Efficient Hardware Implementations of the Warbler Pseudorandom Number Generator

Gangqiang Yang, Mark D. Aagaard, and Guang Gong

Electrical and Computer Engineering,
University of Waterloo

g37yang@uwaterloo.ca

July 21, 2015

Outline

Outline

Lightweight Cryptography in Passive RFID Systems

- Lightweight cryptography is used for these highly constrained devices (such as passive RFID tags and WSN nodes), i.e., the area should be less than 2000 GEs.
- The typical passive RFID systems include three parts: readers, tags, and database.
- The tiny and inexpensive properties of such RFID systems mean that the tags have very limited power consumption, constrained memory and computing capability.
- The pseudorandom numbers are used frequently in the current EPC Class 1 Generation 2 RFID systems and will also play a critical role in the future passive RFID standards.

The Random Numbers in the Passive RFID Systems

- RN16 (16-bit random number) is used in many commands of the EPC RFID systems. It is mainly used for providing verification of the reader identity for the tags, and providing cover-code (mask) for the data in access,kill,and write commands.

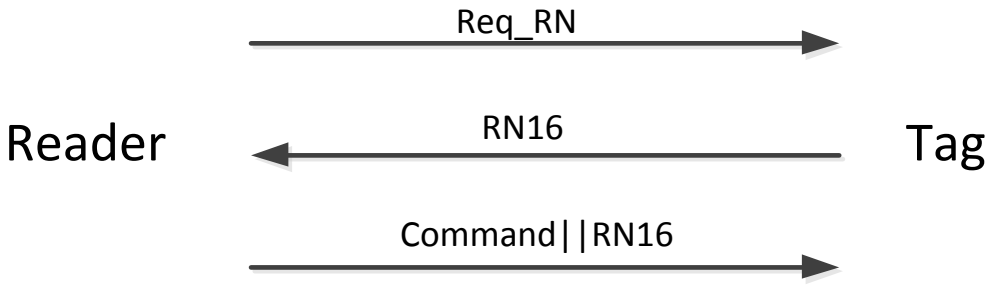


Figure 1: The RN16 Used for Verification of the Reader Identity.

- The random numbers can also be used in the future security extensions of the EPC Class 1 Generation 2 standard.
  - Used in the challenge-response based mutual authentication protocols between the readers and tags.

The Lightweight Pseudorandom Number Generators (PRNGs)

- LAMED is designed based on registers, arithmetic logic unit (ALU), XOR and modular operations. Its estimated area is 1585 GEs.
- Melia-Segui *et al.*'s PRNG and J3Gen rely on the security of linear feedback shift registers (LFSRs) and a truly random number generator (TRNG). The estimated area of Melia-Segui *et al.*'s PRNG is 761 GEs, and the estimated area of J3Gen with an internal state size 64 is 1419 GEs.
- Warbler is designed by using the properties of nonlinear feedback shift registers (NLFSRs) and the WG-5 transformation modules.
- AKARI1B is designed based on the T-function and a non-linear filter function, and the area before the place and route phase for AKARI1B with an internal state size 64 is 1749 GEs in the UMC Faraday 90nm technology.
- The estimated areas (no actual hardware implementations) of the four PRNGs (LAMED, Melia-Segui *et al*'s PRNG, Warbler, and J3Gen) are all below 2000 GEs, the maximum area limit for resource constrained applications.

The Warbler Pseudorandom Number Generator

- The sequences generated by Warbler can pass the EPC C1 G2 standard's statistical tests as well as the NIST randomness test suite.
- This sequence has guaranteed randomness properties, such as period and linear span.
- Warbler has been proved to be sufficiently secure in the EPC C1 G2 RFID systems.

This Work

In this work, we pay attention to the low-area implementation of Warbler in CMOS 65nm and CMOS 130nm ASICs, and provide the area, maximum clock frequency, and total power consumption results.

- We can achieve areas of 498 GEs and 534 GEs after the place and route phase in the CMOS 65nm and 130nm ASICs respectively.
- The area of our Warbler implementation is smaller than the estimated areas of LAMED, Melia-Segui *et al.*'s PRNG, and J3Gen, and also smaller than the areas of AKARI1B, Grain, Trivium, SIMON, SPECK, PHOTON-80/20/16, and SPONGENT-88.
- Two design options. The LFSR counter-based design is better than the binary counter-based one in terms of area and power consumption.

