A Novel Architecture for High Quality Random Number Generation based on Enhanced WELL Method

Harigovind[1], Vinoj P.G.[2]

[1]M.Tech Student, Department of Electronics and Communication Engineering, SCMS School of Engineering and Technology, Karukuuty, Cochin, Kerala, India

[2]Assistant Professor, Department of Electronics and Communication Engineering, SCMS School of Engineering and Technology, Karukuuty, Cochin, Kerala, India

Abstract: ***This paper presents an efficient architecture for high quality random number generation based on WELL method. This design is capable of achieving a throughput of one sample per cycle. The basic architecture for WELL random generator was based on Block RAM and register hybrid architecture which uses large number of resources on FPGA. The proposed work uses architecture based on Dual port RAM instead of Block RAM and it results in considerable reduction in power dissipation compared to the BRAM architecture. This proposed work also produces random numbers of high quality and non-repeating in nature. The comparison between both the architectures are made on the basis of the power rating estimated using Xilinx ISE tool.***

Keywords: Pseudo random number generation (PRNG), Mersenne Twister (MT), Block RAM(BRAM), Dual port RAM(DPRAM)

1. Introduction

Random numbers of high quality are commonly used in most of the scientific applications. Pseudorandom number generators (PRNGs) are widely adopted in such applications. PRNG is an algorithm for generating a sequence of numbers whose properties approximate the properties of sequences of random numbers. The PRNG-generated sequence is not truly random, because it is completely determined by the PRNG's seed, which is a relatively small set of initial values. Sequences that are closer to truly random can also be generated using hardware random number generators. But, pseudorandom number generators are used because of their speed in number generation and their reproducibility. Some of the most commonly used PRNGs are as, Linear Feedback Shift Register, Reseeding Method, Mersenne Twister, Well Equidistributed Long-period Linear (WELL) method.

Linear Feedback Shift register (LFSR) is one of the easiest method for random number generation. In computing, LFSR is a shift register whose input bit depends on its previous state. Ecxlusive-or (XOR) is the most commonly used linear function to determine the input bit. Thus, an LFSR is most often a shift register whose input bit is driven by the XOR of some bits of the overall shift register value. The major drawback is that it eventually enters into a repeating cycle since the register has a finite number of possible states.

Another method used for random number generation is Reseeding method. Shift registers produces random numbers based on the seeds loaded into look up tables. So this architecture is enhanced using static random access memory instead of Look up tables. By using SRAM, it is possible to make random number generator to process user defined data. This method uses a technique suitable for hardware implementation. It can be designed in the domain of VLSI to generate non-repeating sequence of patterns. But the speed of random number generation is slow compared to other methods.

Mersenne Twister (MT) is the mostly widely used algorithm for pseudo random number generation. It has very long period and good equi-distribution. The most commonly used version of the Mersenne Twister algorithm is based on the Mersenne prime 219937−1.The standard implementation, MT19937, uses a 32-bit word length. Mersenne twister random numbers have the colossal period of 219937-1 iterations are proven to be equidistributed and can be generated faster than other statistically reasonable generators. The major drawback of Mersenne Twister is that it produces numbers that pass randomness test after a long time. Even though it has a very long period of 219937 – 1, the quality of random number generated is not guaranteed. Another problem is that it requires long time to recover from zero-excess initial state and is vulnerable to poor initialization.. The well equi-distributed long-period linear (WELL) algorithm is proposed to fix this problem.

