Generation of Nonlinear Feedback Shift Registers with Special-Purpose Hardware

Janusz Szmidt

Military Communication Institute
Zegrze, Poland

WAIFI 2012, Bochum

NLFSRs - Nonlinear Feedback Shift Registers

- Let $\mathbb{F}_2 = \{0, 1\}$ denote the binary field and $\mathbb{F}_2^n$ the vector space of binary n-tuples.
- A binary n-stage Feedback Shift Register (FSR) is a mapping

$$\mathfrak{F} : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^n$$

of the form

$$\mathfrak{F} : (x_0, x_1, \ldots, x_{n-1}) \longmapsto (x_1, x_2, \ldots, x_{n-1}, f(x_0, x_1, \ldots, x_{n-1}))$$

where the *feedback function* f is a Boolean function on n-variables.

- The FSR is called *non-singular* if the mapping $\mathfrak{F}$ is one-to-one, i.e., $\mathfrak{F}$ is a bijection on $\mathbb{F}_2^n$.

NLFSRs - Nonlinear Feedback Shift Registers

- It was proved that the FSR is non-singular iff its feedback function has the form

$$f(x_0, x_1, \ldots, x_{n-1}) = x_0 + g(x_1, \ldots, x_{n-1}) \tag{1}$$

  where g is a Boolean function on $n-1$ variables.
- The FSR is called linear (LFSR) if the feedback function f is linear one and nonlinear (NLFSR) if the function f is nonlinear; i.e., the function f has higher order terms in its Algebraic Normal Form (ANF).
- Further, we will consider only nonsingular and nonlinear feedback shift registers.

NLFSRs - Nonlinear Feedback Shift Registers

- Consider a binary sequence $\mathbf{s} = (s_i)_{i=0}^{\infty}$ whose first n terms $s_0, s_1, \ldots, s_{n-1}$ are given and whose remaining terms are uniquely determined by the recurrence relation

$$s_{i+n} = f(s_i, s_{i+1}, \ldots, s_{i+n-1}) \quad \textit{for all} \quad i \geqslant 0. \tag{2}$$

- We call **s** an output sequence of the feedback shift register given by (1). The binary n-tuple $(s_0, s_1, \ldots, s_{n-1})$ is called the *initial state vector* of the sequence **s** or the *initial state* of the feedback shift register.
- The recurrence relation (2) can be implemented in hardware as a special electronic switching circuit consisting of n memory cells which is controlled by an external clock to generate the sequence **s** .

NLFSRs - Nonlinear Feedback Shift Registers

The FSR generates a binary sequence in a usual way as the electronic device clocked by an external clock

where $(s_0, s_1, \ldots, s_{n-1})$ are the initial values of the register and f is the feedback function.

De Bruijn sequences

- **Definition 1.** The de Bruijn sequence of order n $(a_0, \ldots, a_{2^n-1})$ of elements from the binary field $\mathbb{F}_2$ is a sequence of period 2^n in which all different n-tuples appear exactly once.
- It was proved by Flye Sainte-Marie in 1894 and independently by de Bruijn in 1946 that the number of cyclically inequivalent sequences satisfying the Definition 1 is equal to

$$B_n = 2^{2^{n-1}-n}. \tag{3}$$

- **Definition 2.** The modified de Bruijn sequence of order n $(a_0, \ldots, a_{2^n-2})$ is a sequence of period $2^n - 1$ obtained from the de Bruijn sequence of order n by removing one zero from the tuple of n consecutive zeros.

Nicolaas Govert de Bruijn, Dutch mathematician, 9 July 1918 - 17 February 2012

Oberwolfach, 1960

Solomon Golomb and Guang Gong, SETA 2012

Books:

- S. W. Golomb. *Shift Register Sequences.* San Francisco, Holden-Day, 1967, revised edition, Laguna Hills, CA, Aegean Park Press, 1982.
- S. W. Golomb, G. Gong. *Signal Design for Good Correlation. For Wireless Communication, Cryptography, and Radar.* Cambridge University Press, 2005.
- A. Klapper, M. Goresky, *Algebraic Shift Register Sequences*. Cambridge University Press, 2012.

De Bruijn sequences

- In 1990 Mayhew and Golomb investigated sequences satisfying the Definition 2 and their linear complexity. Recently, they were also investigated by Ferruh Ozbudak *et. al.*
- These sequences were called by Gammel, Goetffert and Kniffler (in their project on Achterbahn stream cipher) the *primitive sequences*.
- In the case of linear feedback shift registers these sequences are generated by primitive polynomials and their theory is understood quite well.
- The primitive sequences are very important in cryptographic applications since:

Primitive (span n) sequences

- They exist. There are B_n primitive sequences altogether (the linear and nonlinear ones). The number of primitive LFSRs is equal to

$$\frac{\varphi(2^n - 1)}{n},$$

  where φ denotes the Euler phi function, hence there are much more NLFSRs than LFSRs.
- The primitive sequences have good statistical properties.
- The linear complexity of a NLFSR (the order of a LFSR generating the same sequence) is on average bigger than 2^{n-1} and many of them have the most possible linear complexity equal to $2^n - 2$.
- Let us recall that the linear complexity of a primitive LFSR of order n is just equal to n.

