On Comparison of Configurable Encoders in Xilinx and Altera FPGAs

Petr Pfeifer, Farnoosh Hosseinzadeh, Heinrich Theodor Vierhaus

Brandenburg University of Technology Cottbus-Senftenberg, Faculty 1 (MINT), Germany

email: {petr.pfeifer | farnoosh.hosseinzadeh | heinrich.vierhaus}@b-tu.de

Abstract – **Encoders using generator polynomials and linear-feedback shift registers are the key parts of communication technologies widely used in most of today's integrated as well as field systems. This paper presents a detailed comparison of three ways of implementation of configurable encoders arranged in PENCA and implemented in Xilinx and Altera FPGAs.**

Keywords—communication dependable systems; PENCA; LFSR; re-configurable encoder; RLUT; SLICEM; Zynq; reconfiguration; Artix; CycloneV; Ultrascale; FPGA.

I. INTRODUCTION

Communication technologies themselves have experienced an explosive growth over the last quarter of century. There are many new technologies developed and used actively in wide communication networks and internet technologies, but also in short-distance connections. Modern systems gain from complex architectures and sophisticated algorithms; however, many industrial systems need very high levels of dependability. Reliability and fulfilment of safety requirements are crucial. The general data transmission, transmitter / receiver systems consists of digital circuit in the signal path, digital sub-circuits off-path for encoding and decoding from bits to chips and reverse, and, of course, analogue and also RF parts in wireless systems [1]. Clearly, an error detection and correction algorithm, block or circuit is the very basic and crucial building component in all such systems. However, algorithms implemented in software are typically slow and with strongly variable processing time. Hardware implementations can use some fixed approach and dedicated circuits, or fully programmable circuits and flexible algorithms can be implemented. It is the case of our advanced industrial wireless communication system and architecture introduced already in [2,3], and the PENCA (Programmable Encoder Array) architecture in [4,5].

II. MOTIVATION

Field programmable gate array (FPGA) have enjoyed continuous improvements in all their metrics also due to technology scaling, new micro- and nano-circuits and architectural advances since the first commercial release in 1985. Implementation of FEC (Forward Error Correction) algorithms and LFSRs in FPGAs is a very standard approach, further enhanced using PENCA and its architectural advantages, creating a programmable, universal and dependable platform for baseband applications not only in communication systems. Since both the two leading FPGA platforms are used in our project, what is the optimal implementation and performance in Xilinx and Altera FPGA hardware and architectural platforms with respect to the resources and impact to the system latency caused by the updates of the encoder itself?
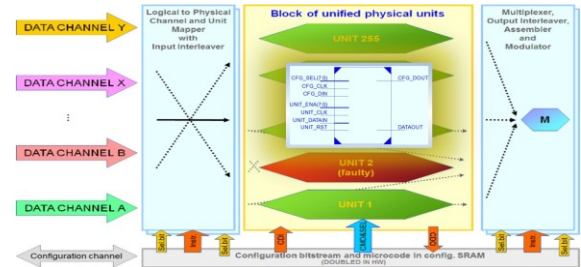
Fig. 1. Basic scheme of the basic dependable PENCA architecture (no honeycomb architecture used), with a detail of a PENCA frame in a block of unified configurable units in the array.

III. PENCA ARCHITECTURE AND ENCODERS

The PENCA architecture is based on an array of run-time programmable LFSRs (Figure 1), supporting a programmable mixture of various independent data sources, flexibly combining them and forming desired areas in the final virtual pool of bits and adding special data or parity bits by supporting many block codes and error detection or correction algorithms. Each channel can perform fully programmable polynomial multiplication and division. Together with a programmable counter belonging to each programmable LFSR, each PENCA channel perform a general data encoding and also selected testing tasks. Very low system latency, testability and dependability is the key in dependable industrial systems. For the implementation of forward error correction schemes, we can distinguish between a few basic schemes and implement or use Hamming [6], Hsiao [7], Reed-Solomon [8], BCH [9], extended code [10,11], and other codes [12,13,14]. Hence test, configuration, error detection and correction, and design of such systems in general is an extremely complex task. In addition, the new system developed requires a very high level of flexibility not offered by standard architectures and implementations. PENCA is also used for on-line test purposes of all the baseband system chain, since consisting of many independent programmable paths and LFSRs, even performing primarily BCH encoding task, issuing tests and processing also data form used for internal test procedures. PENCA details can be found in [4,5].