Outline

The Description of Warbler

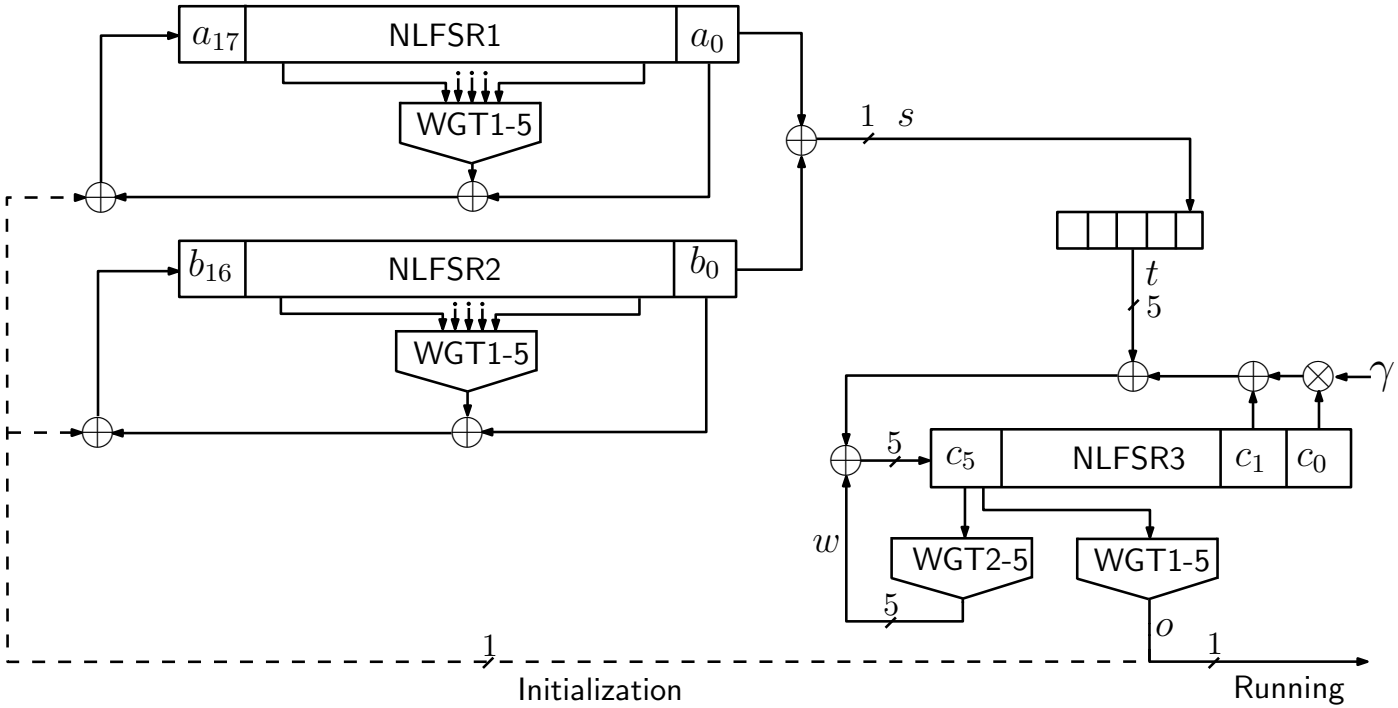
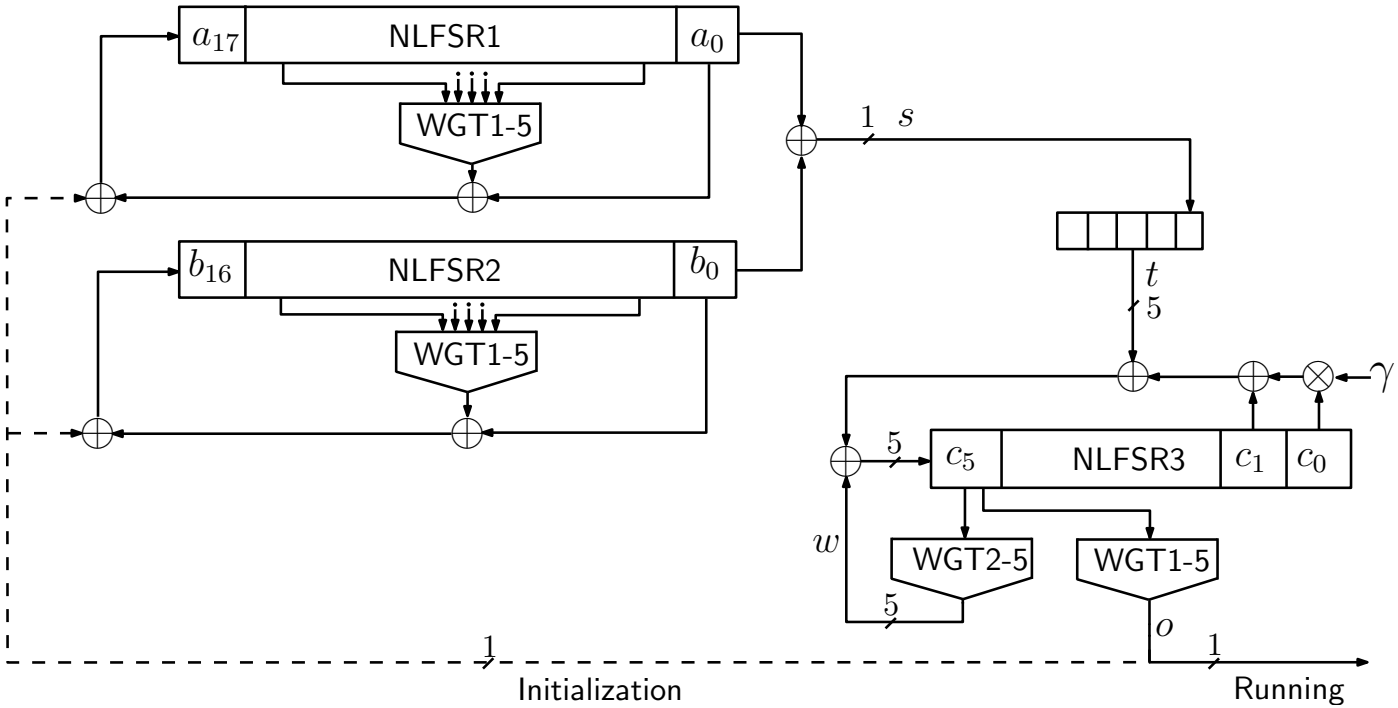


