International Journal of Electrical, Electronics and Computer Engineering **1**(2): 55-59(2012)
ISSN No. (Online) : 2277-2626
Special Edition for Best Papers of Michael Faraday IET India Summit-2012, MFIIS-12

ARM7 Microcontroller Based Digital PRBS Generator

S. Mondal, A. D. Barman, and A.K. Datta

** Academy of Technology, West Bengal, India*
***Institute of Radio Physics and Electronics, West Bengal, India,*
****BOM Charitable Trust, Kolkata, India,*

(Received 15 October, 2012 Accepted 01 December, 2012)

ABSTRACT: This paper reports a Pseudo Random Binary Sequence (PRBS) generation by an ARM7 Microcontroller. It generates 32 bit shift register equivalent pseudo noise sequence at variable bit rate with a maximum 2.25 Mbps for 60 MHz clock frequency of the microcontroller. The maximum bit rate can be further increased using high frequency microcontroller. Keil µversion-4 compiler has been used for programming of ARM7 microcontroller. Quality of PRBS bits are measured in terms of rise time, fall time, amplitude jitter and these values are found to be within satisfactory limits.

Index Terms— ARM7 Microcontroller, Autocorrelation, PRBS, Shift Register.

I. INTRODUCTION

Pseudo Random Bit Sequence is widely used for performance analysis in electronics applications, digital data encryption and digital communication techniques etc. The pseudo random sequence is defined as a random coded sequence of ones and zeros with a periodicity of random data and associated with the randomness properties like Balance, Run and Correlation property. There are few techniques of generating PRBS. The conventional way of developing PRBS is Linear Feedback Shift Register (LFSR) array. This LFSR technique [1,2] based on polynomial operation under modulo-2 arithmetic [3]. The length of cascading stages of shift register associated with the length of PRBS sequence. FPGA, Digital Signal Processor (DSP) based approaches [4,5][6] are reconfigurable just by changing programming but these are little expensive. In FPGA based design [2] first true random number is generated and then it produces pseudo random number, which reduces the throughput of the design to 225 bps [2]. A VHDL based synthesized hardware [7] simulates the results with a 2.81 fold increase in throughput compared to non pipelined architecture.

Here we have presented a relatively inexpensive reconfigurable technique of PRBS generation based on memory and ARM7 microcontroller. This is an inexpensive method to generate a 32 bit PRBS with a specified bit rate and quality obtained through professional Bit Error Ratio Testing Instruments [8]. Reconfigurable means that by programming one can change the PRBS sequence. We demonstrated that a 60 MHz ARM7 microcontroller can produce a PRBS sequence 2.25 Mbps with an initial delay of 95 µsec for 4 bit equivalent shift register sequence. After this initial delay PRBS sequence generation will be in real time with the specific bit rate.

A microcontroller generates the sequence depending upon the user given length. The desired sequence is loaded in an external memory. This stand alone system is able to generate PRBS for a given user length. The programming for ARM7 Microcontroller LPC2148 has been developed in Keil µversion4 platform. The real time PRBS data output is stored and processed offline by MATLAB to evaluate bit eye-pattern and autocorrelation of PRBS data.

II. METHODOLOGY

A. PRBS Generation

The basic method of generation of PRBS employs LFSR under modulo-2 arithmetic [3,9]. The shift register sequence $\{a_n\} = \{a_0, a_1, a_2, \ldots\ldots\ldots\}$ describing the history of the first flip flop is shown in Fig.1. Generating function G(x) holds all the flip flop data in polynomial form as shown below.

$$G(x) = \sum_{n=0}^{\infty} a_n x^n \tag{1}$$

$$a_n = \sum_{i=1}^{r} c_i a_{n-i} \tag{2}$$

Where a_n satisfy the recursive relation with the r number of flip flop stages and n is the instant.

$$G(x) = \frac{g(x)}{f(x)} \tag{3}$$

Where

$$g(x) = \sum_{i=1}^{r} c_i x^i [\, a_{-i}x^{-i} + a_{-i+1}x^{-i+1} + \cdots + a_{-1}x^{-1}]$$

And rth order characteristic polynomial is as follows

$$f(x) = [1 - \sum_{i=1}^{r} c_i x^i]$$

The Generating function entirely expressed in terms of the initial conditions $\{a_{-1}, a_{-2}, \dots a_{-r}\}$ and the feedback coefficients $c_1, c_2, \dots c_r$. When initial condition is $a_{-1} = a_{-2} = a_{-3} = \dots = 0$ and $a_{-r} = 1$, the period of the sequence can be found as p for which the characteristic polynomial $f(x)$ divides $(1 - x^p)$ under modulo-2 arithmetic. The maximum length of a LFSR array of r flip flop stages is $p = (2^r - 1)$. The recursive relation for a_n defines the feedback connection required for any valid pseudo noise sequence.