TABLE I. SUPPORTED BCH ERROR CORRECTION BLOCK CODES

Total bits	Payload bits	Code Rate	Bit errs corr.	Gener. polyn. order	Total bits	Payload bits	Code Rate	Bit errs corr.	Gener. polyn. order
7	4	0,571	1	3	255	247	0,969	1	8
15	11	0,733	1	4	255	239	0,937	2	16
15	7	0,467	2	8	255	231	0,906	3	24
15	5	0,333	3	10	255	223	0,875	4	32
31	26	0,839	1	5	255	215	0,843	5	40
31	21	0,677	2	10	255	207	0,812	6	48
31	16	0,516	3	15	255	199	0,780	7	56
31	11	0,355	5	20	255	191	0,749	8	64
31	6	0,194	7	25	511	502	0,982	1	9
63	57	0,905	1	6	511	493	0,965	2	18
63	51	0,810	2	12	511	484	0,947	3	27
63	45	0,714	3	18	511	475	0,930	4	36
63	39	0,619	4	24	511	466	0,912	5	45
63	36	0,571	5	27	511	457	0,894	6	54
63	30	0,476	6	33	511	448	0,877	7	63
63	24	0,381	7	39	511	439	0,859	8	72
127	120	0,945	1	7	1023	1013	0,990	1	10
127	113	0,890	2	14	1023	1003	0,980	2	20
127	106	0,835	3	21	1023	993	0,971	3	30
127	99	0,780	4	28	1023	983	0,961	4	40
127	92	0,724	5	35	1023	973	0,951	5	50
127	85	0,669	6	42	1023	963	0,941	6	60
127	78	0,614	7	49	1023	953	0,932	7	70
					1023	943	0,922	8	80

The encoder used in the industrial system supports all Hamming and BCH codes with packet length from 7 up to 1023 data bits and with up to 8 bits error detection and correction capability, as shown in Table I. It means the generator polynomial should be up to the order of 80, hence 80 DFFs must be reserved per each unit in the FPGA. The code rate is k/n, for every k bits of useful payload information, the encoder generates totally n bits of data as the total bits, of which $n-k$ are parity bits. A block code shortening of all the block codes is supported as well.

IV. PENCA ARRAYS AND CONFIGURABLE LFSRS - - THE THREE WAYS OF IMPLEMENTATION

The idea of a fully programmable architecture is not completely new one. There are many patents and papers already published or available, see e.g. [15,16,17]. Area-efficient encoders are typically based on Linear Feedback Shift Registers. The PENCA architecture (used in encoder as well as decoder) is based on the array of configurable encoder where each configurable LFSR and counter is configured in a given way in order to perform the desired data encoding algorithm and function. In our case of the following experiments, all the unit contain the same circuit of selected type of configurable LFSR. In order to do the best comparison of our experiments, all the PENCA units also have the same parameters. It means no a honeycomb architecture as introduced in [4] and utilizing also neighbouring units was used in our experiments. The PENCA array is ready for 255 units using 8 bits for the address buses. Both the configuration and communication working data paths are separated, hence one unit can perform the desired task and another unit can be reconfigured at the same using also completely different clock frequencies.

In general, there are the following basic ways of implementation of configurable LFSRs in today's FPGAs: a) a classic, conventional way using programmable logic elements and muxes, b) using reconfigurable LUTs in SLICEM and configuring them directly instead of using auxiliary configuration elements, c) and using partial reconfiguration.
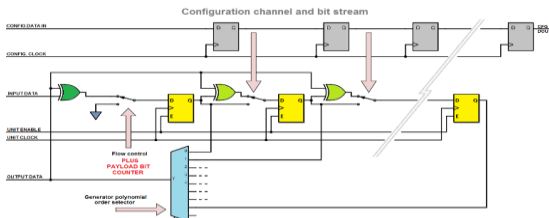

Fig. 2. The basic principle of a conventional configurable LFSR.