Figure 2: The Initialization and Running Phases of Warbler

- Warbler is mainly built upon three NLFSRs and four WG-5 transformation modules.
  - WGT1-5 module: WGT-5(x^3), the WG-5 transformation with decimation 3.
  - WGT2-5 module: WGT-5(x), the WG-5 transformation with decimation 1.
- Warbler has an internal state of 65 bits: a 45-bit Key and a 20-bit IV.

The Behavior of Warbler

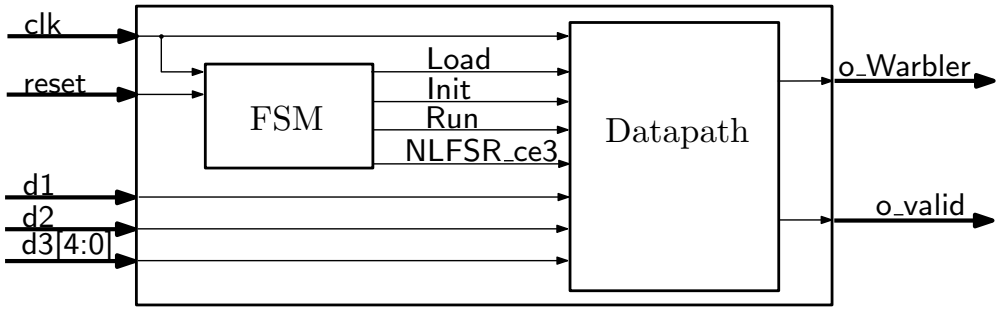


- Load the Key and IV (18 clock cycles).
- The output of the WGT1-5 module in NLFSR3 is used to feed back to the inputs of NLFSR1 and NLFSR2 in the 36-round initialization phase not in the running phase.
- t_k can be obtained by every five clock cycles from the 5-bit shift register, which results in a $1/5$ (i.e., 1-bit per five clock cycles) throughput of the Warbler output sequence o_{k+1}, $k \geq 35$.