Primitive (span n) sequences

- There are primitive NLFSRs for which the Algebraic Normal Form of the Boolean function F in formula (1) is quite simple; it has low algebraic degree and a possibly small number of terms.
- Since there are $2^{2^{n-1}}$ different Boolean functions on $n-1$ variables, hence the probability that a randomly chosen function of the form (1) is a primitive NLFSR is equal to

$$\frac{2^{2^{n-1}-n}}{2^{2^{n-1}}} = \frac{1}{2^n}$$

- In fact, this probability is equal to $\frac{1}{2^{n-3}}$ according to some properties of the involved feedback functions.

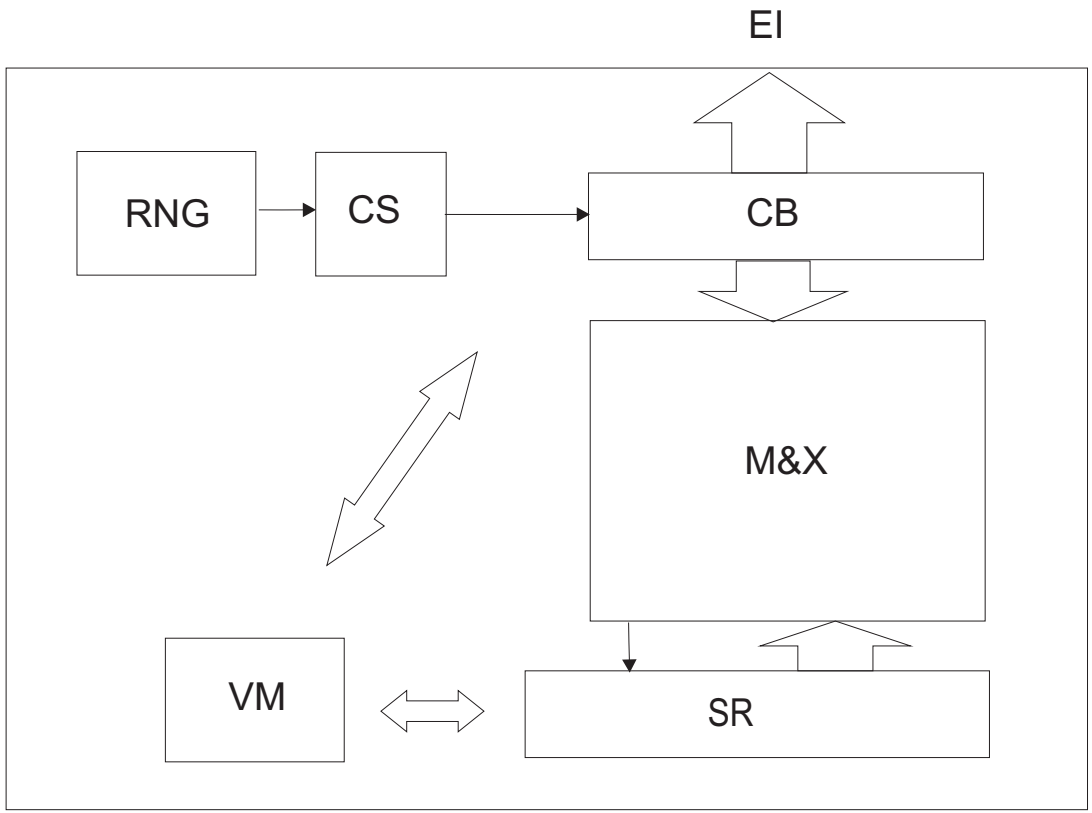
Primitive (span n) sequences

- The task is to find primitive NLFSRs with a possibly simple algebraic form and this is much more difficult.
- A method how to construct such primitive NLFSRs is not known and we have to search for them.
- Gammel *et al.* found simple primitive NLFSRs up to the order 33 and they used them in the design of the stream cipher Achterbahn, but neither the method of searching nor the average time needed to find such good NLFSRs have been revealed.