A. Conventional configurable encoders

The classic and widely used way of implementing configurable encoder (Fig. 3) is based on two paths: a series of configuration storage elements (shift registers) keeping the information for each configuration element of a standard LFSR creating the desired generator polynomial. In general, the generator polynomial has 2 basic parameters: its order (number of parity bits) and its coefficients. The order of generator polynomial is selected by a multiplexer as the desired length of the series of flip-flops and XOR gates. The hardware must also contain XOR gates and mux selectors for each coefficient of the

generator polynomial. The configuration of all these points must be controlled by the storage elements. In our case, the configuration chain has 97 flip-flops, where 10 are dedicated for the payload counter, 7 for the generator polynomial order mux selector, and 80 control bits per each coefficient controlling the XOR gates and forming the desired generator polynomial. The circuit area overhead in this case comes mainly from the additional configuration chain of DFFs. A programmable XOR gate is created by a single LUT, which is only a slightly slower than a single XOR gate. The main delay and performance penalty is caused by the multiplexer performing the programmable order of the polynomial.
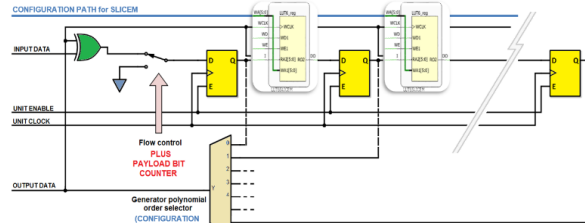

Fig. 3. A configurable LFSR using SLICEM.

B. Using RLUTs in SLICEMs

Xilinx FPGAs contain SLICEL and SLICEM configuration units [18], where SLICEL has LUTs only, and SLICEM with reconfigurable ones enabling implementation of a distributed memory. SLICEM can significantly reduce the FPGA resources required for implementation of shift registers [19]. In the SLICEM circuits (Fig. 3), the content and therefore the logic function of these 6-input LUTs is not fixed by the FPGA configuration bit stream, and its function can also be controlled from the FPGA area logic during run time. There are also key architectural changes in the Xilinx Ultrascale FPGA generation [20], especially in the connection of flip-flops to the outputs of LUTs. Although this SLICEM approach has some architectural advantages and is sometimes discussed as RLUT (Reconfigurable LUT), and it can result in much more efficient utilization of this configurable logic block resources as shown in figure 4, it is near impossible to find any reference on this theme with respect to the LFSR. Obviously, the circuit area overhead consisting mainly from the additional configuration chain of DFFs is replaced by a 9-bit wide configuration path plus LUT configuration write enable decoders (compressed to a single bit input with clock in the final PENCA block). A programmable XOR gate is directly created by the SLICEM programmable LUT. Unfortunately, one additional LUT at the top of each SLICEM is typically left unused due to the first LUT shared data address write signals. In our case, most of such LUTs are also utilized and successfully forming the configuration decoders. The main delay is again caused by the multiplexer performing the programmable order of the generator polynomial. Since the level and ease of configurability of this circuit is much higher than the previous case A), it may open doors for config. faults or unwanted functionality, including hardware Trojans [21].
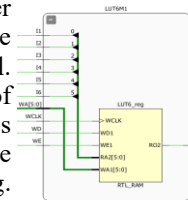

Fig. 4. An RLUT in a SLICEM.