Outline

Entire Architecture of Warbler

- The Top-level Architecture.



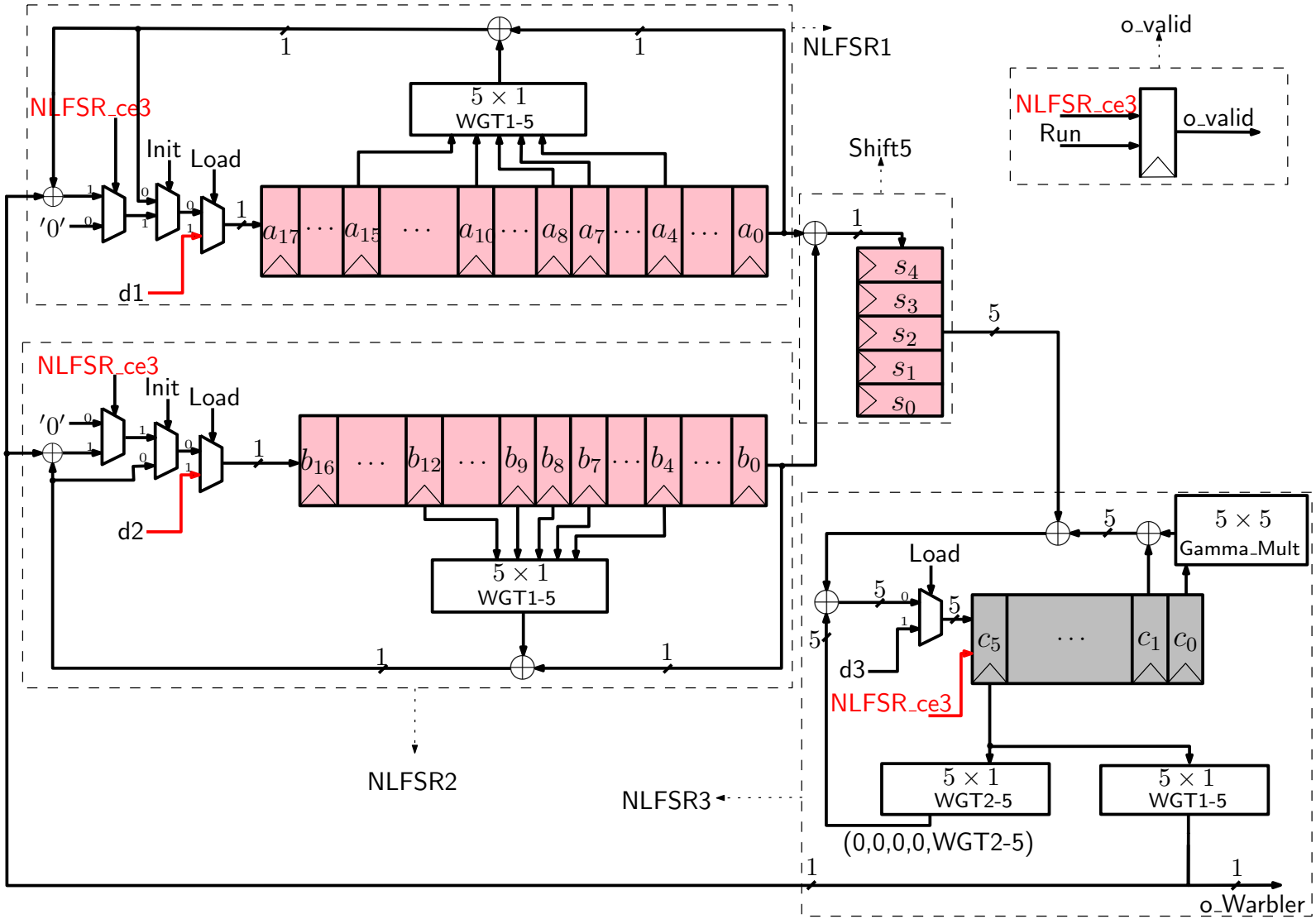
- The FSM.
  - Our FSM has three states: loading, initialization, and running.
  - NLFSR1 and NLFSR2 always run after reset, which makes them use only the standard registers without chip-enable signals.