2. Literature Survey

Compared with MT, WELL has better equidistribution while retaining an equal period length. As application sizes scale, one emerging trend is to develop parallelized version of the applications to exploit the available parallel hardware resources such as FPGAs, to achieve high speed in performance. Being the key component of various scientific applications, designing PRNGs that can rapidly provide independent parallel streams of high quality random numbers is also becoming increasingly important in modern systems. The fast jump ahead technique provides an efficient method to determine the starting point of a new sub stream from an existing sub stream. Thus allowing multiple PRNGs to generate independent sub streams in parallel [1]. Most PRNGs focus on algorithms and software implementations. Only a few hardware implementations can be found. In previous works, a BRAM-and-register-hybrid architecture for WELL19937 with a throughput of one sample per cycle was used. Later a method with a more resource-efficient structure that reduces the usage of BRAMs from four to two was proposed and it retained the same throughput. The total resource used is also reduced as much as 50% compared with the original structure. This design also provides a software/hardware framework to parallelize its output stream based on the new structure and a resource-efficient hardware architecture for WELL with a throughput of one sample per cycle [1]. It has a dedicated 6R/2W RAM structure for WELL, which is capable of providing six Reads and two Writes concurrent in a single cycle, with little resource overhead. But one of the major drawback of this method is the power dissipation. Therefore, this paper proposes an architecture in which the BRAMs are replaced by a dual port RAM (DPRAM) so that the power and area are minimized. The simulations result of both architectures is same and the power is considerably low in the case of dual port ram based architecture.

3. Architecture for Basic WELL random number generator using BRAM

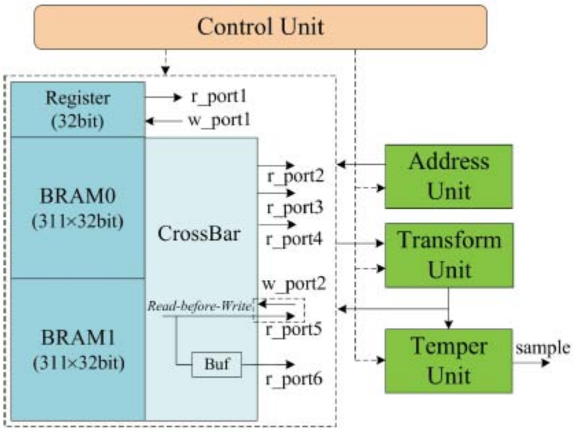


Figure 1: Architecture of WELL based on BRAM

It consists of five blocks:

- Control Unit
- Address Unit
- Transform Unit
- Temper Unit
- 6R/2W RAM.

The core component is the BRAM, which stores the 624 32 bit state vectors and is capable of concurrently supporting six Reads and two Writes. The Address Unit generates appropriate R/W addresses for the RAM. The Transform Unit and the Temper Unit perform the transform and temper operations of the WELL algorithm, and can be fully pipelined. The Control Unit produces the control signals to coordinate the system.

FPGA basically consists of elements like Block RAMs present inside in it. Usually they run parallel inside FPGA. its possible to have both single port as well as dual port block RAMs. Based on the transformation process of WELL algorithm, in each generation process, six blocks from the state vector are fetched while two blocks are updated. Therefore, to achieve the expected throughput, the RAM should read six operands and store two results concurrently in a single cycle. Such a RAM can be directly implemented using 624 32-bit registers, but this is not area-efficient and is impractical when building parallel PRNGs. It is also not straightforward to provide eight ports by simply assembling four BRAMs together, as we need to guarantee that the read and write operations are distributed across different BRAMs evenly. Instead, we propose a BRAM-and-register hybrid structure to build the required 6R/2W multiport RAM, which is the key component to achieve one sample per cycle throughput.

