BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iaşi
Tomul LIX (LXIII), Fasc. 1, 2013
Secţia
ELECTROTEHNICĂ. ENERGETICĂ. ELECTRONICĂ

# MASKING THE INSTRUCTIONS OF A MICROCONTROLLER USING A “CHAOTIC” POWER SUPPLY

BY

**EMANUEL-FLORIN IFTENE[1] and**
**HORIA-NICOLAI TEODORESCU[*1,2]**
Corresponding Member of the Romanian Academy

[1]“Gheorghe Asachi” Technical University of Iaşi
Faculty of Electronics, Telecommunications and Information Technology,
[2]Institute of Computer Science of the Romanian Academy, Iaşi Branch

Received: March 5, 2013
Accepted for publication: March 29, 2012

**Abstract.** A protection method based on injecting fault currents in the power supply lines of microsystem by means of a power source controlled by a chaotic generator is proposed. This preliminary, short paper, studies the efficiency of this method in masking the pattern of executed instructions on an 8-bit microcontroller.

**Key words:** side attack; power analysis; microcontroller; DPA; SPA.

## 1. Introduction

Embedded systems are subject to side attacks – power analysis attacks, such as Simple Power Analysis (SPA) or Differential Power Analysis (DPA) – that reveal and extract sensitive data from microsystems. The power analysis was first introduced in the late 1990s as a method to analyse and minimize the power consumption on microsystems. Kocher *et al*. (2012) showed that this method could be used by an attacker to reveal sensitive data from microsystems. This kind of attack is named *simple power analysis* or *differential power analysis*, depending on the details of the attack. This analysis relies on the

[*]Corresponding author: *e-mail*: hteodor@etti.tuiasi.ro

internal circuits of the microcontrollers, which are simple to complex switching CMOS gates; the current depends on the number of switching gates during each performed operation. Based on the facts that logic CMOS gates absorb current from the power supply mostly on transitions from logic level “0” to logic level “1” and that one instruction activates a pre-defined number of gates, it results that each executed instruction has its own unique pattern. Thus, each instruction has its own signature, partly changed by the data manipulated.

## 2. Protection Method

### 2.1. Setup

The proposed protection method aims to mask the instructions performed by the system. It makes use of the empirically determined ability of microcontrollers (we tested only PIC 16 series) to properly operate under relatively large fluctuations of the power supply. The protection consists in using the chaotic signal to control the voltage drop on a series transistor, thus modifying the current consumption “seen” by the attacker.

Fig. 1 represents a simplified diagram for the method implementation and for power analysis, simple or differential. It consists of a power supply for the microsystem, a digital oscilloscope for recording the data, and a series resistor, $R_1$ = 50 Ω, used for measuring the voltage drop resulted from the current variations during the execution of the program by the microsystem. The PIC16F877A-I/P microcontroller was used for the experiments.

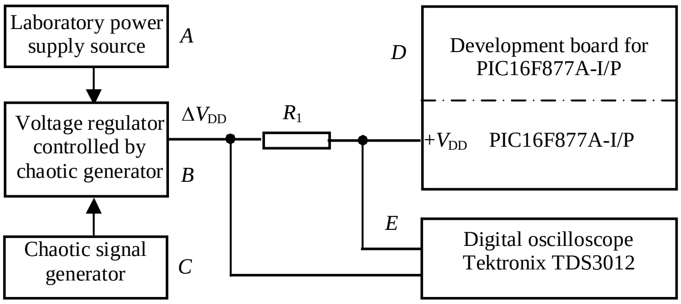

Laboratory power supply source
A
Voltage regulator controlled by chaotic generator
B
Chaotic signal generator
C
$\Delta V_{DD}$
$R_1$
D
Development board for PIC16F877A-I/P
$+V_{DD}$
PIC16F877A-I/P
E
Digital oscilloscope Tektronix TDS3012

Fig. 1 – Block schematics of the protection method.

The schematic block from Fig. 1 is composed of a laboratory power supply (*A*) that provides the necessary voltage for the voltage regulator (*B*); a variable voltage supply; a chaotic signal generator (*C*) that controls the voltage switching of the voltage supply (*B*); the resistor $R_1$, used as current-to-voltage sensor; a digital oscilloscope (*E*) that measures the voltage signal across the resistor $R_1$, and a development board (*D*) based on PIC16F877A-I/P microcontroller.

