Floating Gate EPROM Instability for True Randomness Generation

Jordan Genoff Genoff

Abstract – **True randomness sources are an essential part in numerous scientific, industrial and real-life applications. Today there are many established approaches to build such a source by exploiting different physical phenomena. This paper presents a new one – a floating gate EPROM device put into unstable behavior by loading an appropriate amount of charge into its floating gate. Implementation issues of the idea and properties of the generated randomness are discussed.**

Keywords – **True Random Stream, Floating Gate EPROM**

I. INTRODUCTION

True random binary source (TRBS) with good statistical properties and (optionally) high output rate is a crucial component of any deterministic digital (e.g. computational) system which requires stochastic interference for its proper operation. Every TRBS fits in the common general scheme, shown in Fig.1:

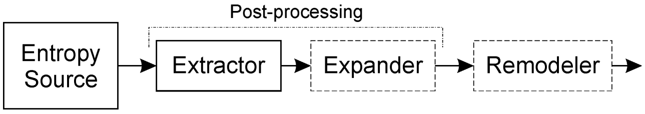

Fig. 1. General TRBS scheme

At the core stands the entropy source, which is some natural process with acceptable properties, concerning stochastic behavior and unpredictability. Unfortunately, the latter two are not presented in as much quantity and quality as desired, according to the requirements of even the simplest statistical verification tests. That is why there is a need of getting the pure randomness out by applying a so called "extracting" procedure, which eliminates redundancy, correlations, and pseudo-randomness, but reduces the output rate proportionally. The extractor output is supposed to pass most of the statistical tests and to show uniform probability distribution for any model.

In order to compensate the loss of rate, sometimes a so called "expanding" procedure follows, which introduces controlled pseudo-randomness in a way that both satisfies the statistical testing and preserves the distribution uniformity for any model. Finally, a "remodeling" algorithm is optionally present, which produces output with a given probability distribution for a given model.

It is obvious that the most important factors with impact on the quality of randomness and on the output rate are located in the entropy source and in the extractor. The worse the entropy source, the more severe must be the extractor. This can lead to several consequences for the produced random stream: it looses its "natural" properties, e.g. rare events are lost in most cases, especially for higher level models; the extractor itself is able to introduce statistical bias; the output rate reduces dramatically. And vice versa, the better the entropy source, the gentler the extractor is allowed to be.

The struggle for devising better entropy sources has been going on for decades. The result is a wide variety of physical concepts: thermal electronic noise, radio-wave noise, atmospheric electricity noise, radioactive decay, photoelectric effects, quantum effects, event handling in computer systems, etc. Each of them and the devices built according to them have their advantages and disadvantages.

This paper presents an idea for true randomness generator, which, to the extent of the exhaustive author's investigation, has not yet been proposed. The sections that follow have sufficiently clear titles: Section II – Motivation Part 1; Section III – Floating Gate Devices; Section IV – Related Work; Section V – Motivation Part 2 and Proposal; Section VI – Experiments and Results.

II. MOTIVATION – PART I

As was formulated above, it is very important for the entropy source to possess as good statistical properties as possible. The motivation for this work goes as follows:

1. Theory implies that a proper mixture of random sources may express better statistical properties concerning randomness, than any of the sources alone.

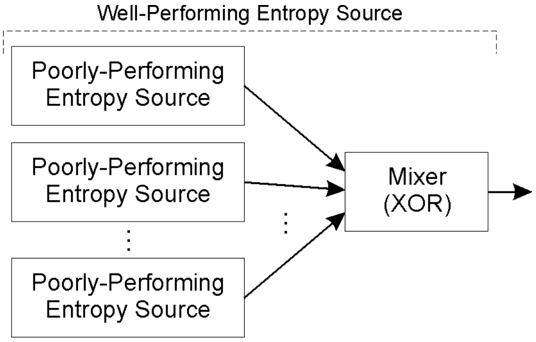
Well-Performing Entropy Source
Poorly-Performing Entropy Source
Poorly-Performing Entropy Source
Poorly-Performing Entropy Source
Mixer (XOR)

Fig. 2. Well-Performing entropy source by XOR mixing of poorly-performing sources

2. One of the most interesting mixing functions for binary streams is the XOR function [1] – it is very simple, so: it lacks a potential to introduce bias or unwelcome pseudo-randomness and it can be easily analyzed. The problem with XOR is that being simple and dealing with relatively bad (poorly performing) sources, it will require a large number of them to achieve relatively good results at its output. Fig.2 shows the concept as it is presented so far.