C. Using partial reconfiguration

FPGAs can change their functionality in its parts [22]. It is achieved by changing the configuration bit stream locally, it means using partial reconfiguration [23], especially in Xilinx FPGAs [24]. Selected physical areas of FPGAs are reserved for the reconfiguration task, and PENCA designs fit right in this area. However, there is some amount of FPGA resources required to perform the reconfiguration task itself. This area overhead may be very big for small designs, and this fact must be considered. On the other hand, the encoder's LSFR itself doesn't contain any configuration interface used before, since all the desired function of the encoder is simply given and fixed by the reconfiguration bitstream (figure 5). It means that a higher number of partial bitstreams related to the block code's encoders is required.
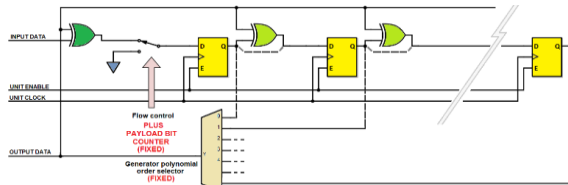

Fig. 5. The basic principle of the LFSR design using partial reconfiguration.

The presence of XOR gates at all the desired locations is directly encoded in the partial bitstreams. Since Altera FPGAs doesn't offer such reconfiguration options, we have tested our approach only on Xilinx Zynq XC7Z020 SoC containing Artix type of FPGA. The total number of occupied SLICEs is 919, where 252 SLICEs are dedicated for the partially reconfigured area and it is linked to the PENCA design only, and remaining about 677 SLICEs are required only for FPGA resources performing the partial reconfiguration task (reconfiguration engine overhead and AXI bus connection). The configuration time scales almost linearly as the configuration bitstream grows with the number of configurable frames with small variances depending on the location and the content of the configuration frames. We have used the PCAP interface. It is 32-bit wide and clocked at 100 MHz. We have used the standalone version. The full bitream for XC7X020 part is 4,045,564 Bytes and it takes about 32 ms to reconfigure the entire FPGA. The reconfiguration time for the partial bitstream of 162,696 Bytes used in our case of the discussed array of 15 PENCA units is about 1.2 ms. This source file compressed using a standard ZIP file compression method fits into about 17 KB (about 1 KB per one encoder unit). Each file contains only one selected single generator polynomial and encoding method, or their combinations for multiple units. The bitstream of the empty reconfigurable area with no any design can be compressed to 768 Bytes (a standard ZIP file again).

V. IMPLEMENTATION, RESULTS AND DISCUSSION

Were possible we have implemented the proposed PENCA solution in Altera and Xilinx 28 nm SoCs, widely used and also cost-efficient solutions: Xilinx Zynq [25] XC7Z020-1-CLG484, manufactured using TSMC's 28 nm high performance low power process, combining Artix FPGA and ARM hardware processor into SoC technology solution, and used also on the Zedboard development kit [26]. We have used the very last available ISE software version 14.7 and Vivado 2015.3. A direct competitor is Altera CycloneV [27] 5CSXFC6D6F31C6, made also on TSMC's 28 nm, but low-power (28LP) process, combining also FPGA and ARM and used on SoCkit. Quartus Prime Lite 15.1.0 Build 10/21/2015 was used. A comparison of both the families can be found in [28]. Some experiments were performed also on KCU105 development kit [29], containing Ultrascale XCKU040-2FFVA1156E FPGA [30] and having obvious architectural advantages. Source codes were generated in VHDL by our PENCA generator.

The classical version of configurable LFSR requires 91 core DFFs plus 97 DFFs keeping the configuration. Even BCH codes in Table I. do not require all 80 XOR gates to be all programmable ones, the entire LFRS is programmable in all its parts. Hence, the sum of DFFs is 188 per unit, or 2820 in total for 15 units in a typical size and configuration. All 255 PENCA units utilizes the Xilinx FPGA at 93% (near full FPGA), while Altera is at 54%. Altera design is obviously slower than Xilinx, especially the configuration clock can be up to 1169 MHz in Xilinx and only about 600 MHz in Altera FPGA. Ultrascale required about 31 CLBs per unit. Altera FPGA uses the resources in more efficient and predictable way.

