data paths do not cost area. It is because the data paths set is directly available in the device. However, increasing input multiplexer size could cost area.

Cost-less watermark is a low-cost one with an usable reduced set of internal values. It employs the existing dynamic transient outputs during the free slots (internal values which do not required new path creation in the datapath). To introduce this kind of watermark, the only change required consists in modifying IP control unit to drive selected internal data to output ports. HLS tools design the IP control unit with a Finite State Machine (FSM). The cost-less watermark affects only the FSM. Experimental results on the FPGA target presented in section IV will assure that the modifications are cost-less and even could decrease the IP area occupation. In fact, the area increase or decrease depends on the FSM modification and the logical synthesis process. Hence, it is hard to estimate.

Figure 2 and Figure 3 present the main distinction between the low-cost and the cost-less watermark. In these examples the architecture required modifications are represented with dotted lines. Figure 2 describes the case of low-cost watermarking mode. It allows the algorithm to allocate new data-path (multiplexers and wires). It is to drive the

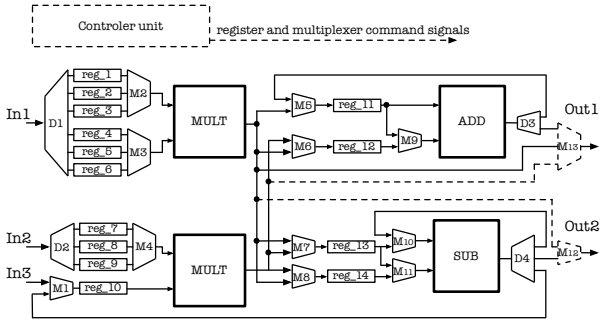

Figure 2. Low cost watermarked architecture reusing existing resources, including also new dedicated data path allocations (dotted line).

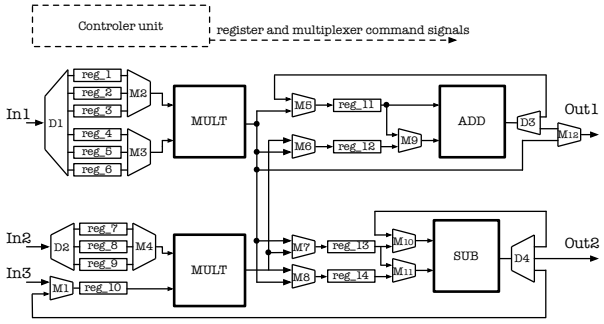

Figure 3. Cost-less watermarked architecture reusing existing paths controller with multiplexer control modifications (dotted line).

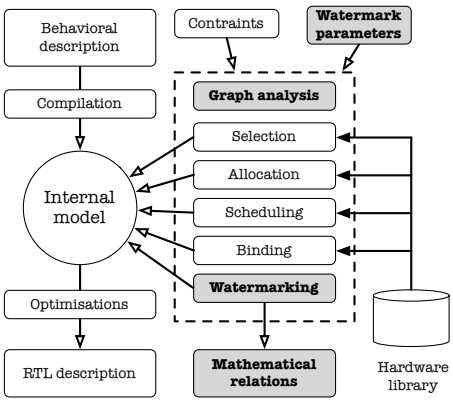

Figure 4. High-Level Synthesis design flow including the proposed watermarking technique (grey-colored).

inaccessible data to outputs, increasing the design singularities. Figure 3 presents the case of second watermarking scheme (cost-less watermarking). Here, the data paths are already allocated to implement the behavior computations. They drive the internal data to outputs, only design controller is modified.

V. WATERMARKING AUTOMATION FLOW

The proposed technique is integrated as a part of a HLS design flow. It permits the designer to automatically include copyright information in generated circuits. The designer configures the watermark generation specifying: (1) the watermark length (number of sub-marks). (2) the number of distinct clock cycles to mark (3) the choice between *cost-less* or *low-cost* watermarking technique. After the automatic watermarking step, the designer has a watermarked IP, generated under the system constraints. The tool provides him a file containing the design watermark. This file contains the output time slots where the sub-marks are produced and the existing mathematical relations. This information may keep secret from a consumer point of view, as it may be used to proof the circuit ownership.

A. HLS design flow modifications

Figure 4 presents the design flow modifications compared to the usual HLS ones [17]. The watermarking stage composed of four steps: (1) Graph analysis, to find the usable internal data (2) Possible watermark enumeration (3) Internal values for random selection. (4) Architecture modifications: datapath change, multiplexer resizing and control unit modification.

B. Selecting the marks and modifying the design

By using the designer provided parameters, an automatic process computes the number of possible watermarks (depending on the number of registers, the number of internal values and their associated lifetime). Depending on this result, it computes the average number of marks to introduce per clock cycle. The watermark repartition is then randomly performed to tag the required number of clock cycles. Once these computations are performed, a mapping algorithm is applied to find the most singular data from the design. It is done (1) to select them for sub-mark usage (2) to drive them to one of the output port.

