Sampling Based Random Number Generator for Stochastic Computing

M.Burak Karadeniz and Mustafa Altun
ECE Department, Istanbul Technical University, Istanbul, TURKEY
karadeniz17@itu.edu.tr, altunmus@itu.edu.tr

***Abstract*—Linear feedback shift register (LFSR) has been widely used to generate stochastic bit streams. Although using LFSR's offers feasibility because of their compatibility with CMOS technology, lack of randomness and related area consumption which is linearly proportional to the number of bits in a stream satisfying a certain probability value, can easily go beyond practical limits. Until now, no distinguished and practical way has been found to compete with LFSR to generate stochastic bit streams. True random number generators (TRNG) are widely used to compensate the poor randomness of LFSR but their complex design which is increased by the sake of acquiring random source, and their uncontrollability to generate random bit stream with desired probability, which is necessary for stochastic applications, make them out of action. Here we propose a novel programmable sampling based stochastic number generator (SBRNG) using CMOS technology. We achieve 100x higher speed, and 640x effective length of stochastic bit streams compared to LFSR based generators. We also claim that the circuit area complexity in terms of the number of effective bits is much better for SBRNG compared to LFSR based generators.**

Keywords—LFSR; TRNG; stochastic number generator; quantization; analog to digital converter (ADC), CMOS

I. INTRODUCTION

In stochastic applications, one of the major problems is a need to generate extensively large stochastic bit streams in a limited area. To generate bit streams, linear feedback shift registers (LFSR's) are conventionally used and their area performance and randomness quality are not satisfactory. The consumed area is linearly increasing with the number of effective (randomly distributed) bits in a stream satisfying a certain probability value. This probability can be calculated as the number of 1 valued bits divided by the total number of bits in a stream. Also it cannot be programmable, once the tap bits (D-type F/F's to be XOR'ed) are determined. The third major drawback of the LFSR is the power delay product which is limited by the speed of clock and the number of bits in a stream. To improve area efficiency, LFSR's are used in special networks to generate stream bits in parallel, not in series [1]. To improve randomness, other kinds of designs are used tinkering with LFSR's [2],[3]. Additionally, true random generators are used [4],[5],[6]. However, their feasibility is questionable because of the used emerging technologies with weak assumptions for the source of randomness. Also, since true random generators cover all frequency domain, it is almost impossible to achieve a stream satisfying desired probability and speed (min. duration of a bit).

Considering these shortcoming, we propose a novel programmable sampling based stochastic number generator (SBRNG) using. We are motivated by the fact that the sampling and quantization processes result in random quantization noise [7]. Although quantization error can be drawn out after ADC-DAC operation from the original signal, this would be burden and slow way to get streams; area increases terribly and speed goes down immediately. We show that same stochastic behavior right ahead of the sampling before quantization can be achieved.

In our approach, we first generate a periodic source signal that can be sinusoidal or triangle. For this purpose, we use an oscillator circuit. Then we sample the signal followed by subtraction from the original signal. Thus, we achieve a random noise source which is similar to quantization noise. Finally, depending on the desired probability value, we convert random signals to a stochastic bit stream by using a comparator. The general flow of our approach is shown in Fig. 1. The foremost advantage of our technique is to generate stochastic signal much faster than the sampling (clock) frequency. The other striking point is that the design is completely programmable. LFSR is producing the same number sequences in every overlapping period. In our design, adjusting the reference voltage ensures to give binomially distributed bit streams with different sequences in every cycle.

This paper is organized as follows. First, generating bit streams from sampling process and building blocks of the proposed SBRNG have been addressed. Second, compatibility of generated stochastic bit stream is analyzed by considering probability density functions (PDF's) and their histograms. Third, the proposed design and the conventional method (LFSR) outputs are simulated in HSPICE and analyzed in MATLAB in a way of randomness quality, effective number of bits, and speed. Finally, some recent random number generators (RNG's) existing in literature are compared with the proposed SBRNG.

II. BUILDING BLOCKS

A. Definitions

Stochastic Bit Stream: Binomially distributed 0's and 1's in a binary stream.
Desired Probability (P_X): Desired number of binary 1's divided by the length of a stochastic bit stream.
Stochastic Random Number Generator (SNG): Stochastic bit stream generator with desired probability.
Sampling Frequency (f_c): Frequency of sampling signal
Signal Frequency (f_S): Frequency of source signal
Resolution(Res): The minimum desired probability that can be generated by SNG
Non-overlapping cycle: The period of stochastic bit stream which has different sequence of binary numbers.