### 2.2. Masking Method Based on Chaotic Generator

The detailed schematic of the chaotic generator used to drive the power supply of the development board was presented and discussed in previous works (Teodorescu, 2010; Teodorescu & Cojocaru, 2011). The resulted waveform of the chaotic generator is presented in Fig. 2. The average frequency of the chaotic generator is about 1 MHz. Fig. 2 illustrates the frequency variations produced by the chaotic generator and its annex circuit. An example of variation is marked with the dotted line circle in Fig. 2.

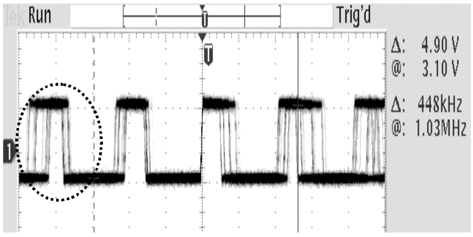


Fig. 2. – Output waveform of the chaotic generator.

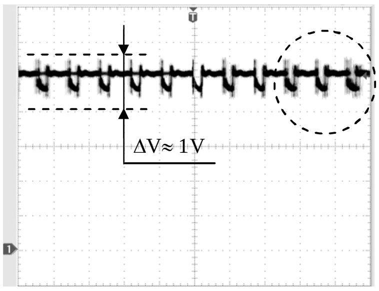


Fig. 3. – Output waveform of the voltage regulator (*B*).

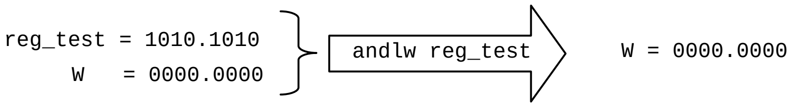
The voltage regulator (*B*) is based on the schematic for voltage regulators with series transistor. It has two branches for regulating the output voltage and an additional transistor that switches the active branches, according to the amplitude of the command signal received from the chaotic generator. Therefore, the “regulator” actually modifies the voltage chaotically.

The output waveform shown in Fig. 3 is the resulted output voltage of the voltage regulator (*B*); on Fig. 3 is marked with a dotted circle the frequency variations injected by chaotic clock and with two dotted lines the amplitude variations of about 1 V of the output voltage. The output voltage swing is computed based on voltage range accepted by PIC16F877A-I/P microcontroller of min. 3.5 V and max. 5.5 V. Given this range, the first branch of the voltage regulator limits the maximum output voltage at about 5.5 V, while the second branch limits the minimum output voltage at about 4.5 V.

For testing the efficacy of this method of protection for the microsystems, we collected the signals and determined the unperturbed pattern of the executed instructions by the microcontroller, and then the patterns of the same instructions, with the power supply perturbed as described. The test program executes a single instruction repeated in a loop. Using the digital oscilloscope the current variations on the series resistor are registered and analysed. Then we repeated the experiments using the proposed method for protection from Fig. 1. The resulted waveform should be different from the first case, given the efficiency of the masking method of the pattern for the executed instruction.

### 2.3. The Test Program

The test program is written in Assembly Language and comprises one single loop program that has 100 consecutive `andlw` instructions; each `andlw` instruction is executed in a single machine cycle. The test register `reg_test` manipulated by the instruction `andlw` has the same value during the test program. The resulted value is stored in the accumulator register, *W*; in this way the resulted pattern is the same and is not affected by the value manipulated by the test register or by the resulted value.



```
rutina_andlw:
       bsf     led_galben     ; control LED activated
       andlw   reg_test       ; execution #1 of the instruction andlw
       andlw   reg_test       ; execution #2 of the instruction andlw
       ................................ ; ....
       andlw   reg_test       ; execution #99 of the instruction andlw
       andlw   reg_test       ; execution #100 of the instruction andlw
       bcf     led_galben     ; control LED deactivated
       goto    rutina_andlw   ; go to rutina_andlw
```

In the first case the microcontroller is using the power supply and a quartz driven clock oscillator at the frequency of 4 MHz. The PIC16F877A-I/P microcontroller uses a four stage pipeline, namely fetch, decode, execute and write-back and one clock period for each stage; therefore for each one machine cycle instruction are required four clock periods. That gives a period of $T = \dfrac{1}{F_{\text{osc}} / 4} = \dfrac{4}{4\text{ MHz}} = 1\ \mu\text{s}$ per machine cycle.