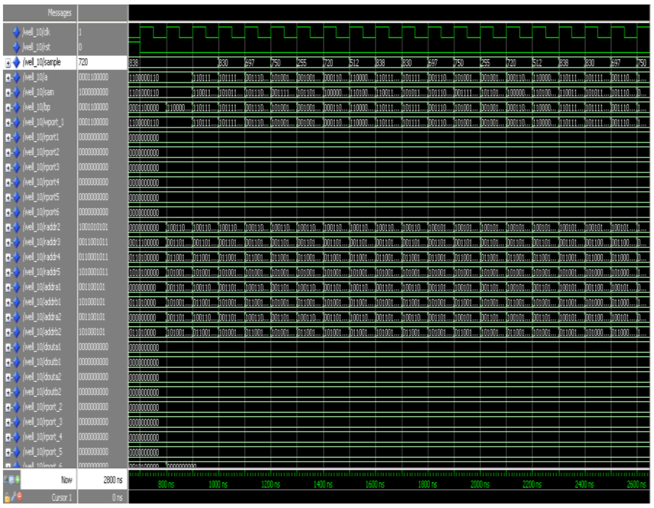
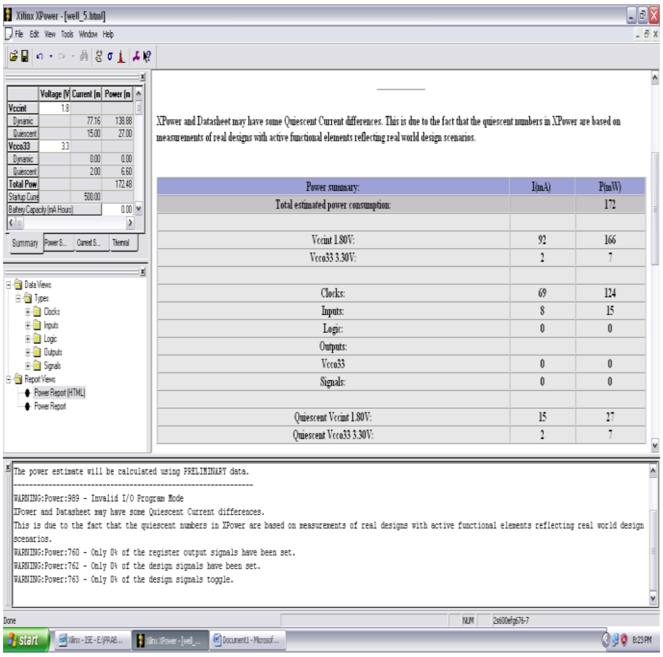


Figure 2: Simulation result of WELL random number generator using BRAM

Fig 2 shows the 5-bit simulation result of WELL random generator based on the Block RAM. The simulation was performed using ModelSim 6.3f. It shows that random numbers of very high quality can be generated by using the well equi-distributed long period linear algorithm. This method used the BRAM architecture and generated random numbers which are difficult to predict. Moreover, this method provides two write and six read operations in single clock cycle. The whole process starts at the rising edge of the clock and the unique random numbers are generated in each clock cycle.



	Voltage (V	Current (m	Power (m
Vccint	1.8		
Dynamic		77.16	138.88
Quiescent		15.00	27.00
Vcco33	3.3		
Dynamic		0.00	0.00
Quiescent		2.00	6.60
Total Pow			172.48
Startup Curre		500.00	
Battery Capacity (mA Hours)			0.00

XPower and Datasheet may have some Quiescent Current differences. This is due to the fact that the quiescent numbers in XPower are based on measurements of real designs with active functional elements reflecting real world design scenarios.

Power summary:	I(mA)	P(mW)
Total estimated power consumption:		172
Vccint 1.80V:	92	166
Vcco33 3.30V:	2	7
Clocks:	69	124
Inputs:	8	15
Logic:	0	0
Outputs:		
Vcco33	0	0
Signals:	0	0
Quiescent Vccint 1.80V:	15	27
Quiescent Vcco33 3.30V:	2	7

```
The power estimate will be calculated using PRELIMINARY data.
--------------------------------------------------------------
WARNING:Power:989 - Invalid I/O Program Mode
XPower and Datasheet may have some Quiescent Current differences.
This is due to the fact that the quiescent numbers in XPower are based on measurements of real designs with active functional elements reflecting real world design
scenarios.
WARNING:Power:760 - Only 0% of the register output signals have been set.
WARNING:Power:762 - Only 0% of the design signals have been set.
WARNING:Power:763 - Only 0% of the design signals toggle.
```

Figure 3: Power rating of WELL random number generator using BRAM

Fig 3 shows the power analysis in the case of WELL random generator based on BRAM. The power rating has been estimated using Xilinx ISE 8.1 and the total estimated power consumption was found to be 172mw.

The major drawback here is the usage of Block Ram (BRAM). The Block Ram (BRAM) is the embedded memory in FPGA which uses more resources in it. The power consumption of Block RAM is comparatively high since it uses many resources continuously. Even though, the random numbers generated by this method is of very high quality, this method is discarded because of the power consumption. Moreover, it uses more area and takes a fair amount of time while accessing the addresses. To overcome this we go for Enhanced WELL method where we use Dual Port Ram (DPRAM) instead of BRAM which reduce the resource usage.