Two Design Options for the Counter

- Two design options for the counter: binary counter and LFSR counter.
  - LFSR counter is designed by using a primitive polynomial ($X^6 + X + 1$) with an initial value (1, 1, 1, 1, 1, 1).
  - The feedback logic of the LFSR counter is smaller than the full-adder of the binary counter.
  - Different states transition conditions affect the area.
- States Transition Conditions for our FSM.

States	**Binary counter-based**	**LFSR counter-based**
Loading (100) $\rightarrow$ initialization (010)	17	17
Initialization (010) $\rightarrow$ running (001)	35	39

The Datapath



The Implementations of Functions in Finite Field

- Logical equation.
  - 2-input AND gate.
  - o <= a(1) and a(2);
- Constant array.
  - constant and2: std_logic_vector(0 to 3) := (‘0’, ‘0’, ‘0’, ‘1’);

The Implementations of Gamma_Mult Module

- The Gamma_Mult module is used for the calculation of γc_k in $\mathbb{F}_{2^5}$. Under the polynomial basis representation, the element $X \in \mathbb{F}_{2^5}$ multiplied by $\gamma = \alpha^{15}$ can be computed as follows:

$$\begin{aligned} X \cdot \alpha^{15} &= (x_0 + x_1\alpha + x_2\alpha^2 + x_3\alpha^3 + x_4\alpha^4) \cdot \alpha^{15} \\ &= x_0\alpha^{15} + x_1\alpha^{16} + x_2\alpha^{17} + x_3\alpha^{18} + x_4\alpha^{19} \\ &= (x_2 + x_4) + (x_2 + x_3 + x_4)\alpha + (x_0 + x_3 + x_4)\alpha^2 + \\ &\quad (x_0 + x_1 + x_2)\alpha^3 + (x_1 + x_3 + x_4)\alpha^4. \end{aligned}$$

- Thus, Gamama_Mult module can be implemented using the logical equation method.

The Implementations of WGT1-5 and WGT2-5 Modules

- We can compute WGT-5(x^3) and WGT-5(x) in polynomial basis for every $x \in \mathbb{F}_{2^5}$ by using the logical equation or pre-storing them to two constant arrays (WGT1-5 and WGT2-5 respectively).
- The hardware implementations of WGT1-5 module and WGT2-5 module are more efficient if the constant array method is used rather than the logical equation method *.

*G.Yang *et al*, "Design space exploration of the lightweight stream cipher WG-8 for FPGAs and ASICs", WESS 2013.

Table 1: Comparison of Hardware Implementations of Lightweight Primitives.

	Algorithms	Key Size	IV/ Block Size*	Internal State Size	Area (GEs) Before P & R	Area (GEs) After P & R	Max Frequency (MHz)	Throughput @100KHz (Kbps)	Total Power	Tech (nm)
PRNG	Warbler (LFSR counter)	45	20	65	**491**†	**534**‡	250	20	0.296 μW @100KHz	130
	Warbler (Binary counter)				**500**†	**543**‡	270		0.298 μW @100KHz	
	Warbler (LFSR counter)	45	20	65	**464**†	**498**‡	1430	20	1.239 μW @100KHz	65
	Warbler (Binary counter)				**475**†	**511**‡	1370		1.274 μW @100KHz	
	LAMED	32	32	64	1585△	–	–	–	–	–
	Melia-Segui *et al.*	16	0	16	761△	–	–	–	–	–
	J3Gen	64	0	64	1419△	–	–	–	–	–
	AKARI1B	–	–	64	1749†	–	–	14.2	0.182 μW▽ @100KHz	90
Stream Cipher	Grain	80	64	160	–	1259‡	–	100	0.78 μW @100KHz	130
	Trivium	80	80	288	–	2088‡	–	100	1.44 μW @100KHz	130
	Grain	80	64	160	–	1126‡	1020	100	2.04 mW @1020MHz	65
	Trivium	80	80	288	–	1986‡	962	100	3.88 mW @962MHz	65
Block Cipher	SIMON◇	64	32	96	523†	–	–	5.6	–	130
	SPECK◇	64	32	96	580†	–	–	4.2	–	130
Hash Function	PHOTON-80/20/16	–	–	100	865†	–	–	2.82	–	180
	SPONGENT-88	–	–	88	738†	–	–	0.81	1.57 μW @100KHz	130