TABLE II. CLASSICAL CONFIGURABLE ENCODER – XILINX

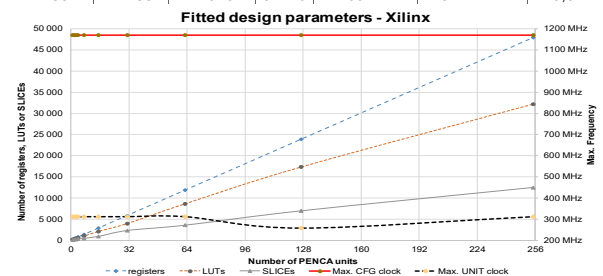| PENCA units	SLICEs	Registers	LUTs	Max. CFG clock	Max. UNIT clock	SLICEs per unit
1	70	188	132	1169 MHz	312 MHz	70,0
2	131	376	263	1169 MHz	312 MHz	65,5
3	199	564	410	1169 MHz	312 MHz	66,3
4	253	752	536	1169 MHz	312 MHz	63,3
7	411	1316	959	1169 MHz	312 MHz	58,7
15	852	2820	2085	1169 MHz	312 MHz	56,8
31	2262	5828	3960	1169 MHz	312 MHz	73,0
63	3525	11844	8634	1169 MHz	312 MHz	56,0
127	6935	23876	17342	1169 MHz	256 MHz	54,6
255	12488	47940	32219	1169 MHz	312 MHz	49,0

Fig. 6. Fitted design results - XILINX

TABLE III. CLASSICAL CONFIGURABLE ENCODER – ALTERA

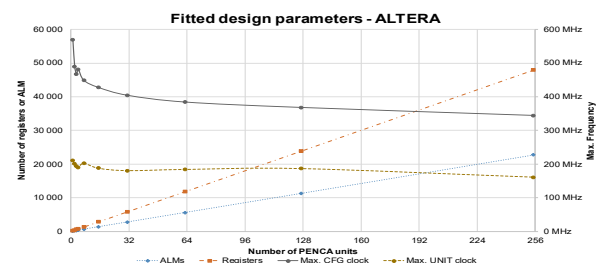| PENCA units	ALMs	Registers	Max. CFG clock	Max. UNIT clock	ALMs per unit
1	89	188	569 MHz	211 MHz	89,0
2	180	376	489 MHz	202 MHz	90,0
3	273	564	467 MHz	195 MHz	91,0
4	365	752	481 MHz	190 MHz	91,3
7	621	1316	449 MHz	203 MHz	88,7
15	1348	2820	428 MHz	188 MHz	89,9
31	2761	5828	404 MHz	181 MHz	89,1
63	5598	11844	384 MHz	184 MHz	88,9
127	11283	23876	368 MHz	187 MHz	88,8
255	22704	47940	344 MHz	161 MHz	89,0

Fig. 7. Fitted design results – ALTERA
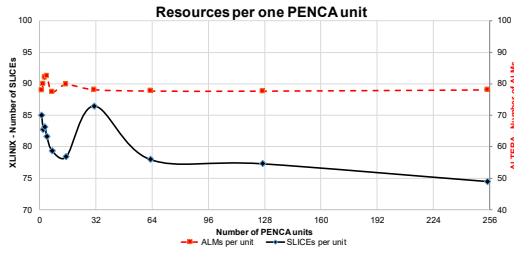
Fig. 8. Fitted design – Comparison Xilinx and ALTERA

For configurable encoders using RLUTs in SLICEM, all the units implemented are designed using distributed memories. It means that 79 SLICEMs are required just only for the generator polynomial coefficients, 13 SLICEMs for programmable order and 2 for the programmable 10-bit counter. Hence 94 configuration RLUT circuits in SLICEMs should be used in this version and it requires 6016 bits to be stored for each encoding algorithm and generator polynomial (Table IV and Fig. 9). The solution with 63 units cannot fit into the selected Xilinx FPGA, hence we end with 31 units. Each single PENCA unit implemented in Ultrascale FPGA requires 102 to 117 CLBs (approx. 3.3x more to the classical version, 40% savings to Artix). The same design fitted in a similar way into Altera FPGA requires a huge number of 2663 registers, 2128 ALMs and block memories.