3. Large number (tens or even hundreds) of distinct independent sources is not an easy requirement to satisfy, even though they may be allowed to perform poorly. This means a large amount of circuitry and expenses.

J. Genoff is with the Department of Computer Systems and Technologies, Faculty of Electronics and Automatics, Technical University – Sofia, branch Plovdiv, 63 Sankt Peterburg Blvd., 4000 Plovdiv, Bulgaria, e-mail: jgenoff@tu-plovdiv.bg

4. A promising solution may be the option of a microelectronic device. This provides an opportunity for hundreds, even thousands of micro-devices – possible entropy sources, placed on a single IC.

Having reached the intention to search for a microelectronic solution, two choices come up naturally:

1. Image sensor matrices: These are used mostly in digital cameras and classify in several distinct types. Each cell utilizes the photoelectric effect in some way and represents a pixel in a 2D image plane. Each cell can perform as an independent entropy source. There exist several popular solutions in practical use. Some of them work by exposing the matrix on a light flux of stochastic or chaotic nature. Others prefer to put the matrix in a complete darkness, thus being able to register the thermal noise in the cells.
2. Floating gate arrays: They are the basis of all digital semiconductor non-volatile writable memories. The array consists of floating gate transistor (FGT) cells, each of which performing as a unit of memory. Each FGT determines the digital contents of its cell as a direct consequence of the charge injected into its floating gate. Because of such a role, the FGT cell is expected to be an extremely stable element, no matter which digital value it represents, or for how long time, or how unfriendly or changing is the environment.

The *core idea discussed in this research* is to use floating gate array as a set of entropy sources by putting each FGT cell in a role exactly opposite to stability – undetermined operation with consequent random output. This is to be achieved by loading an appropriate amount of charge into its floating gate. And this is accomplished by slightly, or not so slightly diverging from the recommended programming and/or erasing procedures.

III. Floating Gate Devices

The following is a review of the floating gate devices and technology, based on [2].

A. Basic Structure

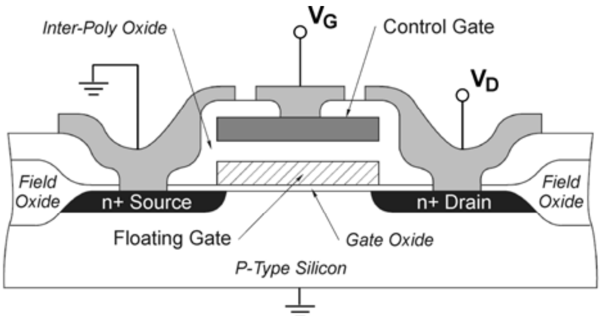

Fig. 3. FGT basic structure

Though since its invention in 1967 FGT has gone through a significant development, its structural design and functional concepts have remained unchanged, Fig.3 – the source-drain conductivity is dependant not only on the control gate voltage, but on a second, “floating” gate, too. It can hold an amount of charge capable to additionally and significantly alter the conductance of the S-D channel. The floating gate charge is trapped, so it will not change through time and by changing environment.

Differences in FGT design concern mostly the way the charge is put into or taken out of the floating gate. Today there are three major types of FGT cell designs and respective types of non-volatile memory devices that emerged chronologically:

– EPROM – (UV) Erasable Programmable ROM. These are the oldest. Each cell consists of a single FGT with a classical design, as in Fig.3. The programming is accomplished through Hot Carrier Injection (HCI), the erasing is done by exposing the device to UV light.

– EEPROM – Electrically Erasable Programmable ROM. These came after EPROM (1976) as a solution to the desire to make EPROM electrically erasable. The substantial differences from EPROM are: each cell consists of two transistors – a floating-gate one and a control-gate one; inversed functional meaning of charge presence into the floating gate; a very much thinner gate oxide layer in the floating gate transistor; Fowler-Nordheim tunneling (FNT) used for both programming and erasing.

