Efficient Implementation of PN Sequence Generator Using Vedic Mathematics

[1]Nilima Patle, [2]Rahul Nawkhare

[1]Research Scholar, [2]Assistant Professor
Department of Electronics Engineering
Wainganga College of Engineering & Management,
Nagpur University, India

***ABSTRACT*: In the wireless communication system security and content privacy of user data is of prime importance. In military application to provided security to the message signal and to overcome the interference problem, we develop the use of pseudo- noise sequence (PN) for spreading digital data in DS-SS (Direct Sequence Spread Spectrum) . PN belongs to the class of Linear Feedback Shift Register (LFSR). In this paper the generation of PN sequence is done using multiplier. In order to reduce processing delay and to consume memory storage during sequence generation we adopted the concept of Vedic multiplier. The whole Vedic mathematics uses 16 sutras (formulas). Among the various method of multiplication Uardhava tiryakbhyam (vertically and crosswise) is discussed in details. It enables the parallel generation of partial products and eliminates unwanted multiplication steps. The modeled is proposed using VHDL and synthesis is done using Xilinx ISE 13.1/13.2 simulator**

***Keywords*: pseudo-noise sequence generator, LFSR, digital communication, spread spectrum, Uardhava tiryakbhyam, Vedic Mathematics, VHDL.**

1. Introduction

Spread spectrum techniques for digital communication were originally developed for military applications because of their high security and their susceptibility to interference from other parties .In order to spread the bandwidth of the transmitting signals, the binary pseudo-noise (PN) sequences have been used extensively in spread spectrum (SS) communication systems. One of the most commonly used PN sequences in DS-SS is maximal length sequences (m-sequences) . The length of m-sequences depends on the number of shift registers. Generation of pn sequence is done using multiplier. As multiplication require substantially more hardware resource and processing time than addition and subtraction.. Multiplication using vedic method is one of the fastest and low power multiplier. Vedic mathematics is the ancient methodology .It has 16 sutras . Among the various method of multiplication Uardhava tiryakbhyam (vertically and crosswise) is discussed in details. It enables the parallel generation of partial products and eliminates unwanted multiplication steps. On account of this formulae, the partial product and sums are generated in one step thus eliminates multiple unwanted step. This reduction makes system efficient in speed, ,area and also reduce memory storage.

2. Vedic Mathematics

The word "Vedic" is derived from the word **"Veda"** which means the store-house of all knowledge. Vedic mathematics is mainly based on 16 Sutras (or aphorisms) dealing with various branches of mathematics like arithmetic, algebra, geometry etc. Multiplication methods are extensively discussed in Vedic mathematics. Various tricks and short cuts are suggested by VM to optimize the process. Various methods of multiplication proposed in VM are:-

a) UrdhvaTiryagBhyam - vertically and crosswise
b) Nikhilam navatashcharamam Dashatah: All from nine and last from ten
c) Anurupyena: Proportionately Vinculum

URDHVA TIRYAGBHYAM

Urdhva tiryakbhyam Sutra is a general multiplication formula applicable to all cases of multiplication. It literally means "Vertically and Crosswise. Here, "Urdhva-Tiryagbhyam" (Vertically and Crosswise) sutra is used to propose such architecture for the multiplication of two binary numbers. The beauty of Vedic multiplier is that partial product generation and additions are done concurrently. Hence, it is well adapted to parallel processing. The feature makes it more attractive for binary multiplications. This in turn reduces delay, which is the primary motivation behind this work.

The Proposed Vedic Multiplier Architecture

The hardware architecture of 2x2, 4x4 and 8x8 bits Vedic multiplier module are displayed in the below sections.