TABLE IV. SLICEM CONFIGURABLE ENCODER – XILINX

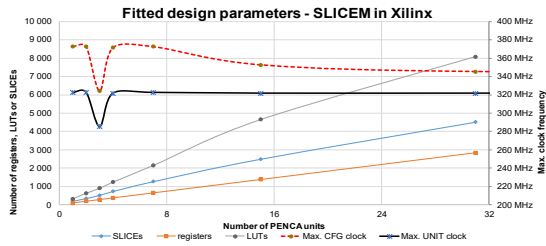| PENCA units	SLICEs	Registers	LUTs	Max. CFG clock	Max. UNIT clock	SLICEs per unit
1	193	99	319	373 MHz	322 MHz	193,0
2	347	190	622	373 MHz	322 MHz	173,5
3	515	281	898	324 MHz	285 MHz	171,7
4	731	372	1232	372 MHz	322 MHz	182,8
7	1267	645	2146	373 MHz	322 MHz	181,0
15	2494	1373	4651	353 MHz	322 MHz	166,3
31	4517	2829	8090	345 MHz	322 MHz	145,7

Fig. 2. Fitted design results for SLICEMs in Xilinx

The partial reconfiguration version of configurable LFSR requires 10 DFFs for a 10-bit counter, 1 loop lock flip flop, 80 DFFs ready for the generator polynomials, and no any additional DFFs keeping the configuration, since all the generator polynomial coefficients are hardwired in the configuration bit stream. It results in 91 DFFs per unit, or 1365 in total for 15 units. This design requires 252 SLICEs (1365 registers and 833 LUTs) when fitted outside of the reconfiguration area, it means the reconfiguration area forces the design system to be a bit more efficient in this case. The maximal unit clock achieved is 642 MHz. Artix architecture has 8 DFFs in each SLICE, 259 occupied SLICEs and 1365 required registers leads to 5.27 DFFs utilized in average per SLICE, it means 65.9% of resources. It is also about 17.3 SLICEs per unit. Since Altera FPGAs does not support this way of partial reconfiguration, its only for information that the same source design fitted into Altera FPGA achieved 409 MHz maximal clock speed (+64%) and occupied 716 ALMs. The number of registers is naturally the same. Since CycloneV has 4 DFFs per ALM, 716 ALMs per 1365 registers result in utilization of 1.9 register per ALM, it means 47.7% of DFFs only and about 47.7 ALM per channel. The same design fitted into Ultrascale FPGA runs at 568 MHz (+39%) and requires only 200 CLB (77% or -23%). The Ultrascale architecture requires about 13 CLBs per unit, 200 CLB (3200 DFFs) for 15 units, where only 1365 registers are used, it means 42.7%.

CONCLUSION

We have presented a detailed comparison of fully configurable multichannel encoder architectures, utilizing multiple algorithms and based on programmable LFSRs implemented in Altera and Xilinx FPGAs. All approaches can be combined to reach the best performance and area parameters.