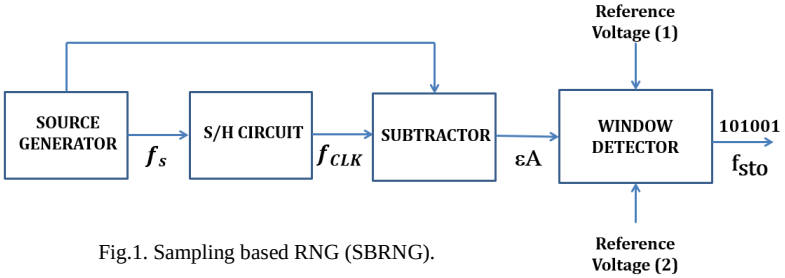

Fig.1. Sampling based RNG (SBRNG).

Stochastic Bit Stream Frequency (f_{sto})**:** 1 over minimum period of bits in a stream in non-overlapping cycle.
Effective Number of Bits: Number of bits in non-overlapping cycle of stochastic bit stream.
Quantization Error (eQ)**:** The leftover signal after the source signal is quantized and subtracted from the original source.
Sampling Error (eA)**:** The leftover signal after the source is sampled and subtracted from the original source.
Sample and Hold **(S/H):** Technique to sample signal with f_{clk}
Sampling Rate **(*SR*)*:*** Rate of signal frequency over sampling frequency.

B. Proposed Design Blocks

Assumption: If the sampling frequency and the source frequency are co-prime (no common divider), quantization error (eQ) and sampling error (eA) has maximum randomness or stochastic behavior.

Statement : PDF of eA has similar behavior as PDF of eQ because eA is a subset of eQ. Since eQ is uniformly distributed between half of quantization step (-/+) q/2, eA is uniformly distributed in its max and min sampling error values. The input voltage of source is expressed as;

$$V_{in} = A * Sin(2\pi f_S t) \quad (1)$$

The derivative of signal points to maximum points;

$$\frac{dV}{dt}(max) = A * 2\pi f_S * Cos(2\pi f_S t)(1 - Duty\ Cycle) \quad (2)$$

$$\text{eA} = \text{A}\pi f_S / f_c \ (DC = 0.5) \quad (3)$$

Case 1: A=2v, f_S=1730Hz, f_c=11717Hz, # of quantization bits (B)=3, (DC)=%50, Max Errors are calculated such that:

$$\text{eA} = \text{A} * \frac{2\pi fs}{fc} * (1 - DC) = 0.92$$

$$\text{eQ} = \Delta = \frac{A}{2^{B-1}} = 0.25\text{V} \left[-\frac{\Delta}{2}, \frac{\Delta}{2}\right] \text{(Fig.2)}$$

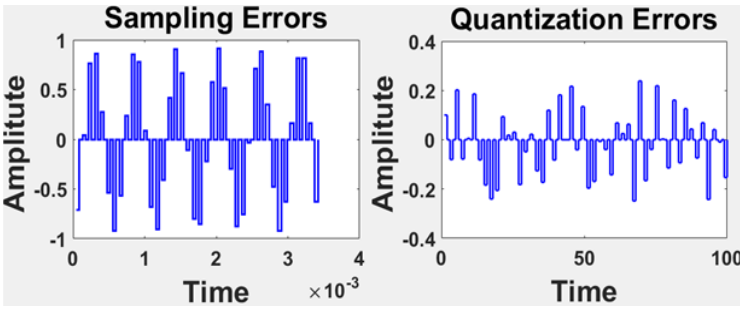

Fig. 2. Sampling error, eA (left); quantization error,eQ (right).