2.1 **Vedic Multiplier for 2x2 bit Module**

The design starts with multiplier design that is 2x2 bit multiplier. This is the basic of all multiplier that is NxN bit multiplier Line diagram for multiplication of two 2-bit numbers 43 and 68 in shown fig.2.1

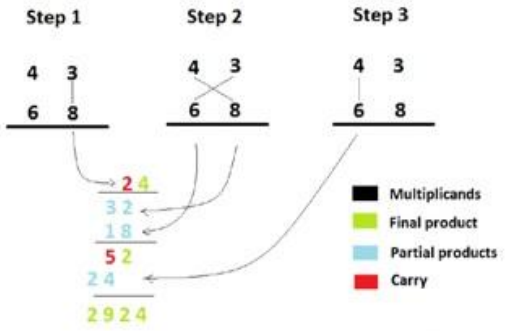


Fig.2.1 .bit multiplier design (43*68)

2.2 Vedic Multiplier for 4x4 bit Module

Let us consider the multiplication of two binary numbers a3a2a1a0 and b3b2b1b0. As the result of this multiplication would be more than 4 bits, we express it as ---r3r2r1r0.. Least significant bit r0 is obtained by multiplying the least significant bits of the multiplicand and the multiplier. The process is followed according to the steps shown in Fig.2.2 As in the last case; the digits on the both sides of the line are multiplied and added with the carry from the previous step. This generates one of the bits of the result (rn) and a carry (say cn). This carry is added in the next step and hence the process goes on. If more than one line are there in one step, all the results are added to the previous carry. In each step, least significant bit acts as the result bit and the other entire bits act as carry. For example, if in some intermediate step, we get 110, then 0 will act as result bit and 11 as the carry (referred to as cn in this text). It should be clearly noted that cn may be a multi-bit number. Thus we get the following expressions with c7r6r5r4r3r2r1r0 being the final product. Hence this is the general mathematical formula applicable to all cases of multiplication. The hardware realization of a 4-bit multiplier using this Sutra is shown in Fig.2.2

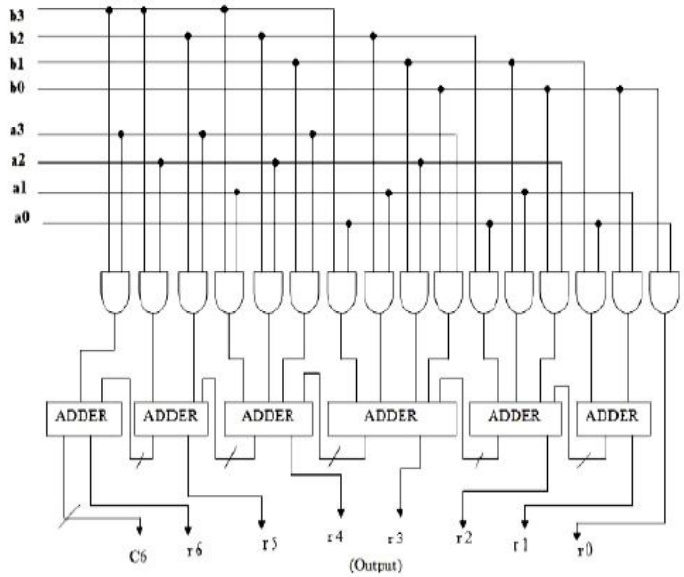


Fig 2.2.The hardware realization of a 4-bit multiplier using this Sutra

2.3 Vedic Multiplier for 8x8 bit Module

The 8x8 bit Vedic multiplier module as shown in the block diagram in Fig.2.3 can be easily implemented by using four 4x4 bit Vedic multiplier modules as discussed in the previous section. Let's analyze 8x8 multiplications, say A= A7 A6 A5 A4A3 A2 A1 A0 and B= B7 B6 B5B4 B3 B2 B1B0. The output line for the multiplication result will be of 16 bits as – S15 S14 S13S12 S11 S10 S9 S8 S7 S6S5S4 S3 S2 S1 S0. Let's divide A and B into two parts, say the 8 bit multiplicand A can be decomposed into pair of 4 bits AH-AL. Similarly multiplicand B can be decomposed into BH-BL. The 16 bit product can be written as: P = A x B = (AH-AL) x (BH-BL) = AH x BH + (AH x BL + AL x BH) + AL x BL Using the fundamental of Vedic multiplication, taking four

bits at a time and using 4 bit multiplier block as discussed we can perform the multiplication. The outputs of 4x4 bit multipliers are added accordingly to obtain the final product. Here total three 8 bit Ripple-Carry Adders are required as shown in Fig.2.3.