– Flash memories – These are the contemporary high volume non-volatile memories. They elaborate from the EEPROM cell (1984), but there are differences in the overall device architecture – cells are interconnected in NAND or NOR structures, hence two types of flash memories; specific operations are performed on page/block access. Some of them use HCI, others use FNT for programming, and all use FNT for erasing.

No matter what is the memory type, no matter exactly what is the internal architecture of the memory device, and how complex is the latter, there is a simple generalized representation of how a single cell fits into the whole device’s structure, Fig.4:

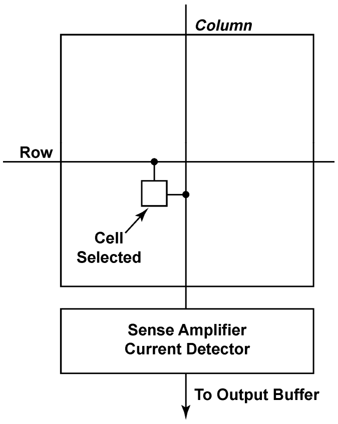

Fig. 4. A single cell in a memory cell array

The important thing here is the current sense amplifier (SA): if the instable behavior of the cell is feasible, then a SA with a well pronounced hysteresis is not welcome.

B. Basic Operation and Programming/Erasing

There are three possible operations with a non-volatile memory cell: reading; programming (writing); erasing (writing). In order to explain what happens when any of these happens, the FGT I-V characteristic should be presented and analyzed, Fig.5:

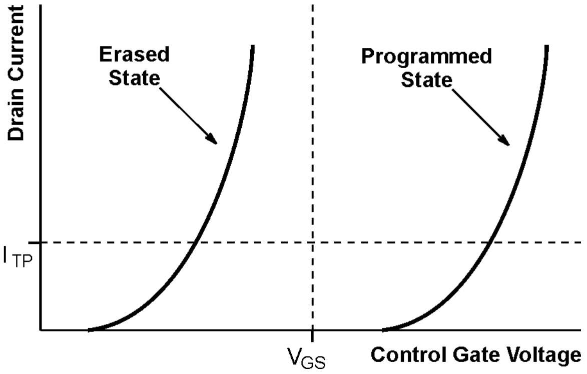

Fig. 5. FGT I-V characteristic

In I-V characteristic terms:

– "Reading" a cell means applying a select threshold voltage V_{GS} on the control gate and registering the drain current by the SA current detector. The binary value stored in the cell is determined by SA according to whether the drain current is above or below the SA trip point I_{TP}.

– "Programming" a cell means, for example, filling the floating gate with charge. This leads to moving the I-V curve where it is on the Fig.5 characteristic. Reading such a cell by applying V_{GS} on the control gate will result in always getting one of the binary values (e.g. "0").

– "Erasing" means the opposite of programming, i.e. for example, emptying the floating gate of charge. This leads to moving the I-V curve where it is on the Fig.5 characteristic. Reading such a cell by applying V_{GS} on the control gate will result in always getting the other one of the binary values (e.g. "1").

IV. Related Work

The core idea of using FGT as a source of entropy is the subject of attention in the last decade. There are existing implementations [3] and they utilize a common basis – NAND flash memory cells are partially programmed in a degree that will lead to instability due to Random Telegraph Noise (RTN) phenomenon. Other disturbing factors, as flicker noise and thermal noise, are welcome too, but with less impact on instability.

Flash memories are the choice, because: they are the modern non-volatile memory type and hence they are present in vast number of devices; RTN is clearly observable due to the extremely small sizes of the flash memory cells; partial programming of a NAND flash memory cell can be done without any additional hardware; contemporary flash memories contain billions of cells.

Above mentioned applications use unstable flash memory cells for two different purposes: randomness generation and device fingerprinting. There are several recent patents pending on these developments [4].