Wave Generator: The random source has been acquired from periodic signal which is built in SBRNG. There has been variety types of oscillators in the literature such as Hartley, Crystal, Armstrong, and Wien Bridge oscillators with unique futures. Wien bridge oscillator, which yields smooth sine wave with oscillating frequency that is easily controllable, is selected in our design (Fig.3) [8], and it works efficiently in the range of frequency up to 100 KHz which is quite proper to be used in SBRNG for which signal is sampled by much higher clock frequency.
For Wien bridge oscillator, if R1=R2=R, C1=C2=C then the oscillating frequency is determined such that:

$$f_S = \frac{1}{2\pi RC} \quad (4)$$

S/H Circuit & Subtractor: The generated wave is sampled at above or equal to Nyquist rate and then subtracted from the original signal to get sample and hold error which has similar behavior with quantization signal. Basic track and hold topology is selected for simplicity. In the track mode, the switch is ON and the voltage at the output followed; in the hold mode the switch is OFF and the capacitor cannot discharge because of the high impedance at the input of the op-amp. RC values are set by the charge-discharge (5τ) of the network according to the switch time;

$$RC = \frac{1}{5fc} \quad (5)$$

Window Detector: It determines the probability of the stochastic bit streams. If the input signal is between reference voltage levels, it gives analog high as expected.

Output probability is extracted from the characteristic of uniform distribution.

$$P_x = (b - a)/|2eA_{max}| \quad (6)$$

where a and b are the reference voltages that are to be fed through window detector (Fig. 4).

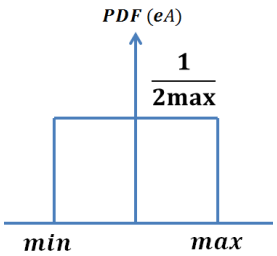

Fig.4. Probabilty density function of sampling error (eA).

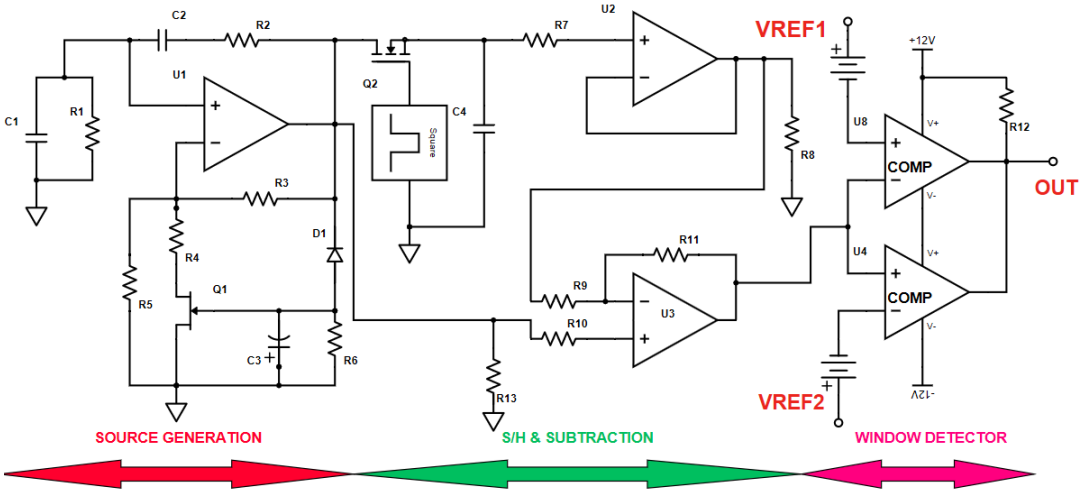

Fig.3. Proposed sampling based stochastic random number generator circuit.

Case 2: A=5v, f_S=127Hz, f_C=3127Hz, DC=%60, a=1mV, b=201mV.

$$eA = A * \frac{2\pi fs}{fc} * (1 - DC) \quad = (+/-)510\ mV(eAmax)$$

$$P_x = \frac{b-a}{|2eA_{max}|} \approx \%20$$

C. Design Flowchart

Proposed SBRNG design consists of three parts as discussed above in Fig. 3. All of the parts are built as scalable/adjustable. Design specs are set by the user according to the needs in stochastic design. The flowchart in Fig. 5 represents the steps needed to be followed for circuit design. Considering the fact that the output characteristics of the design are determined first, the reverse engineering is expressed by the following equations;

$$\frac{\mathrm{dV}}{\mathrm{dt}}(\max) = \mathrm{A} * 2\pi f_s * \mathrm{Cos}(2\pi f_s t)\ (CosTerm = 1) \quad (7)$$

$$f_s = \frac{f_{sto}\Delta V}{2\pi A} \quad (8)$$

From Eq. (6) and P_x = Res, $\Delta V = V_{ref1} - V_{ref2}$;