4. Architecture for Enhanced WELL random number generator using DPRAM

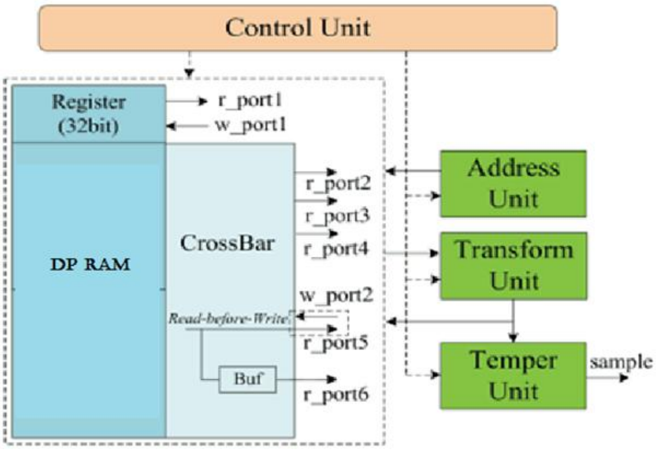
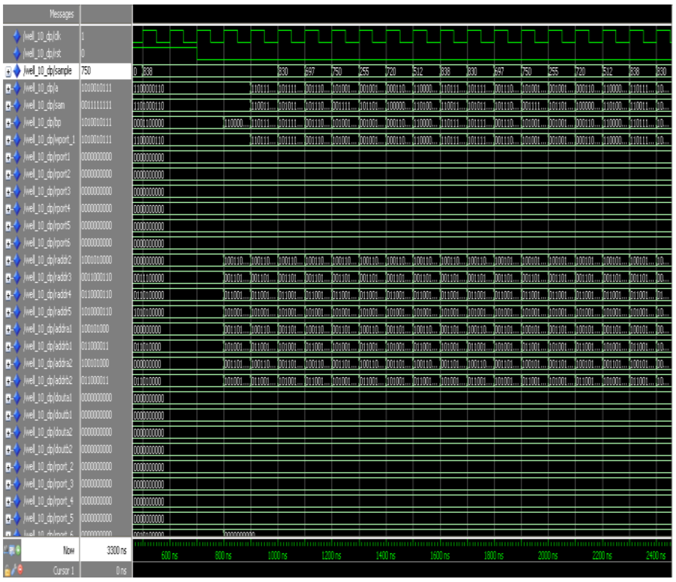


Fig.4. Architecture of WELL based on DPRAM

The architecture of WELL random number generator based on dual port RAM (DPRAM) is similar to that of the architecture based on Block RAM (BRAM). The dual port ram (DPRAM) is implemented within the look up table (LUT). In our proposed method the single dual port ram (DPRAM) which have both synchronous read and write operation simultaneously replaces two BRAM. The power is given only to the used resources (LUT) in it, which greatly consumes low power when compared to the BRAM.

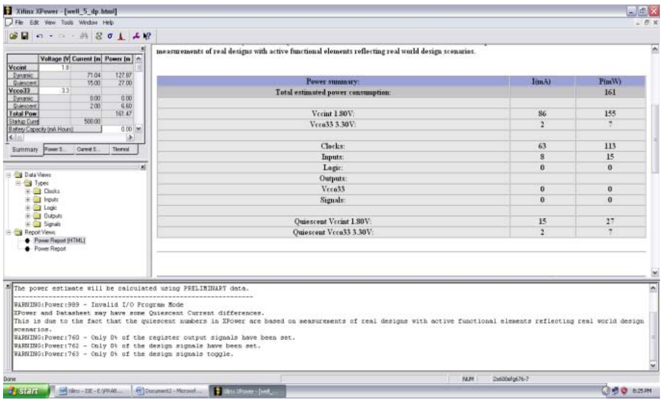
The address unit is used to generate the address for Dual Port Ram (DPRAM). The read address and write address is given to the Dual Port Ram (DPRAM) which reads the specific memory location of the RAM and will produce the data for their corresponding address. The transform unit will perform the transform operation. The structure of the transform unit consists of XOR gate and AND gate in it. The main operation of the transform unit is to perform row wise shifting in the input bit. The concat unit is used to concat the input bits. The output is given as input to the temper unit which will perform the temper operation. The structure consists of XOR gate and AND gate. The main operation of the temper unit is to perform column wise shifting in the input bit(z4) and produce the final sample output.