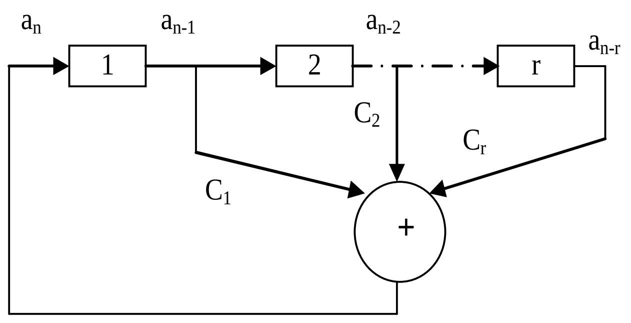


Fig. 1. The LFSR Structure

B. ARM7 Microcontroller Software Algorithm

ARM7 microcontroller [10] has been used to implement the algorithm of PRBS generation. Each of the 32 bit data field represents 32 flip flops of a shift register. According to the user data the number of bits of a data field will be selected and the mask data will be one of the preloaded data stored in microcontroller memory according to the recursive relation of a_n as given by Eqn-(2). Table-1 shows the mask data format, corresponding to any shift register length, mask data is saved in memory. For 4 bit shift register the 32 bit mask data is 0x0000000C. The first seed data will be any nonzero value having the same bits of the data field. The feedback binary digit can be calculated by logical XOR operation of the data bits according to the mask data field which is then shifted to find the parity bit. For r bit data field (2^r-1) nonzero patterns can be produced, which are to be loaded in successive address locations of external RAM by the microcontroller. Then microcontroller selects the starting address of RAM and update address location with a proper delay to adjust the bit rate of PRBS. The Fig. 2 describes the software flow diagram of ARM7.

The main program involves the following software libraries:

- Generation of PRBS
- LCD interfacing through parallel mode
- PC keyboard interfacing using I^2C mode
- External RAM interfacing

Example:
Number of flip flops, r=4

Step-1: The 32 bit mask data is MDATA = 0x0000000C.
Step-2: Let the initial seed value of register
VAL=0x00000001.
Step-3: After each iteration the new data pattern is as follows.

VAL = [(VAL<<1) AND (0xFFFFFFFE)] OR [0x0000000P]
Where COUNT = NUMBER OF ONES IN [(VAL) AND (MDATA)] and

$$P = \begin{cases} 0, & \text{COUNT IS EVEN} \\ 1, & \text{COUNT IS ODD} \end{cases}$$

The total iteration required is (2^r-2). Table-2 shows each iteration output. Each iteration requires $[(r-1)^2+1]$ times bit shifting operation and $[r+1]$ times logical AND operation.

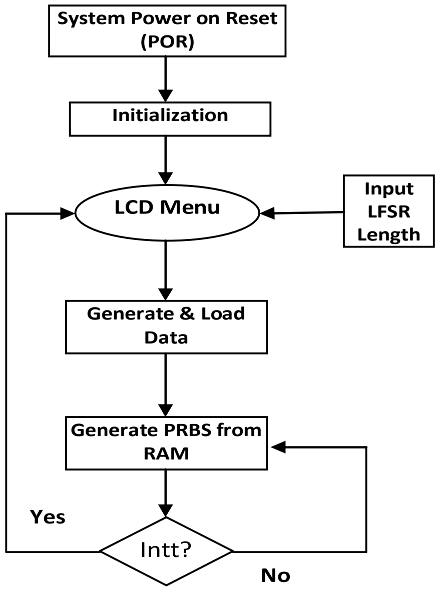


Fig. 2. The Flow Diagram of Software.

Table-1 Mask Data Format.