### 2.4. Test for Masking Strength

To verify that the microsystem operates correctly with the proposed method of protection (Fig. 1), the program activates a control LED at the beginning of the loop; at the end of the loop the control LED is deactivated. The control LED is connected on port RA2 of the microcontroller with a series resistor that limit the LED current to about 8 mA. This extra-current is seen on the digital oscilloscope as a voltage drop on series resistor $R_1$ for a period of 3 µs, given by the execution of the instructions `bcf` and `goto` from the end of the loop of the test program, as shown in Fig. 4 *a*, or once per 100 µs, as represented in Fig. 4 *b*.

Fig. 4 *a* shows the waveform recorded on $R_1$ when the microcontroller is using power from the laboratory power supply; notice the end of the program loop with a marked period of 3 µs and a voltage drop of about 400 mV when the control LED is deactivated.

The waveform in Fig. 4 *b* was recorded when the microcontroller was using the protection illustrated in Fig. 1, with the power supply controlled by the chaotic generator; the marked time of 100 µs represents the duration of one cycle in the loop of the test program. From this observation we conclude that the microsystem is working correctly with this power supply controlled by the chaotic generator.

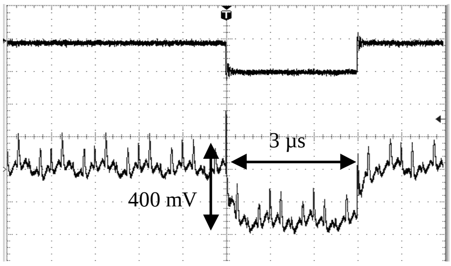


Fig. 4 *a* – The resulted waveform for a section of the test program.

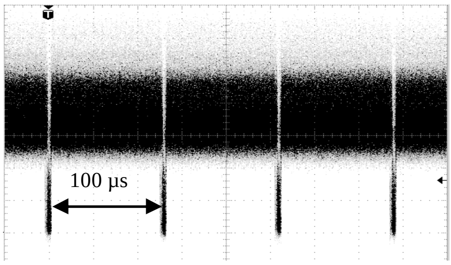


Fig. 4 *b* – The resulted waveform for the test program .

## 3. Results

### 3.1. Correlation Analysis for andlw Instruction

Consider first that the system operates with a stable clock and a stable power supply (no active protection). The pattern for andlw instruction under normal conditions is shown in Figs. 5 *a* and *b*, top side, in two different running instances. The recorded data was analysed on Matlab using the correlation and auto-correlation functions, $C_{xx}[k]=\sum_{i=1}^{n} x[i]x[i+k]$, $\mathrm{C}_{xy}=\sum_{i=1}^{n} x[i]y[i+k]$, and the resulted pattern is shown in the bottom panels of the Figs. 5 *a* and *b*.

The digital oscilloscope can store up to 10,000 samples on a single record; therefore, for a time base of 200 ns, 2 µs can be stored, that correspond to two complete executions for andlw instruction. As a result of the correlation function shown in Figs. 5 *a* and *b* (bottom) are a maximum at 1,250 samples meaning one stage from the pipeline and a maximum at 5,000 samples which is the period of the executed instruction. We checked that the correlation pattern is the same for different instances runs of the test program.

In the second phase of the experiment we used the protection method as in Fig. 1. The resulted waveforms are shown in Figs. 6 *a* and *b*. Notice, on the top side of Figs. 6 *a* and *b* that the pattern for the executed instruction, andlw, is different for the pattern from Figs. 5 *a* and *b*, and from one instance to another.

This is due to current fault injection from the power supply controlled by the chaotic generator. On the bottom side of the Figs. 6 *a* and *b* are represented the resulted correlation patterns; notice that the resulted correlation graph does not reveal the corresponding machine cycle or the period of the pipeline stages.

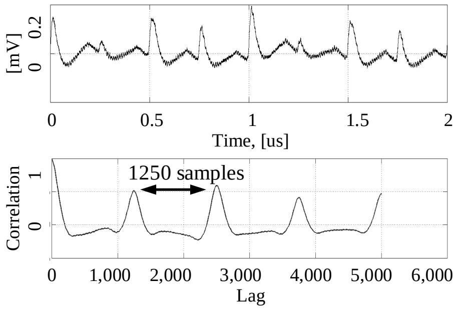


Fig. 5 *a* – Instruction `andlw` – top side default pattern, bottom side auto-correlation pattern – instance run #1.