.

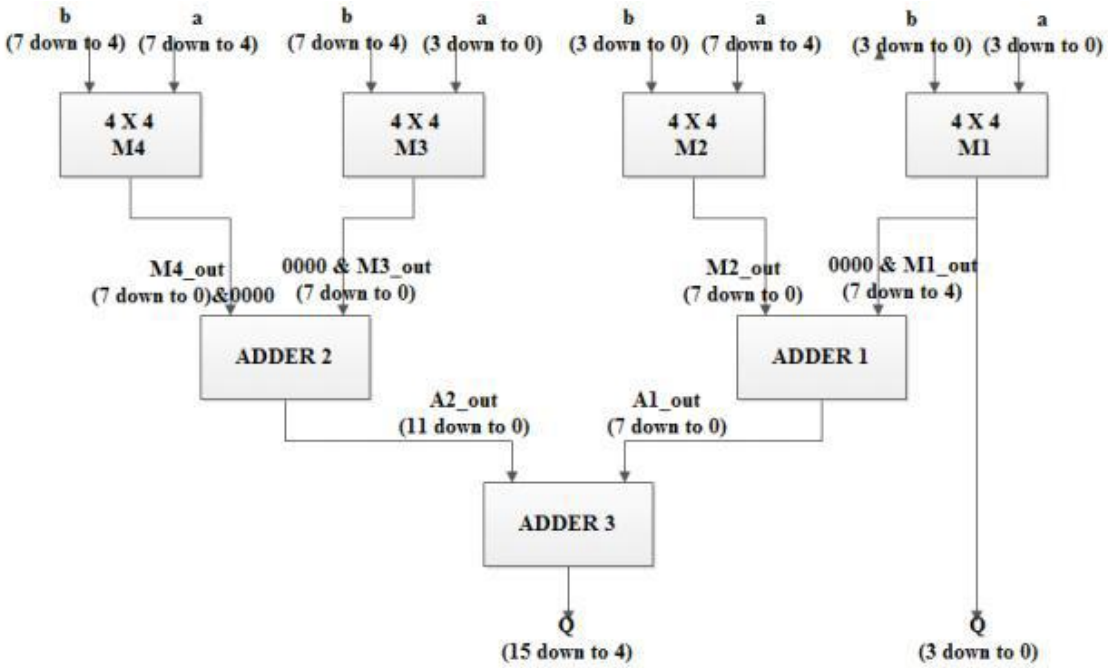


Fig2.3 The 8x8 bit Vedic multiplier module

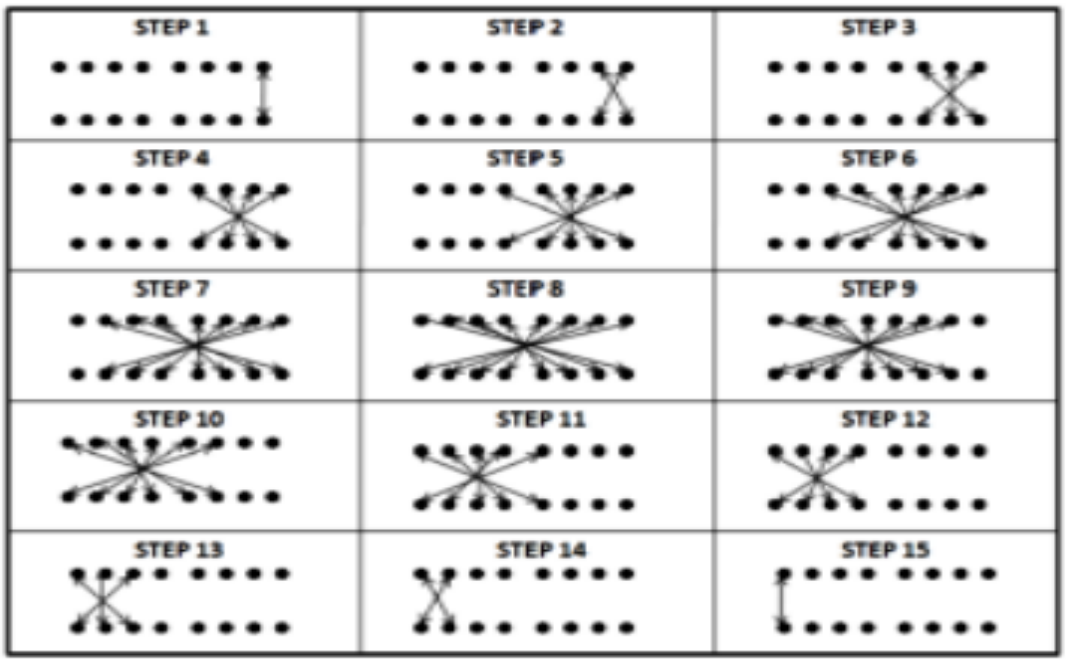
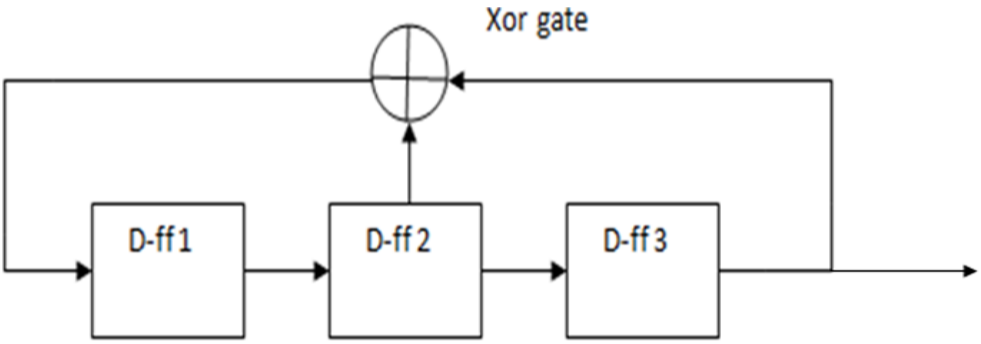


Fig 2.4. Bit binary multiplication using Urdhva Tiryakbhyam Sutra

3. The PN Sequence Generator

The PN Sequence Generator block generates a sequence of pseudorandom binary numbers using a linear-feedback shift register (LFSR).. A pseudo-noise sequence can be used in a pseudorandom scrambler and descrambler. It can also be used in a direct-sequence spread-spectrum system. This block can output sequences that vary in length during simulation. A PN-sequence

generator consists of D-Flip Flop and a XOR gate. At every rising edge of the clock pulse the contents of the registers are moved