V. Motivation – Part II and Proposal

The *specific idea proposed in this research* is to investigate whether partial programming of EPROM memory cell can derive enough instability for randomness generation. This research is motivated by the demand to find a different from already patented implementations to use FGT devices as an entropy source. Old EPROM memories are chosen because:

– They combine entirely different physical mechanisms for programming and erasing, so they offer more opportunities for diversified attempts to achieve partial programming.

– They are the oldest design, technology and production, and therefore there is an assumption that:

– their FGT cells and overall device architecture are the simplest, with minimized amount of extra circuitry for acceleration and reliability;

– their SAs will exhibit some imperfectness, especially concerning hysteretic properties;

– their programming depends in highest degree on external voltages, so by varying the latter a widest range of programming regimes can be investigated;

The main concerns about using EPROM devices are:

– Which kind of noise – RTN, flicker or thermal will be the dominant one, and hence the randomness properties.

– Whether the FGT cell instability will overcome the SA histeresis, if there is any.

The aims of the research are graphically shown on Fig.6:

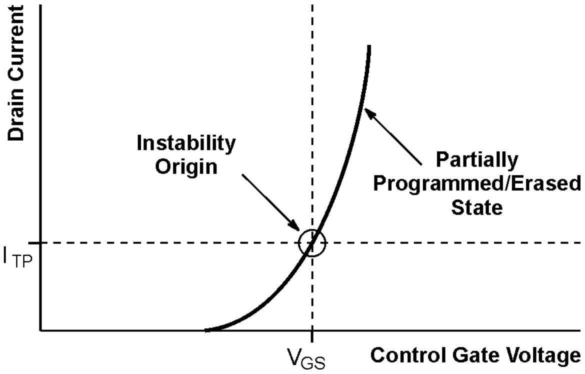

Fig. 6. Specially programmed/erased FGT I-V characteristic

Somehow the I-V curve should be placed where it is on the Fig.6 characteristic. The belief is that: this will make the threshold voltage fluctuations caused by various noises critical; reading such a cell will get a drain current as near the SA trip point as possible, and SA itself will produce unstable output.

Placing the curve at this position is possible in two ways:

– by partially programming a fully erased cell;

– by partially erasing a fully programmed cell.

VI. Experiments and Results

A. Experimental Setup

The following EPROM memory chips are put to test:

TABLE 1. EPROM CHIPS IN TEST

Chip	Manufacturer	Production
MBM2716	Fujitsu Component Ltd.	1982
AM2732	AMD	1980
M27C256	ST	1988
TMS27C512	TI	1992
M27C512	ST	1995
AM27C040	AMD	1990
M27C1001	ST	1994

An entropy measure must be established. Let a single FGT cell be sampled by reading S times during time interval of length τ. Let $S^{[0]}$ times a "0" be read and $S^{[1]}$ times a "1" be read. Then

$$\hat{p}^{[0]} = S^{[0]} / S \quad , \quad \hat{p}^{[1]} = S^{[1]} / S \tag{1}$$

are assessments of the probability to read "0" and "1" respectively. The Shannon entropy measure is used:

$$\hat{H} = -(\hat{p}^{[0]} \log_2 \hat{p}^{[0]} + \hat{p}^{[1]} \log_2 \hat{p}^{[1]}) \tag{2}$$

The overall entropy for N FGT cells is the mean (ME)

$$\overline{\hat{H}} = N^{-1} \sum_{\forall cell}^{N} \hat{H} \tag{3}$$

Two distinct types of tests are conducted:

1. *Partial erasing of fully programmed FGT cells.* A special EPROM reader was designed for the purpose of these tests. It allows an EPROM chip to be read under computer control very frequently and for the time while it is being erased by UV light. First, the chip is fully programmed with "0"s by a conventional programmer. Then it is put to special reading under UV exposure for a time period t_x little longer, than t_e recommended by the manufacturer as sufficient for full erasure. While erasing goes on, the cells in test are intensively read and the data is accumulated in the computer memory. After completion the whole time period is split in intervals of length τ. The ME for each interval is calculated and the sequence of MEs for the whole period can be presented graphically. These experiments can be interleaved by various UV filters.
2. *Partial programming of fully erased FGT cells.* A special EPROM programmer was designed for the purpose of these tests. It allows an EPROM chip to be programmed under computer control by varying the width and number of programming pulses, as well as the programming and power supply voltages. First, the chip is fully erased, resulting in "1" in every cell. Then it is put to special programming and various regimes are tested. All of them have two common features: much shorter and much more in number programming pulses for each cell, than recommended by the manufacturer; and the resulting aggregate energy E_x of programming pulses is a little bit more than E_p inferred from the recommended regime for full programming. After every R pulses, the cells in test are read S times and the ME is calculated. The sequence of MEs corresponding to the sequence of R-pulse packets can be presented graphically. These experiments can be interleaved by programming and power supply voltages.