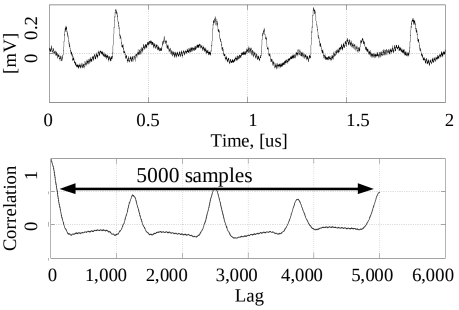


Fig. 5 *b* – Instruction `andlw` – top side default pattern, bottom side auto-correlation pattern – instance run #2.

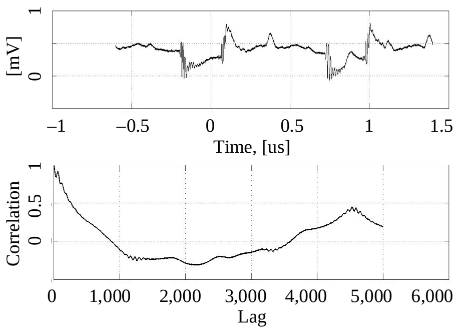


Fig. 6 *a* – Instruction `andlw` – top side resulted pattern with active protection; bottom side auto-correlation pattern – instance run #2.

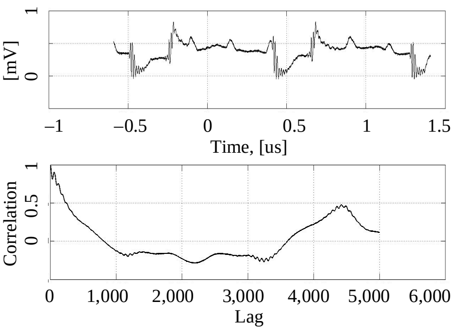


Fig. 6 *b* – Instruction `andlw` – top side resulted pattern with active protection; bottom side auto-correlation pattern – instance run #3.

From the results shown in Figs. 7 *a* and *b*, we derive that the default pattern of the `andlw` instruction with no protection circuit (top side) is different from the pattern of the `andlw` instruction (middle side) when the protection circuit is active. Furthermore, the correlation between two successive waveforms, shown on the bottom side of the Figs. 7 *a* and *b*, neither determines a period of 1,250 samples, nor a period of 5,000 samples (the last one corresponding to one machine cycle) and are different from one instance of execution to another. Based on these observations one can conclude that this protection method is efficient against power analysis attacks based on the pattern correlation.

This conclusion was validated for several instructions. A supplementary example is shown in Figs. 8 *a* and *b* for the `addwf` instruction. The test program add the `reg_test` register, `reg_test = b'00000000'`, with *W*. The pattern of the `addwf` instruction with no protection circuit activated is shown in top side of Figs. 8 *a* and *b*; on the middle of Figs. 8 *a* and *b* is represented the resulted waveform with protection circuit activated and on the bottom of Figs. 8 *a* and *b* are depicted the resulted correlation patterns.

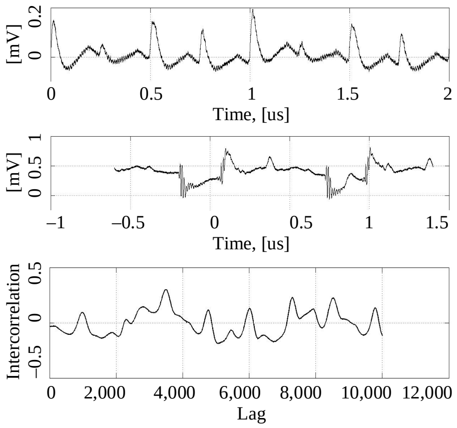


Fig. 7 *a* – Pattern of the `andlw` instruction with no protection circuit (top); with protection circuit (middle) – instance run #1; correlation (bottom).

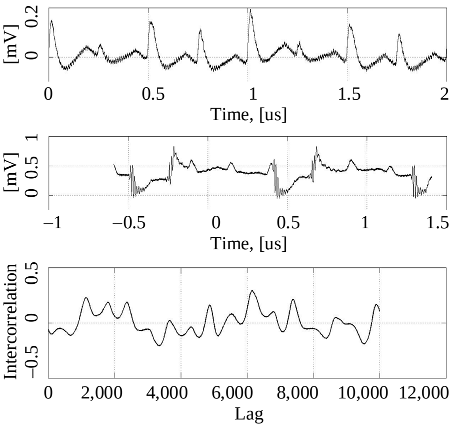


Fig. 7 *b* – Pattern of the `andlw` instruction with no protection circuit (top); with protection circuit (middle) – instance run #2; correlation (bottom).