1. $outList \leftarrow$ IdentifyDesignOutputsPorts ($graph$)
2. $fosList \leftarrow$ SearchFreeOutputSlotsDuringExecution ($outList$, $graph$)
3. **if** CountSlots($fosList$) $< W_{count}$ **or** CountClockCycles($fosList$) $< h$ **then**
4. **return false**
5. **end if**
6. $ufosList \leftarrow$ RandomlySelectFreeOutputSlots(W_{count}, h)
7. $regList \leftarrow$ SearchRegistersHavingAnExistingPathToOutputs($graph$, $ufosList$)
8. **if** LowCostMode $= true$ **then**
9. $regList \leftarrow regList +$ RegistersWithoutExistingPathToOutputs($graph$, $ufosList$, $authorized\_cost$)
10. **end if**
11. $dataList \leftarrow$ ListUsableInformationFromRegisters ($regList$, $ufosList$)
12. $ufosList \leftarrow$ RandomlySelectDataForOutputWatermarking ($dataList$, W_{count}, h)
13. ModifyCircuitAccordingToWatermarkingChoices ($graph$, $ufosList$)
14. StoreRelationsBetweenInternalComputationAndOutputs ($ufosList$, $filename$)
15. **return true**

Figure 5. Generic cost-less and low-cost watermarking procedure.

For each internal data mapping to output, the tool analyzes the required logical glue over-cost (due to multiplexer input ports increase) in order to find the best couple (internal data, output) for low area design cost. An overview of the watermarking algorithm is provided in Figure 5.

This process is repeated for each sub-mark that the tool must insert in the design to respect the watermark length constraint given by designer.

C. Generating the watermark file

The last step of the watermarking procedure is the Watermark properties file creation. This stage aims to create a file, containing all the inserted IO marks related information (1) the couples {output port, sub-mark} (2) the clock cycle during which the sub-mark is produced (3) the mathematical equation that defines the sub-mark (see Equation 2).

These data correspond to the information required to identify a cloned IP.

VI. EXPERIMENTS

To evaluate the proposed watermark design methodology efficiency in terms of area and critical path delay, experiments with signal and image processing benchmarks are conducted. The watermarked IP implementations are done using a Xilinx Virtex-5 FPGA. Results obtained using cost-less watermark are presented in Table 1.

For each design (i.e. IP), the following circuit parameters are provided: the number of FSM states, the number of I/O ports, the maximum mark length (number of free output slots), the introduced watermark length (0% for reference design, 50% or 100% for watermarked ones). The right columns provide: the number of available different watermarks and the circuit area and critical path obtained after logical synthesis. Penalties for watermarked IP are obtained from comparison to unprotected one. Logical synthesis results were obtained using the Xilinx ISE 10.1 tool.

Area and timing overhead are functions of the watermark selection algorithm. As it is shown in section III, area and critical path length overhead come from the datapath changes (some multiplexers are allocated) and from the control unit changes (FSM instruction decoder is modified to drive data and control new multiplexer).

TABLE I. PROPOSED COST-LESS WATERMARKING IMPLEMENTATION COSTS ON A VIRTEX-5 DEVICE.

				Without watermark		Watermark length = 50%			Watermark length = 100%		
Application	# of FSM States	# of I/O ports	# free slots	Area	C. Path	# of cost-less watermarks	Area overcost (%)	C. Path overcost (%)	# of cost-less watermarks	Area overcost (%)	C. Path overcost (%)
FIR 64-taps	26	1/1	25	12351	15,499	2^46	0,05 %	0,01 %	2^50	0,02 %	-0,01 %
	38	1/1	37	6612	14,613	2^70	-0,02 %	0,00 %	2^74	0,05 %	0,01 %
LWT 16-taps	25	2/2	34	14079	16,312	2^72	0,04 %	-0,27 %	2^87	-0,18 %	-1,31 %
	64	2/2	114	12028	16,346	2^317	-0,36 %	1,05 %	2^421	0,17 %	-0,61 %
SSD 16x16	35	8/1	34	11078	16,103	2^65	0,09 %	0,00 %	2^68	-0,01 %	0,00 %
	81	1/1	80	3193	15,689	2^156	-0,06 %	0,00 %	2^160	0,09 %	0,00 %
1d DCT 8 taps	15	4/4	56	8818	15,185	2^125	0,10 %	-0,05 %	2^144	0,15 %	-0,02 %
	20	1/1	13	6384	14,991	2^26	0,03 %	-0,03 %	2^33	0,03 %	-0,03 %
2d DCT 8x8 taps	80	8/8	584	31428	17,259	2^1323	-0,32 %	0,41 %	2^1509	0,02 %	-0,24 %
	160	1/1	97	25469	17,355	2^346	-0,30 %	-0,06 %	2^547	0,08 %	-0,59 %
Matrix product 8x8	86	8/4	280	62784	16,104	2^880	-0,24 %	0,63 %	2^1210	-0,11 %	0,71 %
	141	1/1	77	31117	17,201	2^443	0,01 %	0,34 %	2^445	0,05 %	0,13 %
FFT 64 taps	90	8/8	600	51086	17,255	2^1634	0,02 %	-0,01 %	2^2106	0,04 %	0,05 %
	180	2/2	234	31589	17,181	2^814	0,04 %	0,02 %	2^1180	0,17 %	-0,84 %