B.1 Results of partial erasing experiments

As explained in section IV.A, partial erasing tests have 7 degrees of freedom: the chip in test, t_e, t_x, N, S, τ, and UV filtering. Each test produces a sequence of T_x / τ values for ME, which is suitable to be represented as a curve in a 2D coordinate system. In order to minimize the degrees of freedom and to make the representation more comprehensible, the following assumptions are set up: entropy is on the vertical axis; the horizontal axis shows time in a somewhat normalized manner – whatever the chip in test is, its t_e is of constant size on the axis; throughout all experiments N, S and τ do no vary; many repetitive experiments with one and the same chip with the same UV shield are averaged and represented as a single curve.

Here are the results of over 60 tests:

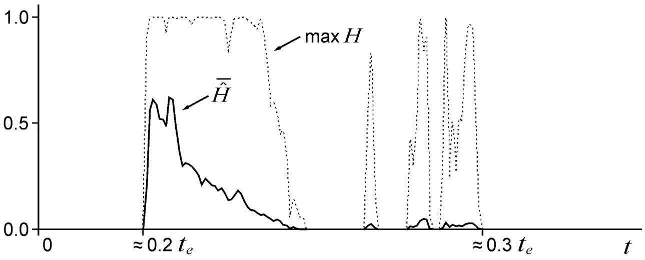

Fig. 7. Partial erasing results

B.2 Results of partial programming experiments

The same reasoning applies to the partial programming tests, except that the degrees of freedom are: the chip in test, E_p, E_x, N, S, R, and programming and power supply voltages. Respectively: entropy is on the vertical axis; the horizontal axis shows the accumulated pulse energy, normalized to E_p; N, S, and R do not vary; repetitive experiments with one and the same chip and the same programming and power supply voltages are averaged.

Here are the results of over 60 tests:

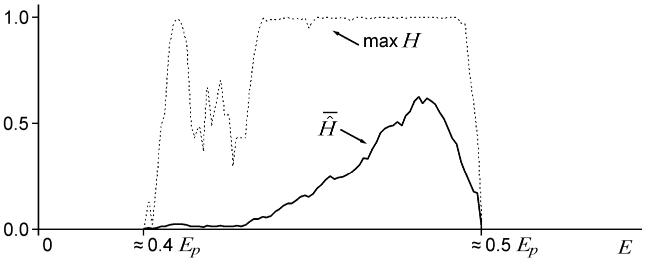

Fig. 8. Partial programming results

VII. Conclusions

According to Fig.7 and Fig.8, the EPROM memory devices from Tbl.1 are capable of generating ME above 0.5 bits, which is far enough acceptable, given the relatively large number of independent entropy sources in a single IC.

Future work plans are: combining partial erasing and programming; and designing more sophisticated protocols for partial programming.

References

[1] Davies R., Exclusive OR (XOR) and hardware random number generators, Technical Report, 2002

[2] Pavan P., Larcher L., Marmiroli A., Floating Gate Devices: Operation and Compact Modeling, Kluwer Academic Publishers, 2004, ISBN: 1-4020-7731-9

[3] Wang Y., Yu W., Wu Sh. , Malysa G., Suh G.E., Kan E.C., Flash Memory for Ubiquitous Hardware Security Functions: True Random Number Generation and Device Fingerprints, 2012 IEEE Symposium on Security and Privacy, 2012

[4] WANG Y., YU W., Kan E.C., SUH G.E., Methods and systems for providing hardware security functions using flash memories, WO Patent App. PCT/US2013/041,615, 2013