$$\Delta V = 2\pi A \frac{f_s}{f_c} Res \quad (9)$$

Converging Eq. (8) by (9);

$$f_s = f_{sto} xSRxRes \quad (10)$$

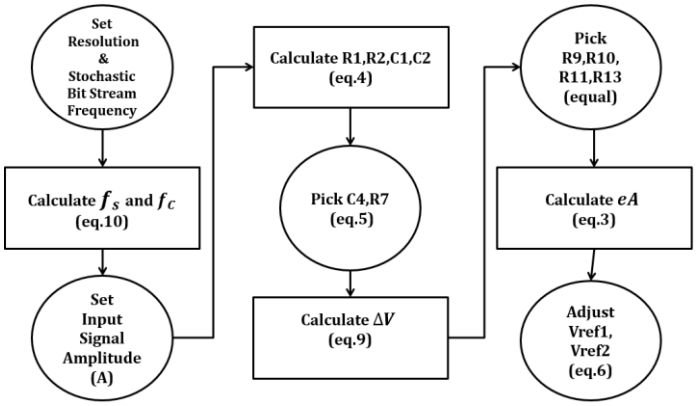

Fig.5. SBRNG design flowchart

III. Analysis of Bit Stream Compatibility

We initially assume that the distribution of sampling errors is uniform and certain parameters are given based on that.

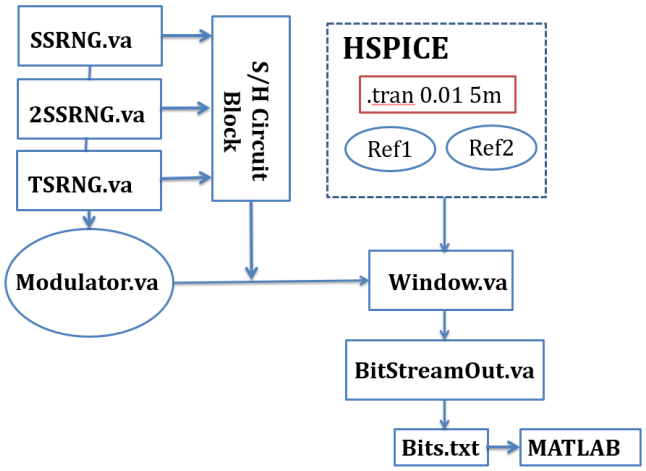

Fig.6.VerilogA/HSPICE setup for different sourced SBRNG

However, the fact is that the sampling errors (eA) from the one sourced sinusoidal wave is not distributed perfectly uniform. Non-ideality associated with the single source is addressed below. In order to compensate the non-ideal uniformity of eA coming from one source, multiple sources are used. Stochastic bit streams are examined in Verilog-A models and out in txt format by HSPICE (Fig. 6).

A. Single Sine-Wave Sampling Based RNG (SSRNG)

Sampling errors sorted from SSRNG are accumulated as sigmoidal as expected because the maximum sampling errors happen at the zero crossings of the source in shorter time; note that smaller errors happen at the peaks in longer time (Fig 7). The PDF of sampling errors is the derivative of sorting and shows that its distribution is not perfectly uniform (Fig 8).

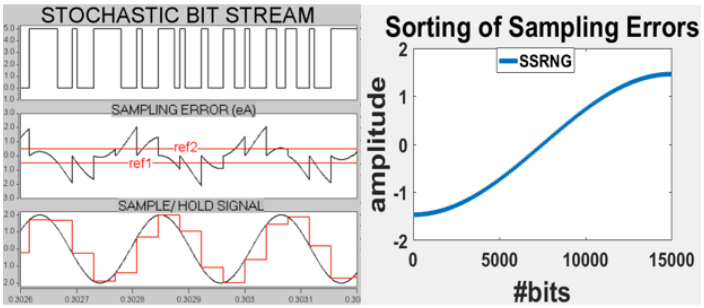

Fig.7. SSRNG HSPICE simulation (left), eA sorting (right)