* IV is for PRNGs and stream ciphers, and Block is for block ciphers.
– The corresponding value is not related or not provided by the authors.
△ The estimated area.
▽ The estimated power consumption in UMC Faraday 90nm library.
◇ The smallest one in the SIMON and SPECK families.

Results Analysis

- Breakdown of the implementation results of warbler before the place and route phase.

Components			CMOS 65nm		CMOS 130nm	
			Binary counter-based (GEs)	**LFSR counter-based** (GEs)	**Binary counter-based** (GEs)	**LFSR counter-based** (GEs)
FSM	State transitions + chip_enable logic		35.25	39.00	36.75	37.50
	Counter		47.50	39.25	51.75	39.50
Datapath	NLFSR1	18-stage register	67.50		76.50	
		WGT1-5	15.50		14.50	
		Other feedback logic	10.00		9.50	
	NLFSR2	17-stage register	63.75		72.25	
		WGT1-5	15.50		14.50	
		Other feedback logic	10.00		9.50	
	NLFSR3	6-stage register (30 registers)	150.00		180.00	
		Gamma_Mult	13.75		14.00	
		WGT1-5	15.50		14.50	
		WGT2-5	9.25		9.50	
		Other feedback logic	32.50		30.75	
	Shift5	5-stage register	18.75		21.25	
		Other combinational logic	8.00		8.00	
	O_valid		6.50		6.50	
Totals	Compile simple		519	515	570	558
	Compile ultra		505	493	555	541
	Compile ultra + clock gating		475	464	500	491
	Compile ultra + clock gating (after place & route)		511	498	543	534

Results Analysis

- The LFSR counters are 17.4% and 23.7% smaller than the binary counters in CMOS 65nm and CMOS 130nm respectively. However, the total areas of the FSM are only 5.5% and 13% smaller accordingly.
- NLFSR1, NLFSR2, and Shift5 are indeed implemented by the standard registers without chip-enable signals.
- The WGT1-5 and WGT2-5 modules are different in the same technology.
  - They are 15.50 GEs and 9.25 GEs respectively in CMOS 65nm, and 14.50 GEs and 9.50 GEs respectively in CMOS 130nm.
  - The specific contents of the WGT1-5 and WGT2-5 look-up tables:

Address	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
WGT1-5	0	0	1	0	0	1	1	1	0	1	1	1	1	0	0	0	1	0	1	0	0	0	1	1	0	1	1	0	0	1	0	1
WGT2-5	0	0	0	0	0	1	0	0	0	0	0	1	1	1	1	1	1	1	0	1	1	1	0	0	1	0	0	1	1	1	0	1

Outline

Conclusion

- The area of our Warbler implementation is smaller than the estimated areas of LAMED, Melia-Segui *et al.*'s PRNG, and J3Gen, and also smaller than the areas of AKARI1B, Grain, Trivium, SIMON, SPECK, PHOTON-80/20/16, and SPONGENT-88.
- We have presented two design options (binary counter-based and LFSR counter-based designs) to efficiently implement Warbler in CMOS 65nm and CMOS 130nm ASICs.
- Our results showed that the LFSR counter-based design is better than the binary counter-based one in terms of area and power consumption.
- Our analysis has verified that the areas of NLFSRs and combinational logic are dependent upon the type of registers and the adopted technologies.
- In addition, we demonstrated that the areas of WG-5 transformation look-up tables depend on the specific decimation values.

Thanks very much!

Q & A?