Table 1 shows, in the case of cost-less watermarking, that the controller changes have a very low impact on the global IP component characteristics. Moreover the number of different watermarks that can be used to protect the design is quite important in the overall examples. Design area varies from -0,36% up to +0,17% when critical path progress from −1, 31% to +1, 05%. However, Table 1 shows that the mark length is much smaller for a high-level security. For some designs, such as SSD 16×16, unusual area and critical path reductions come from FSM signal command modifications. It may involuntarily results into a better logical equation simplification during logical synthesis. While considering the low-cost watermarking experimentations (result table is not provided due to article page limit), the area and timing deterioration are higher like the possible number of watermarks available. Design area increase from 0.07% to 1.02% while circuit critical path evolve from −1.29% to 1.45%. However, watermarking penalties are still low for most of the experiments (area increase average = 0.55% and latency average = 0.08%) These experimental results confirm the interest and the low cost of the proposed watermarking techniques which required low runtimes (only a few seconds).

VII. CONCLUSION

In this paper, a new watermarking technique for behavioral IP components and its synthesis design flow have been presented. The proposed technique is used for automatic Intellectual Property protection by using the HLS tools. The essence of this new approach is the set of mathematical marks on the design output ports which encode the IP watermark (copyright information). The mathematical marks are selected and inserted during the synthesis process. It is done in such a way that they result into the minimal hardware overhead while embedding the signature. Finally, the watermark are difficult to detect and remove.

REFERENCES

[1] M. Pecht and S. Tiku, “Bogus ! Electronic manufacturing and consumers confront a rising tide of counterfeit electronics”, IEEE Spectrum, Available from: http://www.spectrum.ieee.org/print/3423.

[2] http://www.agmaglobal.org.

[3] D. Gajski, et al., “High-Level Synthesis: Introduction to Chip and System Design”, Kluwer Academic Publishers, 1992.

[4] P. Coussy and A. Morawiec (Eds.), “High-Level Synthesis from Algorithm to Digital Circuit”, Springer, 2008.

[5] R. Chapman, T. Durrani, “IP Protection of DSP algorithms for system on chip implementation”, IEEE Transactions on Signal Processing 48 (3) (2000) 854–861.

[6] Rashid, J. Asher, W. Mangione-Smith and M. Potkonjak, “Hierarchical watermarking for protection of dsp filter cores”, In Proceedings of the IEEE Custom Integrated Circuits Conference, 1999, pp. 39–42.

[7] Torunoglu and E. Charbon, “Watermarking-based copyright protection of sequential functions”, IEEE Journal of Solid-State Circuits 35 (3) (2000) 434–440.

[8] Oliveira, “Techniques for the creation of digital watermarks in sequential circuits designs”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 20 (9) (2001) 1101–1117.

[9] F. Koushanfar, I. Hong and M. Potkonjak, “Behavioral synthesis techniques for intellectual property protection”, ACM Transactions on Design Automation of Electronic Systems 10 (3) (2005) 523–545.

[10] D. Kirovski, Y. Y. Hwang, M. Potkonjak and J. Cong, “Intellectual property protection by watermarking combinational logic synthesis solutions”. In Proceedings of ICCAD’98, 1998, pp. 194–198.

[11] Lach, W. H. Mangione-Smith, M. Potkonjak, “Robust FPGA intellectual property protection through multiple small watermarks”, In Proceedings of DAC ’99, NY, USA, 1999, pp. 831–836.

[12] A. Cui, C.H. Chang and S.Tahar, "IP Watermarking Using Incremental Technology Mapping at Logic Synthesis Level". IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 27, No 9, pp. 1565-1570, September 2008.

[13] C.H. Chang and A. Cui, "Synthesis-for-testability watermarking for field authenticatioil of VLSI intellectual property". IEEE Transactions on Circuits and System-I, vol. 57, no 7, pp. 1618-1630, July 2010.

[14] A. Jain, L. Yuan, P. Pari, G. Qu, “Zero overhead watermarking technique for FPGA designs”, In Proceedings of the 13th ACM Great Lakes symposium on VLSI (GLS-VLSI), 2003, pp. 147–152.

[15] G. Sun, Z. Gao, Y. Xu, “A watermarking system for ip protection by buffer insertion technique”, In Proc. of ISQED, 2006, pp. 671–675.

[16] Y. C. Fan, H. W. Tsao, “Watermarking for intellectual property protection”, Electronics Letters 39 (18) (2003) 1316–1318.

[17] B. Le Gal, E. Casseau, S. Huet, “Dynamic memory access management for high-performance DSP applications using high-level synthesis”, IEEE Transactions on VLSI Systems 16 (11) (2008) 454–1464.

[18] Arvind, R. S. Nikhil, D. L. Rosenband, N. Dave, “High-Level Synthesis: an essential ingredient for designing complex ASICs”, In Proceedings of ICCAD’04, DC, USA, 2004, pp. 775–782.

[19] P. Coussy et al., “GAUT - a high-level synthesis tool for DSP applications”, High-Level Synthesis from Algorithm to Digital Circuit (XVI) (2008) 147–169.