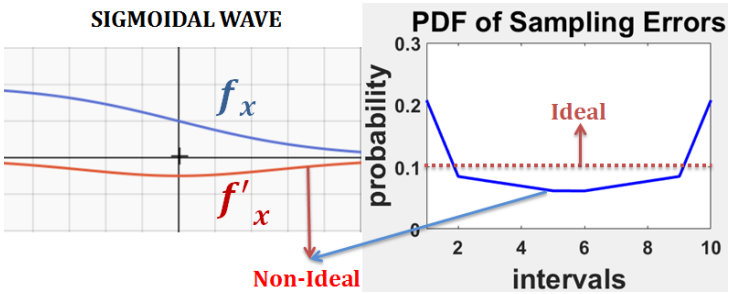

Fig.8. Non-uniformity of sigmoidal shape (left), eA distribution from unique sine-wave (right)

B. Multiple Sine-Wave Sampling Based RNG(2SSRNG)

Quasi-uniform behavior of sampling errors from one source is compensated by using multiple sources with applying Gibbs phenomenon [9]. If the source has uniform shape as square wave, the distribution of errors will be uniform since the sampling wave is uniform so is the sampling error. The square wave (x_t) is expressed as the harmonics of sinusoidal signals;

$$x_t = \frac{4}{\pi}(\sin(x) + \frac{1}{3}\sin(3x) + \frac{1}{5}\sin(5x) + \cdots) \quad (11)$$

Error cocktail from two sources has two poles that compensate the non-ideality of the sorting of the sampling errors to the ideal ramp shape. Therefore, the PDF closes the uniform form (Fig 9).

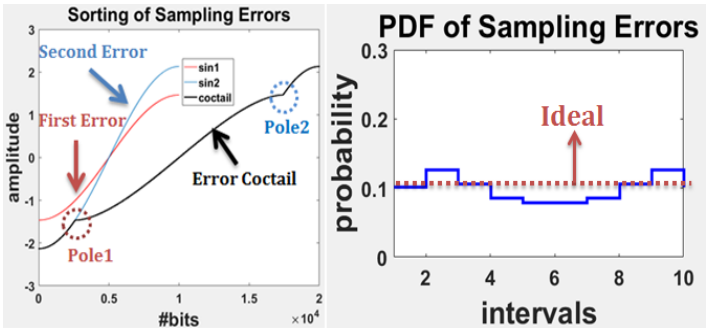

Fig.9 Double source SBRNG (2SSRNG) eA sorting (left), distribution (right)

C. Triangular-Wave Quantization Based RNG

Sampling triangular wave is another application that is done to smooth the distribution of sampling errors even further. Here, sampling frequency cannot be much higher than the source signal frequency, so it cannot be the oversampling rate otherwise the same errors that come from the every hold time spoil the richness of randomness of stochastic bit stream.

D. Binomial Distribution Compatibility

Stochastic bit streams with desired probability generated by the proposed design, have been taken in txt format from HSPICE and then have been loaded into MATLAB simulation. By sampling different bit sizes, comparison is achieved with binomial distribution (generated in MATLAB). Results are shown in Fig. 10.

IV. Experimental Results and Conclusion

As evaluation criteria, we consider speed and number of effective bits for randomness as well as area. HSPICE transient analysis is done both for the proposed design with 2,3,4 sine sources (SBRNG 2,3,4) and for the conventional 8-bit LFSR and 13-bit LFSR. We also consider RNG used in MATLAB as well as some recent studies offering new RNG's or TRNG's. Note that most of the RNG's in the literature are not suitable to produce desired probability values in desired speed since they cover all range of different frequencies. A filter can be used, but this worsens the randomness.

Experimental results in Table I show that the proposed SBRNG's outperforms other methods. Proposed designs give 640x higher throughput over conventional methods. Also its speed is 100 times better than that of LFSR based techniques (if the clock frequency for LFSR and sampling frequency for SBRNG are same). For area calculations we use transistor counts of circuit elements, given in parenthesis: AND (6), NOT (2), NAND (4), D-F/F (12), OPAMP (20), analog window comparator (40), digital comparator (22). Area performance of the proposed technique is also satisfactory; by only adding one oscillator, we can significantly increase effective bit numbers. Of course, to give a concrete conclusion, more detailed simulations by considering variations and layout areas are needed. This is considered as a future work.

Another important advantage of our design is its feasibility and controllability. There are tremendous amount of studies on RNG's and TRNG's in the literature, they cannot be controlled to produce a stochastic stream having desired probability and speed (minimum time duration of a bit). We believe that the proposed SBSNG's have a real potential to replace LFSR based techniques.