Messages	
/well_10_dp/clk	1
/well_10_dp/rst	0
/well_10_dp/sample	750
/well_10_dp/a	1010010111
/well_10_dp/sam	0011111111
/well_10_dp/bp	1010010111
/well_10_dp/wport_1	1010010111
/well_10_dp/rport1	0000000000
/well_10_dp/rport2	0000000000
/well_10_dp/rport3	0000000000
/well_10_dp/rport4	0000000000
/well_10_dp/rport5	0000000000
/well_10_dp/rport6	0000000000
/well_10_dp/raddr2	1001010000
/well_10_dp/raddr3	0011000110
/well_10_dp/raddr4	0110000110
/well_10_dp/raddr5	1010000110
/well_10_dp/addra1	100101000
/well_10_dp/addrb1	011000011
/well_10_dp/addra2	100101000
/well_10_dp/addrb2	011000011
/well_10_dp/douta1	0000000000
/well_10_dp/doutb1	0000000000
/well_10_dp/douta2	0000000000
/well_10_dp/doutb2	0000000000
/well_10_dp/rport_2	0000000000
/well_10_dp/rport_3	0000000000
/well_10_dp/rport_4	0000000000
/well_10_dp/rport_5	0000000000

Figure 5: Simulation result of WELL random number generator using DPRAM

Fig 5 shows the 5-bit simulation result of WELL random generator based on the Dual port RAM. The simulation was performed using ModelSim 6.3f. It shows that random numbers of very high quality similar to the one generated using BRAM architecture can be generated by using DPRAM.



	Voltage (V	Current (m	Power (m
Vccint	1.8		
Dynamic		71.04	127.87
Quiescent		15.00	27.00
Vcco33	3.3		
Dynamic		0.00	0.00
Quiescent		2.00	6.60
Total Pow			161.47
Startup Curr		500.00	
Battery Capacity (mA Hours)			0.00

measurements of real designs with active functional elements reflecting real world design scenarios.

Power summary:	I(mA)	P(mW)
Total estimated power consumption:		161
Vccint 1.80V:	86	155
Vcco33 3.30V:	2	7
Clocks:	63	113
Inputs:	8	15
Logic:	0	0
Outputs:		
Vcco33	0	0
Signals:	0	0
Quiescent Vccint 1.80V:	15	27
Quiescent Vcco33 3.30V:	2	7

```
The power estimate will be calculated using PRELIMINARY data.
--------------------------------------------------------------
WARNING:Power:989 - Invalid I/O Program Mode
XPower and Datasheet may have some Quiescent Current differences.
This is due to the fact that the quiescent numbers in XPower are based on measurements of real designs with active functional elements reflecting real world design
scenarios.
WARNING:Power:760 - Only 0% of the register output signals have been set.
WARNING:Power:762 - Only 0% of the design signals have been set.
WARNING:Power:763 - Only 0% of the design signals toggle.
```

Figure 6: Power rating of WELL random number generator using DPRAM

Fig 6 shows the power analysis in the case of WELL random generator based on DPRAM. The power rating has been estimated using Xilinx ISE 8.1 and the total estimated power consumption was found to be 161mw.