Length of Shift Register	Bit numbers in mask data contains 1's (considering lsb index is 0)
2	0,1
3	0,3
4	2,3
5	2,4
6	4,5
7	5,6
8	1,2,3,7
9	4,8
10	6,9
11	1,10
12	1,9,10,11
13	0,10,11,12
14	1,11,12,13
15	13,14
16	10,12,13,15

C. System Description

We have presented a stand-alone system for generation of PRBS using LPC2148 ARM7 microcontroller. Fig.3 describes the schematic of the system. PS/2 keyboard is used to provide input data from user end via I^2C protocol. A shift register (IC-74164, serial in parallel out) arrangement is used to collect the serial data of keyboard then it fed to a parallel 8 bit to I^2C converter IC-PCF8574, which will interrupt the microcontroller for input data. Microcontroller will develop and load pseudo noise pattern to external RAM according to the user specified bit length. Fig.4 describes the connection between LPC2148 and RAM. The whole process status is displayed in a 128x64 LCD. Individual data pin-out of RAM (D_0, D_1,…..D_{r-1}) generates PRBS.

In this proposed system microcontroller uses a look up table for storing the recursive relation of a_n as mask data. User can select the number of stages (r) of flip flop that can map to a particular mask data from look up table. The initial default value of the LFSR can be taken as any one of the combination of r bits except decimal equivalent zero. This default value is initial data for the algorithm; microcontroller will develop the successive data sequence. These data are loaded to the external memory system according to increasing order of address. The memory is contained with a valid sequence of PRBS in any bit location of each memory address. The controlling signals for RAM are write enable and output enable are used during the data loading and generation of PRBS respectively. At the end of data loading to memory system, microcontroller starts to select the address of memory with a proper delay according to the bit rate. Whenever the user try to change the order of flip flop stage or bit rate of PRBS using keyboard interrupt process, microcontroller adjust the proper data sequence and delay.

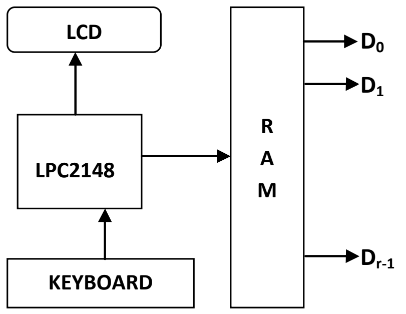


Fig. 3. The schematic of stand-alone PRBS Simulator.

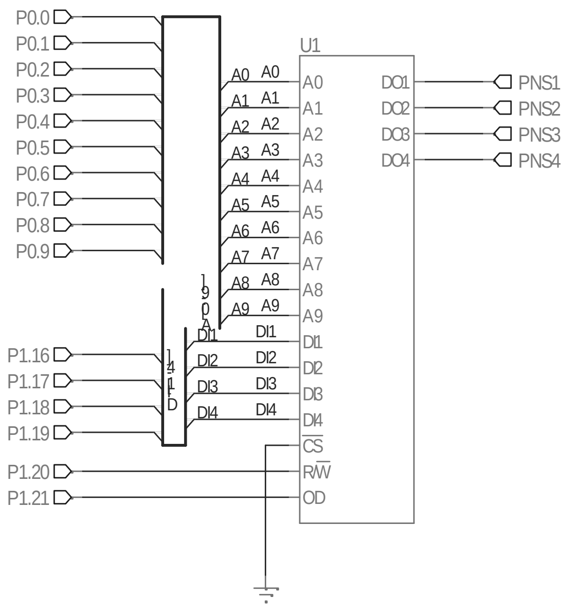


Fig. 4. The schematic of 1024 x 4 bit RAM Interfacing using LPC2148.

III. MEASUREMENTS AND RESULTS

The measurement setup is shown in Fig.5. The proposed system generates PRBS with 2.25 Mbps bit rate. Output bit sequences are stored in a Digital Storage Oscilloscope (DSO) and the sampled data is processed offline by MATLAB to plot the eye-diagram and autocorrelation of the bit sequence. The output sequence satisfies Run and Balance property [3,11] as observed from PRBS data in a DSO.

From the stored DSO data, eye-pattern is plotted as shown in Fig.6(a) which can be used to find the amplitude jitter. The autocorrelation function in Fig.6(b) shows a plot for a quasi periodic signal; the average period value of this signal is the fundamental period of PRBS sequence, which is used to calculate the bit rate of PRBS.