one bit position towards right. There is a feedback from the registers or the taps to the left most register of LFSR via XOR gate or we say In this each bit is right shifted and the output of second and third flip-flop are XORed For together and their output is fed back to the first flip-flop the sake of convenience three flip-flops are considered the block diagram is as follows:

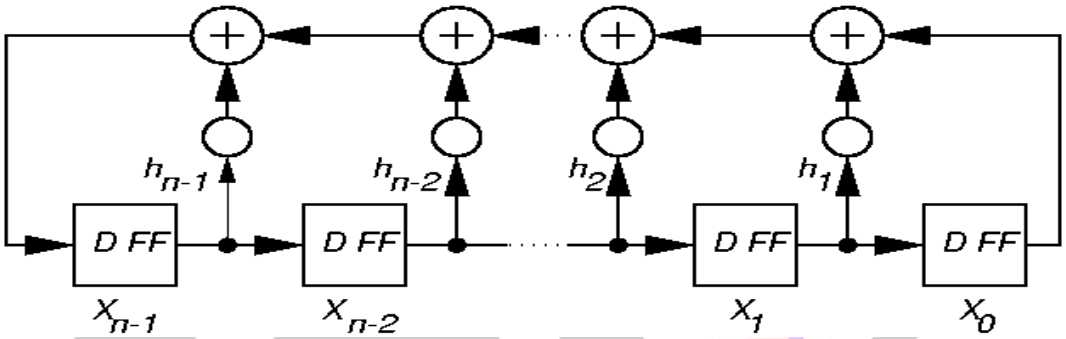


3.1 Basic PN sequence generator

Types of LFSR

There are two conventional forms of LFSR designs:

1) Standard LFSR: Figure 3.2 shows an n-stage standard LFSR. It consists of n flip-flops and a number of XOR gates. Since XOR gates are placed on the external feedback path, the standard LFSR is also referred to as an external-XOR LFSR.