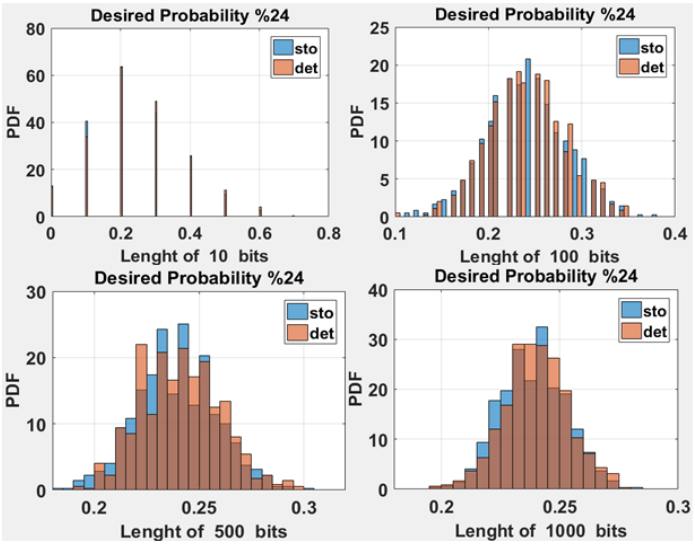

Fig.10. Compatibility test of single source SBRNG, 10 bits (up-left),100 bits (up-right), 500 bits (down-left), 1000 bits (down-right); blue (SBRNG), red (MATLAB generated binomially distributed bit stream), brown if converged (conventional and proposed bit streams)

Acknowledgment: This work is supported by the TUBITAK 1001 project # 116E250.

References

[1] Vikash Sehwag," A Parallel Stochastic Number Generator with Bit Permutation Networks", IEEE Transactions on Circuits and Systems II: Express Briefs, 2017

[2] Hideyuki Ichihara," Compact and Accurate Stochastic Circuits with Shared Random Number Sources", IEEE 32nd International Conference on Computer Design (ICCD), 2014

[3] C-K. Pham, M. Fukuda, "A Stochastic Pulse Bit-Stream with High Accurate Multiplication", The 47th IEEE International Midwest Symposium on Circuits and Systems, 2004

[4] Yuanzhuo Qu, Jie Han, Bruce F. Cockburn and Witold Pedrycz, "A True Random Number Generator based on Parallel STT-MTJs", Design, Automation and Test in Europe (DATE), 2017

[5] Sanu K. Mathew, "2.4 Gbps, 7 mW All-Digital PVT-Variation Tolerant True Random Number Generator for 45 nm CMOS High-Performance Microprocessors", IEEE Journal of Solid-State Circuits, VOL. 47, NO. 11, November 2012

[6] Saman Kiamehr, "Leveraging Aging Effect to Improve SRAM-based True Random Number Generators", Design, Automation and Test in Europe (DATE), 2017

[7] Robert M. Gray and David L. Neuhoff, "Quantization", IEEE Transactions on Information Theory, Vol. IT-44, No. 6, pp. 2325–2383, Oct. 1998

[8] Sergio Franco, "Design with Operational Amplifiers." McGraw Hill Edition, pp. 451-457, 2002.

[9] Hewitt, Edwin; Hewitt, Robert E. (1979). "The Gibbs-Wilbraham phenomenon: An episode in Fourier analysis". Archive for History of Exact Sciences. 21 (2): 129–160

TABLE I. Comparison of the prposed SBRNG (2,3,4 Signal Sourced) with 8-bit LFSR, 13-bit LFSR, and other tehcniques in the literature.

	LFSR 8	**LFSR 13**	**MATLAB Randn (1,160000)**	**GLFSR [3]**	**STT-MTJ [4]**	**SRAM Based [6]**	**All- Digital PVT Variation Tolerant [5]**	**SBRNG2 (this work)**	**SBRNG3 (this work)**	**SBRNG4 (this work)**
Effective bit numbers	256	8192	160000	256	256	256	1048576	8100	60000	160000
Transistor count with normalized%50 probability output	114	174	NA	108	9719	1536	734693	61	81	101
Transistor count with desired probability output	290	460	NA	Not reported	Not reported	Not reported	Not reported	101	121	141
Speed f_{sto} (GHz)	Clock Frequency (CLK)		Non-Available (NA)	CLK	0.177-0.2	NA	0.1-2.9	CLKx100		