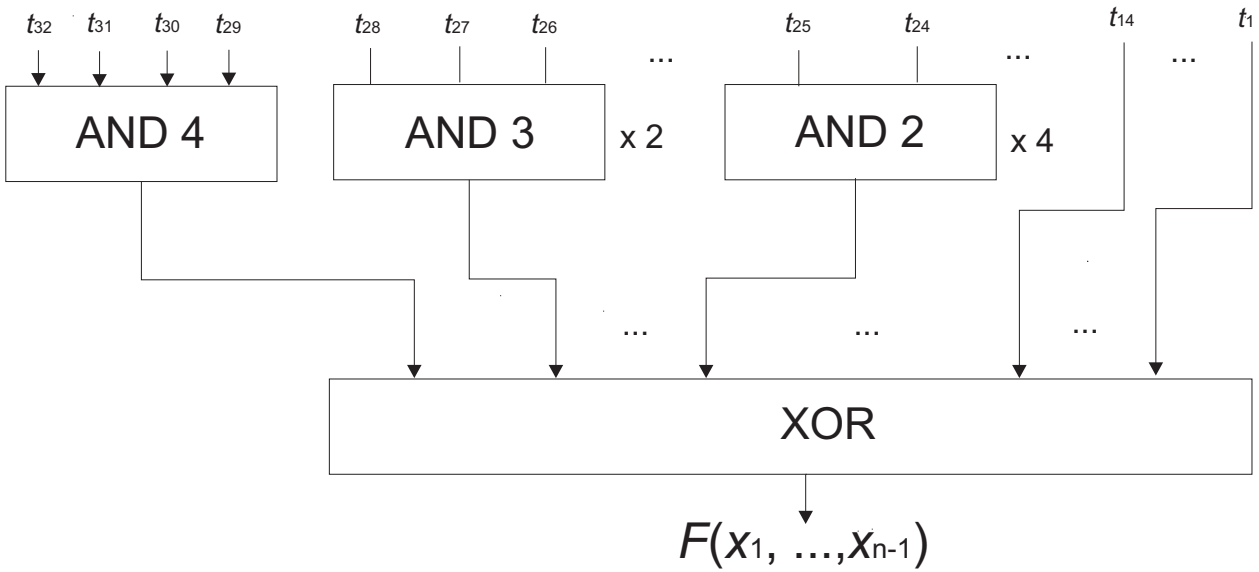
FPGA implementation

- We implemented an algorithm for searching nonlinear feedback shift registers of order n having maximum period $2^n - 1$ using hardware devices from our previous projects.
- They were equipped with Altera EP3C80 Field Programmable Field Arrays.
- We used Altera Quartus II v.9.0. design software to simulate and compile the current project.

A single module of the searching machine



The structure used to generate NLFSR
The multiplexers and XOR block



A single module of the searching machine

The random NLFSR searching module (RNSM) components are:

- The random number generator (RNG)
- The coefficients selector (CS)
- The coefficients buffer (CB)
- The multiplexers and XOR block (M&X)
- The shift register (SR)
- The verification machine (VM)

A single module of the searching machine

- Random numbers are taken from the RNG. Coefficients are downloaded byte by byte into the CS, where their values and repetitions are controlled.
- Then the bytes go to the CB, whose task is to store combinations of coefficients during the test. The multiplexers define the feedback function of NLFSR according to the data buffered in the CB. Their outputs are connected to the XOR gate.
- Next, the output of the XOR function feeds the SR. The SR is set with a seed value at the beginning of a searching process by the VM and it starts to shift. After the first repetition of the seed the test is finished.
- A positive result is sent to the Ethernet Interface (EI), which is the same for all implemented modules. A negative result starts a new process of random generation and testing.

Examples of NLFSRs of order 25 and 27

- During our search of NLFSRs of order 25 nad 27 with maximal periods the 128 RNSM were implemented in four physical devices.
- The 32 modules implemented in a single device worked and stored results independently. The four devices were connected to a hub and a personal computer (PC) with the wireshark sniffer.
- The FPGA was clocked with 65.536 MHz, although the maximal possible clocking is 128 MHz.
- The average time to find one NLFSR of order 25 was 4 hours and the average time to find one NLFSR of order 27 was 21 hours, respectively.

Examples of NLFSRs of order 25 and 27

order 25 (the linear complexity is $2^{25} - 2$) :

$$x_0 + x_8 + x_9 + x_{10} + x_{11} + x_{19} + x_{20} + x_{21} + x_{23} + x_6x_{21} + x_{10}x_{14} +$$

$$x_{12}x_{20} + x_{19}x_{20} + x_4x_{18}x_{21} + x_{11}x_{18}x_{22} + x_1x_5x_7x_{23}$$

order 27 (the linear complexity is not measured) :

$$x_0 + x_4 + x_8 + x_9 + x_{11} + x_{12} + x_{15} + x_{16} + x_{23} + x_{12}x_{22} + x_{13}x_{23} +$$

$$x_{13}x_{25} + x_{22}x_{23} + x_7x_8x_{24} + x_{12}x_{14}x_{26} + x_6x_{11}x_{19}x_{22}$$

Preprint:

Generation of Nonlinear Feedback Shift Registers with Special-Purpose Hardware

Tomasz Rachwalik, Janusz Szmidt

Robert Wicik, Janusz Zabłocki

Cryptology ePrint Archive, 2012/314, www.iacr.org

Thank you