3.2 An n-stage (external-XOR) standard LFSR

2) Modular LFSR: Similarly, an n-stage modular LFSR with each XOR gate placed between two adjacent flip-flops, as shown in Figure 3.3, is referred to as an internal-XOR LFSR. This circuit runs faster than its corresponding standard LFSR, because each stage introduces at most one XOR-gate delay.

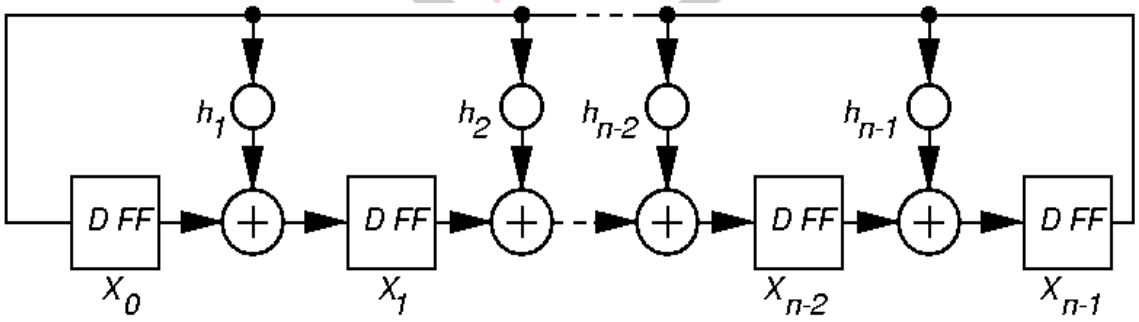


Fig 3.3 An n-stage (internal-XOR) Modular LFSR

Despite of different state trajectories, both structures are capable of generating an m-sequence for each stage output.

Design of 8 Bit LFSR.

The 8 bit LFSR whose maximum feedback polynomial is represented as X8+ X6 + X5 + X4 + 1 will produce 28-1 = 255 PN sequence. The block diagram of 8 bit LFSR is shown in Figure 3.4.

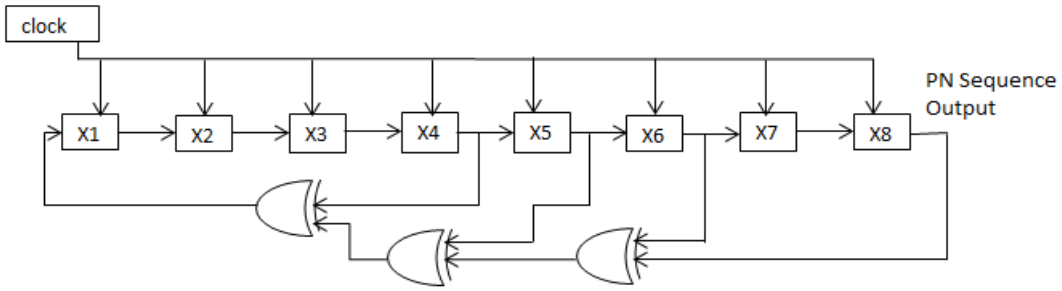


Fig 3.4 The block diagram of 8 bit LFSR

Design of 16 Bit LFSR.

The 16 bit LFSR whose maximum length feedback polynomial is represented as X16 + X14 + X13 + X11 + 1 will produces 216-1 = 65535 PN sequence. The block diagram of 16 bit LFSR is shown in Figure 3.5.