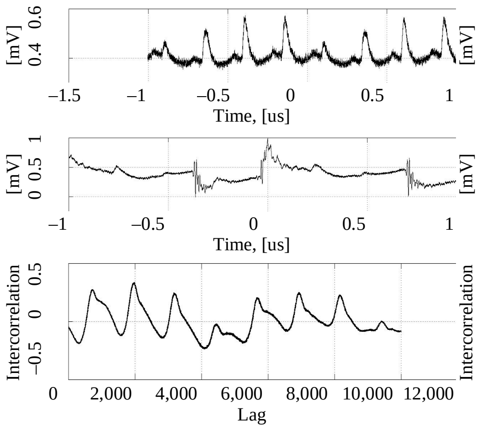


Fig. 8 *a* – Pattern of the `addwf` instruction with no protection circuit (top); with protection circuit (middle) – instance run #3; correlation (bottom).

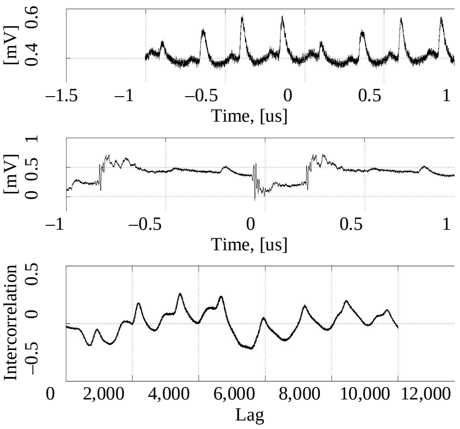


Fig. 8 *b* – Pattern of the `addwf` instruction with no protection circuit (top); with protection circuit (middle) – instance run #7; correlation (bottom).

## 4. Conclusions

This preliminary paper proposes a method for protecting embedded systems against side channel attacks, based on chaotic generators and fault current injection in power supply lines of the microsystem by means of a power source controlled by a chaotic generator. A comparison between the resulted waveforms, with and without protection circuit, shows that the method is effective. Detailed results will be given in a paper to be published later.

The authors are listed in alphabetic order. HNT proposed the method, a scheme based on chaotic generators and fault current injection in the power supply lines by means of a power source controlled by a chaotic generator, the protection method in Fig. 1 and the chaotic generator. EFI built all the circuits based on the design and schemes provided by HNT, wrote the test program based on the indications of HNT, participated in the experiment and contributed to writing the paper.

The authors declare no conflict of interest. The authors retain the copyright of the paper and will publish later an enlarged, definitive version of this research.

## REFERENCES

Kocher P., Jaffe J., Jun B., *Introduction to Differential Power Analysis and Related Attacks*. Crypt. Res. Inc., www.cryptography.com/public/pdf/DPATechInfo.pdf, accessed Jan. 2012.

Kocher P., Jaffe J., Jun B., *Differential Power Analysis.* Crypt. Res. Inc., www.cryptography.com/public/pdf/DPA.pdf, accessed Jan. 2012.

Teodorescu H.-N.L., *O nouă clasă de circuite haotice bazate pe buclă de reacție capacitivă*. Proc. ICTEI 2010, The 3-rd Int. Conf. Telecomm., Electron. a. Inform., **1**, Chişinău, May 20-22, 2010, 319-325.

Teodorescu H.-N.L., Cojocaru V. P., *Complex Signal Generators Based on Capacitors and on Piezoelectric Loads*. In: C. H. Skiadas, I. Dimotikalis and C. Skiadas (Eds.), *Chaos Theory: Modeling, Simulation and Applications.* World Sci. Publ. Co., Singapore, 2011, 423-430.

## MASCAREA INSTRUCȚIUNILOR UNUI MICROCONTROLER CU AJUTORUL UNEI SURSE “HAOTICE” DE ALIMENTARE

(Rezumat)

Se propune o metodă de protecție pentru microsisteme, bazată pe injectarea de semnal cvasi-aleatoriu pe linia de alimentare. Se foloseşte o sursă de tensiune cu element regulator serie a cărei tensiune de ieşire este comandată de un oscilator haotic prin comanda celor două ramuri. Se urmăreşte ca prin injectarea de zgomot pe linia de alimentare „pattern”-ul pentru instrucțiunile executate de microsistem să fie modificat şi vizual nedetectabil la decodarea prin funcția de autocorelație.