HAL
open science

Functional Locking Modules for Design Protection of Intellectual Property Cores

Brice Colombier, Lilian Bossuet

To cite this version:

Brice Colombier, Lilian Bossuet. Functional Locking Modules for Design Protection of Intellectual Property Cores. IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines, May 2015, vancouver, Canada. pp.233, 10.1109/FCCM.2015.17 . ujm-01164036

HAL Id: ujm-01164036

https://ujm.hal.science/ujm-01164036v1

Submitted on 17 Jun 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Functional Locking Modules for Design Protection of Intellectual Property Cores

Brice Colombier, Lilian Bossuet
Hubert Curien Laboratory, UMR CNRS 5516, University of Lyon
42000 Saint-Étienne - France
{b.colombier, lilian.bossuet}@univ-st-etienne.fr

I. Introduction

Electronics systems design is increasingly uses Intellectual Property (IP) cores. The means, however, that can render the IP core unusable if it has been obtained illegally [1] have not yet been identified. We describe lightweight locking schemes lacking in the state of the art. In Section II we identify common locking points on an IP, before describing locking schemes in Section III. Section IV concludes.

II. Locking-based protection schemes

Authentication and locking schemes can be combined to fight counterfeiting and overbuilding [2], [3]. In case the IP core was illegally obtained, a locking circuitry makes it unusable. Common features can be turned into locking schemes: the FSM [2] or the CPU, the I-O ports [4], the clock manager or the address part of the memory bus.

III. Description of proposed locking schemes

Locking a finite state machine: The first way to achieve functional locking is to add an extra FSM, the locking FSM, before the start state of the original FSM [2]. This controls access to the normal behaviour of the device.

Processor backup and restore: The processor used here is the Plasma CPU [5]. By holding the current program counter value, the processor stops fetching new instructions. When the processor is locked, it executes `NOP` instructions instead. The locking procedure cannot be initiated at any time though. Indeed, long and branching instructions are problematic and do not allow for a correct return to normal operation. They can be detected using the CPU opcodes. Locking is not time-critical. It can be postponed for several clock cycles, to ensure a safe return to normal operation.

Locking inputs: An IP can be locked by preventing it from receiving data, by acting on the *clock-enable* input of the input flip-flops. By setting this input at low level, the flip flop keeps its previous value, and the circuit will be locked.

Clock signal modifier: Acting on the *clock-enable* input of a clock buffer modifies the clock signal. The aim here is to place as few elements as possible on the clock signal path. FPGAs from Altera and Xilinx have built-in clock buffers, useful for clock-gating, or for high fan-out clocks. The *clock-enable* input of the clock buffer is controlled by the output of a three-to-one multiplexer. It selects one of the following signals: a high logic level, for full-functionality mode, a low logic level, for locked mode, and the output of a mod N counter generating a pulse when its value is 1 for evaluation mode, with lower frequency and thus lower performance.

Phase-locked loop (PLL) reconfiguration: Most modern FPGAs embed reconfigurable PLLs. The reconfiguration procedure, however, is specific to each FPGA vendor, and requires a proprietary module. The overhead is high, and can not be significantly reduced without replacing this module.

Memory bus pseudo-random scrambling: To functionally lock the circuit, the address part of the memory bus can be scrambled to make read data unreliable. An LFSR is used as a pseudo-randomness source for scrambling. We need to carefully chose the LFSR feedback polynomial for a lightweight implantation, so that most of its coefficients are 0s, since each coefficient equal to 1 requires a XOR gate. The n bits of the LFSR are then XORed with the n-bit address bus. Finally, a multiplexer selects the original address bus or the scrambled one.

IV. Conclusion

We compared features of an IP that can be leveraged for functional locking. Clock-based locking is a flexible, powerful yet lightweight option. A balance between efficiency and resources is the main point addressed by the designer.

Acknowledgement: The work presented in this paper was realized in the frame of the SALWARE project number ANR-13-JS03-0003 supported by the French "Agence Nationale de la Recherche" and by the French "Fondation de Recherche pour l'Aéronautique et l'Espace", funding for this project was also provided by a grant from "La Région Rhône-Alpes".

References

[1] C. Gorman, "Counterfeit chips on the rise," *IEEE Spectrum*, vol. 49, no. 6, pp. 16–17, 2012.

[2] Y. Alkabani, F. Koushanfar, and M. Potkonjak, "Remote activation of ICs for piracy prevention and digital right management," in *IEEE/ACM international conference on CAD*, China, 2007, pp. 674–677.

[3] L. Gaspar, V. Fischer, T. Guneysu, and Z. C. Jouini, "Two IP protection schemes for multi-FPGA systems," in *International Conference on Reconfigurable Computing and FPGAs*, Mexico, 2012, pp. 1–6.

[4] A. Basak, Y. Zheng, and S. Bhunia, "Active defense against counterfeiting attacks through robust antifuse-based on-chip locks," in *IEEE 32nd VLSI Test Symp.*, USA, 2014, pp. 1–6.

[5] Opencores website. [Online]. Available: opencores.org