5. Inferences

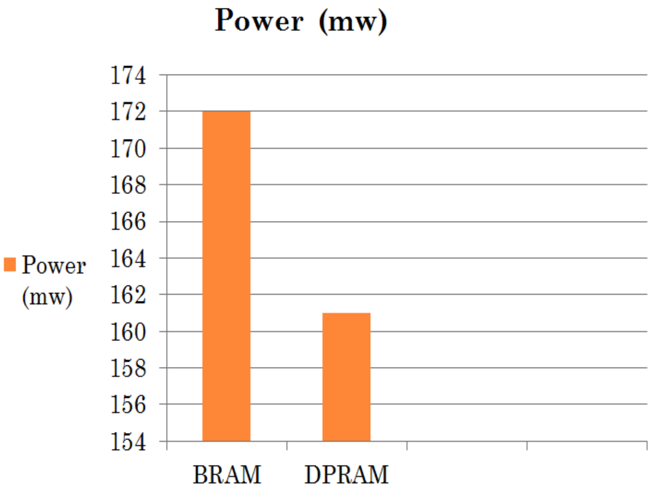


Figure 7: Power consumption comparison

Power is calculated in milli Watt (mW) and the comparison result in shown in fig.7. The table 1 gives the power of both architectures. It shows that WELL random generator based on Dual port RAM requires minimal power. Here power is reduced by several mW using proposed architecture when compared to existing architecture which is based on the Block RAM.

Table 1: Power consumption

	BRAM Architecture	DPRAM Architecture
Power (mW)	172mW	161mW

This indicates that the dual port memory in the proposed scheme performs faster than the block RAM based architecture. Even though, there is only slight reduction in area and delay for Dual port RAM, this architecture provides a good amount of reduction in power consumption. This method is capable to bring down the power from 172mw to 160mW and it limits the use of the resources.

6. Future Scope

The random numbers generated are highly unpredictable which enables this method to be used in cryptography and many scientific calculations like Monto Carlo simulations. The Enhanced WELL random generator can be used for many circuit testing purposes. One of the development that can be done is that, this concept can be used to replace the LFSR used in Built In Self Test (BIST). The LFSR based BIST mostly detects only easy to detect faults. So the enhanced WELL method can be used to replace the LFSR and thereby, the accuracy of BIST can be increased.

As further development, a hardware architecture for enhanced WELL method that includes interleaver address generator which uses bulk of circuitry to generate the address can be done. This greatly reduces the complexity of the circuit and will achieve high performance output. The generated random sequences are then applied to the benchmark circuits for circuit testing.

7. Conclusion

Through our study we demonstrated that our proposed advanced WELL method generates random numbers that are non-repeating in sequence which achieves low power consumption and high quality output. This design using dual port RAM achieves a throughput of one sample per cycle and produces a high quality random numbers which can be used in testing applications. The power consumption is reduced by a considerable amount, thereby increases the overall performance in the testing process.

References

[1] Yuan Li, Paul Chow, Senior Member, IEEE, Jiang Jiang, Minxuan Zhang, and Shaojun Wei, “Software/Hardware Parallel Long-Period Random Number Generation Framework Based on the WELL Method”, IEEE transactions on very large scale integration (VLSI) systems, vol. 22, no. 5, May 2014

[2] D. B. Thomas and W. Luk, “High quality uniform random number generation using LUT optimised state-transition matrices,” J. VLSI Signal Process, vol. 47, no. 1, pp. 77–92, 2012.

[3] D. B. Thomas and W. Luk, “The LUT-SR family of uniform random number generators for FPGA architectures,” IEEE Trans. Very Large Scale Integrat. (VLSI) Syst., vol. 21, no. 4, pp. 761–770, Apr. 2013.

[4] F. Panneton, P. L’Ecuyer, and M. Matsumoto, “Improved long-period generators based on linear recurrences modulo 2,” ACM Trans. Math.Softw., vol. 32, no. 1, pp. 1–16, Mar. 2006.

[5] H. Haramoto, M. Matsumoto, T. Nishimura, and P. L’Ecuyer, “Efficient jump ahead for F2-linear random number generators,” Inf. J. Comput., vol. 20, no. 3, pp. 385–390, 2008.

[6] I. L. Dalal and D. Stefan, “A hardware framework for the fast generation of multiple long-period random number

streams," in Proc. 16th ACM Int. Symp. FPGAs, Feb. 2008, pp. 245–254.

Author Profile

Harigovind received the B.Tech degree in Electronics And Communication Engineering from Mahatma Gandhi University, Kerala at Mar Baselios Christian College Of Engineering And Technology 2013 and now he is pursuing his M.Tech degree in VLSI and Embedded systems under the same university in SCMS School of Engineering and Technology, Cochin.