Rise time and fall time of the bit are measured from DSO as shown in Fig.6(c) and Fig.6(d) respectively. Rise time and Fall time is defined as time taken to change the data level form 10% to 90%. Table-3 summarizes the measured values. Our measured values are commensurate with CCSDS recommendation [12]. The system computational delay is also measured and found to be 95 µs. The system computational delay is defined as the time taken by ARM7 microcontroller to generate the total pseudo noise pattern and then stored it to an external RAM. We have presented here the result for 4 bit shift register with DSO sampling rate 65 GSamp/sec with 4 Kbyte storage capacities. However the result can be shown for maximum of 32 bits shift register with higher storage capacity of DSO.

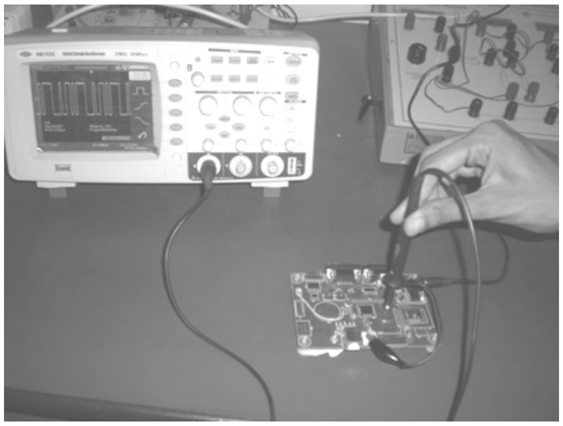


Fig. 5. The Measurement Setup.

Table 3: Measured Values for 4 bit shift register.

Bit Rate	2.25 Mbps
Amplitude Jitter	0.16 volt (4.8% of peak to peak)
Rise Time	13 ns
Fall Time	16 ns
System Computational delay	95 µs

IV. CONCLUSION

In this work we developed PRBS prototype generator with the help of ARM7 microcontroller. The scheme is very effective in generating variable length PRBS with controllable bit rate at very low cost. Quality of PRBS bits are measured in terms of rise time, fall time, amplitude jitter and these values are found to be within satisfactory limits. Though the results are presented in terms of 4 bit shift register, results can easily be extended for 32 bit shift register with the same bit quality. However, to increase the length of the sequence the memory size of the external RAM need to be large. The bit rate can be varied up to a range of 2.25 Mbps with maximum clock frequency 60 MHz of LPC2148. To obtain the higher bit rate of PRBS generation the same algorithm can be used in higher operating frequency microcontroller.

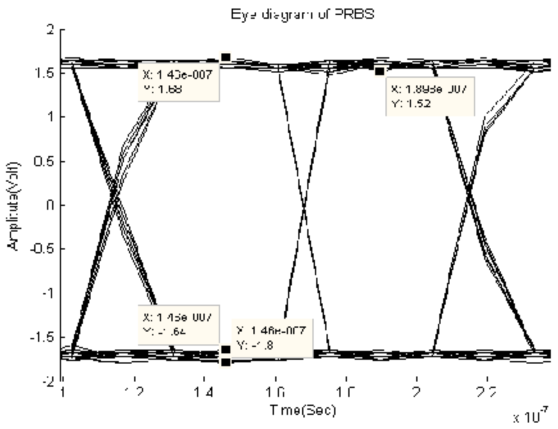


Fig. 6(a). Eye-pattern.

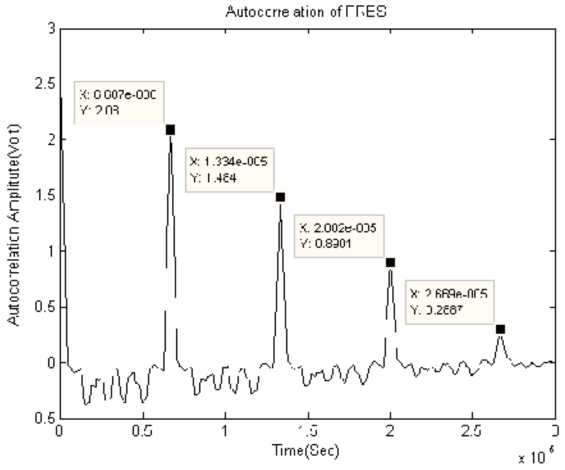
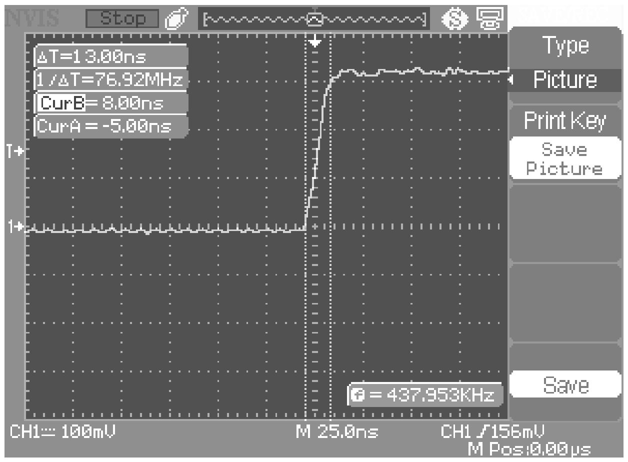