REFERENCES

[1] B. P. Lathi, Zhi Ding, "Modern Digital and Analog Communication Systems", Oxford Univ. Press; 4th edit., 2009.
[2] P. Pfeifer, C. Gleichner, and H.T. Vierhaus, "Flexible Test, Error Detection and Correction in Dependable Communication Systems incl. Results on 28 nm Xilinx and Altera FPGAs", DDECS2016, pp.26-31.
[3] P. Pfeifer, F. Hosseinzadeh, H.T. Vierhaus "On Comparison of Robust Configurable FPGA Encoders for Dependable Industrial Communication Systems," Int. On-Line Test Symposium IOLTS2017.
[4] P. Pfeifer, H. T. Vierhaus : "A New Area-efficient Reconfigurable Encoder Architecture for Flexible Error Detection and Correction in Dependable Communication Systems", 15th Biennial Baltic Electronics Conference (BEC2016), Tallinn, Estonia, October 3-5, 2016, pp. 87-90.
[5] P. Pfeifer, H. T. Vierhaus : "Reconfiguration Aspects of PENCA - An Area-efficient Reconfigurable Encoder Architecture with Built-in Security Features for Flexible Error Detection and Correction in Robust Dependable Communication Systems", PDeS2016, pp. 381-386.
[6] R. W. Hamming, "Error detecting and error correcting codes", The Bell System Technical Journal, Volume 29, No.2, pp. 147-160, April 1950.
[7] M. Y. Hsiao., "A class of optimal minimum odd-weight-column SEC-DED codes". IBM Journal of R&D, Vol.14, I.4. July 1970, pp. 395-401.
[8] I. S. Reed, "A class of multiple-error-correcting codes and the decoding scheme". In: Technical Report 44 (Oct.1953). MIT Lincoln Laboratory.
[9] R.C. Bose, D.K. Ray-Chaudhuri, "On a class of error correcting binary group Codes," Inf. Control, 3, pp. 68-79, March 1960.
[10] E. R. Berlekamp, "Algebraic coding theory," McGraw-Hill, NY, 1968.
[11] R. T. Chien, "Cyclic decoding procedure for the Bose-Chaudhuri-Hocquenghem codes," IEEE Trans. Inf. Theory, IT10, October 1964.
[12] E. Fujiwara,"Code Design for Dependable Systems," Wiley&Sons, 2006.
[13] C. Badack, T. Kern und M. Gössel, "Modified DEC BCH Codes for Parallel Correction of 3-bit Errors Comprising a Pair of Adjacent Errors", IOLTS 2014, pp. 116-121.
[14] B. Varghese u. a. Multiple bit error correction for high data rate aerospace applications, In: IEEE Conference on Information Communication Technologies (ICT). Apr. 2013, pp. 1086-1090.
[15] H. Yoo, Y. Lee, and IC. Park , "7.3 Gb/s Universal BCH Encoder and Decoder for SSD Controllers", 19th ASP-DAC, Singapore 2014, pp.37.
[16] H. Tang, G. Jung, and J. Park, "A hybrid multimode BCH encoder architecture for area efficient re-encoding approach", ISCAS2015.
[17] M. Wang, N. Deng, H. Wu, and Q.He, "Theory study and implementation of configurable ECC on RRAM memory", 15th NVMTS, Beijing, October 2015, pp. 1-3.
[18] Xilinx, "7 Series FPGAs Configurable Logic Block: User Guide", UG474 (v1.8) September 27, 2016.
[19] Xilinx, "Saving Costs with the SRL16E", White Paper: Xilinx FPGAs, WP271 (v1.0) May 22, 2008.
[20] Xilinx, "Ultrascale Architecture Configurable Logic Block: User Guide", UG574 (v1.4) November 24, 2016.
[21] Debapriya et al., "Reconfigurable LUT: A Double Edged Sword for Security-Critical Applications", In book: Security, Privacy, and Applied Cryptography Engineering, pp. 248-268.
[22] Altera, "FPGA Run-Time Reconfiguration: Two Approaches", White paper, WP-01055-1.0, March 2008.
[23] Lie Wang, Feng-yan Wu, "Dynamic partial reconfiguration in FPGAs", 3rd Int. on Intelligent Information Technology Application, 2009.
[24] Christian Kohn, Xilinx, "Partial Reconfiguration of a Hardware Accelerator on Zynq-7000 All Programmable SoC Devices", 2015.
[25] Xilinx, Zynq-7000 All Programmable SoC Overview, DS190 2015.
[26] Digilent, "ZedBoard dev. kit", http://zedboard.org/product/zedboard
[27] Altera, Cyclone V Device Overview, CV-51001, Dec. 2015.
[28] Altera, Architecture Matters: Choosing the Right SoC FPGA for Your Application, White Paper, WP-01202-1.0, Nov 2013.
[29] Xilinx, "KCU105 Board: User Guide", UG917 (v1.6) March 31, 2016.
[30] Xilinx, "UltraScale Architecture and Product Overview", DS890, 2016.