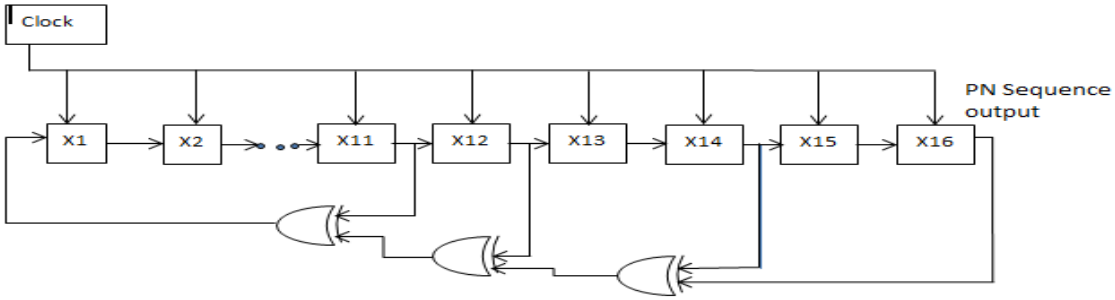


Fig3.5 The block diagram of 16 bit LFSR

4. Implementation Of PN Sequence Using Vedic Mathematics

The two message bits are X and Y are multiplied using vedic and then using the string of multiplied result are masked to generate PN sequence Using modulo -2 operator. The Galiose multiplier is used to generate sequence. In this paper the Galiose multiplier are replaced by Vedic multiplier .

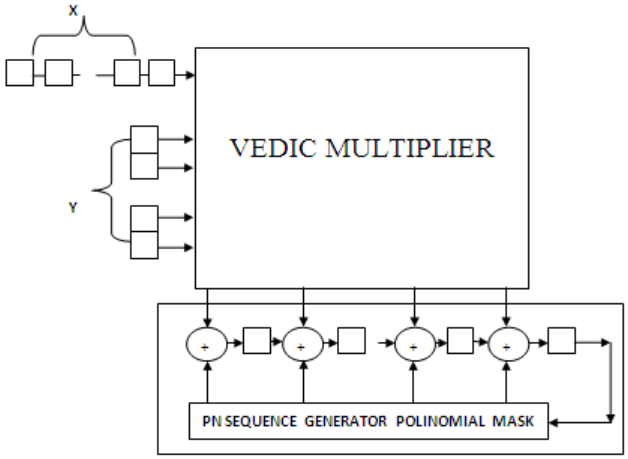


Fig 4. . Implementation Of PN Sequence Using Vedic Mathematics

5. Number of Additions and Multiplications in Different Multipliers

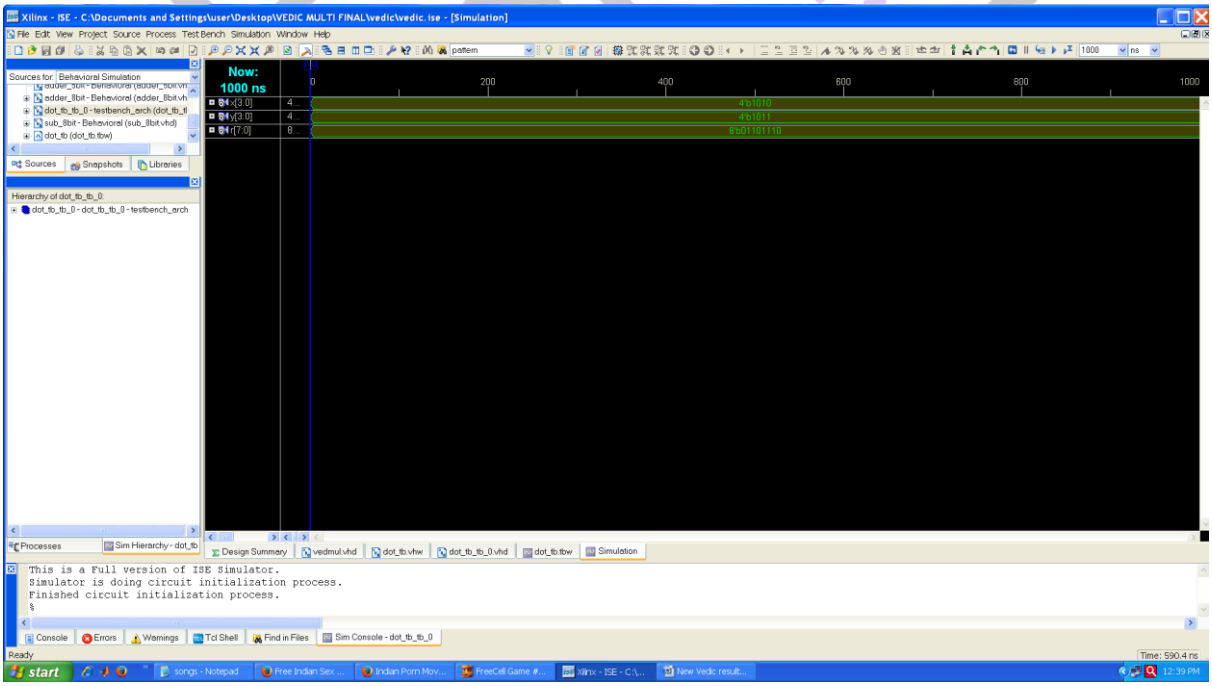
Types of multiplier	8x8-bit	16x16-bit	32x32-bit
Conventional	64-M 56-A	256-M 240-A	1024-M 1022-A
Booth	40-M 26-A	96-M 72-A	288-M 243-A
Vedic	8-M 4-A	16-M 8-A	32-M 16-A