Fig. 6(b). Autocorrelation.



ΔT=13.00ns
1/ΔT=76.92MHz
CurB=8.00ns
CurA=-5.00ns
= 437.953KHz
CH1 100mV M 25.0ns CH1 156mV
M Pos:0.00µs

Fig. 6(c). Rise Time.

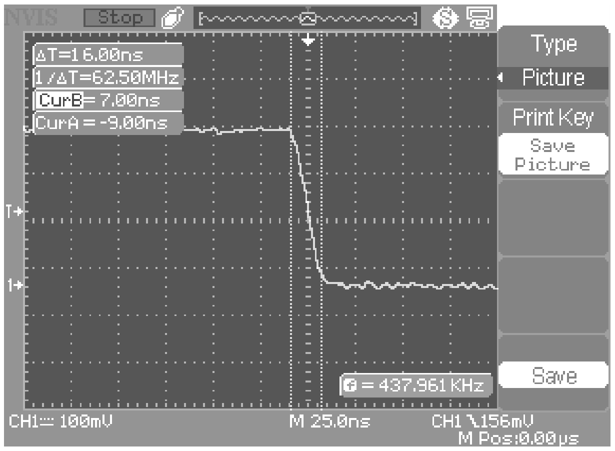


Fig. 6(d). Fall Time.

REFERENCES

[1] Horan. D, Guinee. R,“A Novel Stream Cipher for Cryptographic Applications”,Military Communications Conference MILCOM, *IEEE 2006*, pp. 1-5.

[2] Nur A. Touba and Edward J. McCluskey, “Altering A Pseudo-Random Bit Sequence For Scan-Based Bist”, Test Conference, 1996. *Proceedings., International,IEEE*, pp. 167-175.

[3] S. W. Golomb, Shift Register Sequences, Laguna Hills, CA, 1982.

[4] Tsoi. K.H, Leung K.H, Leong P.H.W,” Compact FPGA-based True and Pseudo Random Number Generators”,*Proceedings of the 11th Annual IEEE Symposium on Field-Programmable Custom Computing Machines* (FCCM’03), pp. 51-61.

[5] Schellekens. D, Preneel.B, Verbauwhede.I,” FPGA Vendor Agnostic True Random Number Generator”, *IEEE International Conference on Field Programmable Logic and Application*, 2006, pp. 1-6.

[6] Qianying Guo, Guangyi Wang- ‘Generation of a Chaos-based PN sequence and its quality Analysis,’ IEEE.

[7] Raj S. Katti and Sudarshan K. Srinivasan,” Efficient Hardware Implementation of a new Pseudo-random Bit Sequence Generator”, *Circuits and Systems,ISCAS 2009. IEEE International Symposium, 2009,* pp. 1393 – 1396.

[8] http://www.home.agilent.com/en/pc-1000000193%3Aepsg%3Apgr/bit-error-ratio-test-bert-solutions?&cc=IN&lc=eng

[9] Sundararajan Sriram, and Vijay Sundararajan,” Efficient Pseudo-Noise Sequence Generation for Spread Spectrum Applications”, *IEEE Workshop on Signal Processing Systems* (SPIS'02), pp. 80-86.

[10] Amol D. Tupkar, Prof. U.A. Rane,”Arm Microcontroller Implementation of Des Using Concept with Time-Variable Key”, *International Journal of advancement in electronics and computer engineering (IJAECE)* Volume **1**, Issue 2, May 2012, pp.62-68.

[11] http://csrc.nist.gov/groups/ST/toolkit/rng/documentation_software.html

[12] http://public.ccsds.org/publications/archive/415x1b1.pdf