'M' stands for multiplications used in the respective width of multipliers and 'A' stands for additions used in the respective width of multipliers. From this Table 1 it is clear that how the Vedic mathematics going to reduce the number of adder and multiplier as compare to the conventional and Booth multiplier. As we are reducing the number of adder in each bit the delay provided by the particular circuit is also going to reduced hence, which causes to increases the speed of MAC unit.

6. Result

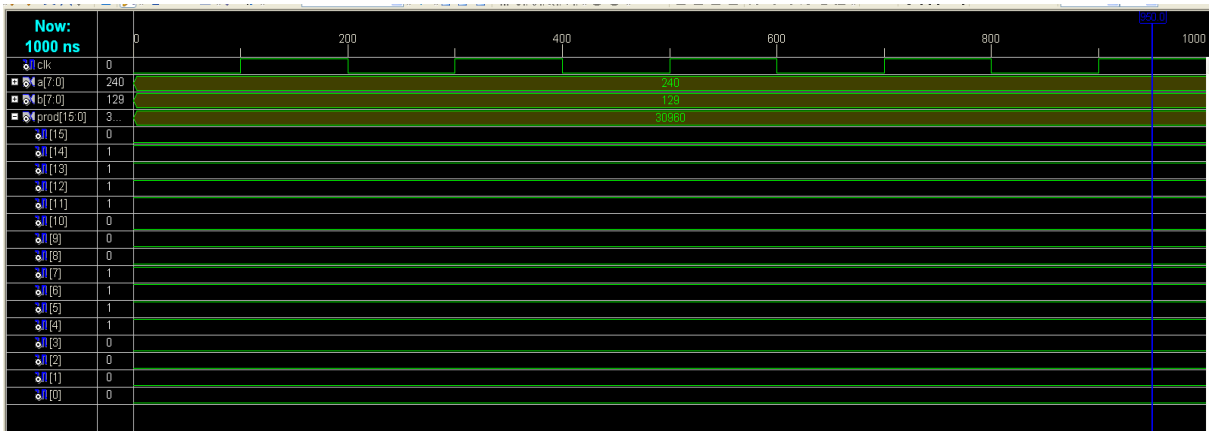
6.1 Vedic output of 4 bit multiplier
In our case-Input A-"1010"

Input B-"1011"
Output P-"01101110"
The simulation of above example is shown below



6.2 Vedic output of 8 bit multiplier

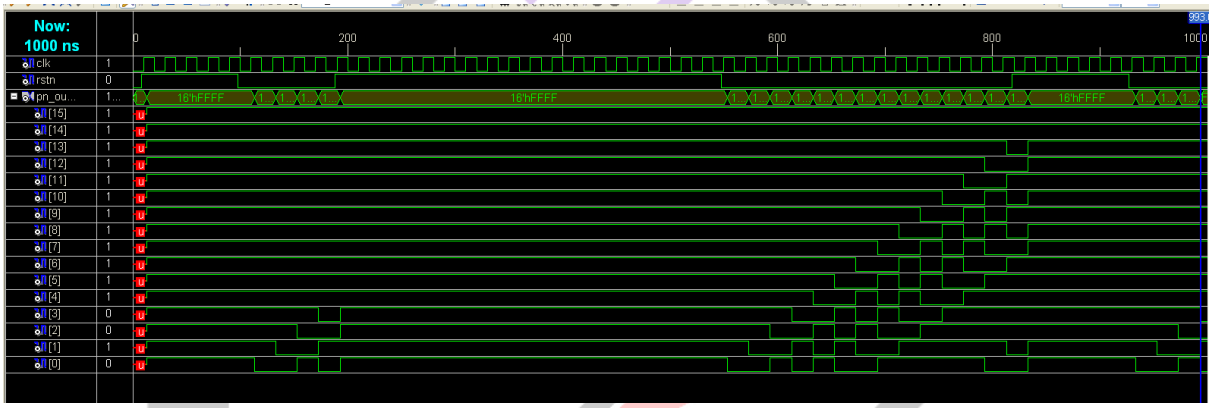
In our case-
Input A-"240"
Input B-"129"
Output P-"30960"
The simulation of above example is shown below



6.3 PN output

The output of the pn sequence can be generated throught modular LFSR with the help of 16 bit output of 8bit vedic multiplier.and other input clock signal is generated by crystal oscillaotor which have to be synchronous with the receiver.

The simulation of above example is shown below



7. Conclusion

The design complexity gets reduced for inputs of large number of bits and modularity gets increased .The advantages of the proposed architecture is efficient in speed and area and is Flexible in design Now this project will present a new method of multiplication. The design will be based on Vedic method of multiplication. This work will achieve the improved delay of convolutional encoder using vedic multiplier than booth multiplier. This gives us method for hierarchical multiplier design. So the design complexity will get reduced for inputs of large number of bits and will a provide faster speed. It is used to design a PN sequence generator and spread spectrum modulation to improve the utilization of bandwidth.the same vedic multiplier can also be used in other application to reduce propogation delay.

Reference

[1] John G. Prokis : Digital communications. *Mc. Graw Hill.*

[2] **[**Haykin-2001] Haykin, S**.** *Communication Systems.* 4th Edition. New York: John Wiley and Sons. .

[3] 1Roshni Jamgade,2 Shrikant Ambatkar,3Sandeep Kakde ,"HDL Implementation of PN Sequence Generatorusing Vedic Multiplication and Add & Shift Multiplication" 2015 Fifth International Conference on Communication Systems and Network Technologies

[4] Siddesh Gaonkar "Design Of 8 Bit, 16 Bit And 32 Bit Lfsr For Pn Sequence Generation Using Vhdl" International Journal of Technical Research and Applications e-ISSN: 2320-8163, www.ijtra.com Special Issue 31(September, 2015), PP. 305-308

[5] Jyoti Kumawat, Sunil Sharma "A 8x8 bit multiplier using Vedic Mathematics "International Journal of Engineering and Technical Research (IJETR) ISSN: 2321-0869, Volume-2, Issue-3, March 2014.

[6] Sweta Malviya1 and Poonam Kumari2, "Implementation of Pseudo-Noise Sequence Generator on FPGA Using VHDL", International Journal of Electronic and Electrical Engineering. ISSN 0974-2174 Volume 7, Number 8, 2014,pp.887-892.
[7] Prof J M Rudagil, Vishwanath Amble, Vishwanath Munavalli , Ravindra Patil , Vinaykumar Sajjan, "Design and implimentation of efficient multiplier using vedic mathamatics" , Proc. of1nt. Con/, on Advances in Recent Technologies in Communication and Computing 2011.
[8] Yogita Bansal , Charu Madhu , Pardeep Kaur, "HIGH SPEED VEDIC MULTIPLIER DESIGNS", Proceedings of 2014 RAECS UIET Panjab University Chandigarh.