High-Speed and Low-Power VLSI-Architecture for Inexact Speculative Adder

Rahul Shrestha, *Member*, *IEEE*
School of Computing & Electrical Engineering, Indian Institute of Technology (IIT) Mandi
rahul_shrestha@iitmandi.ac.in

***Abstract*—This paper presents carry-lookahead adder (CLA) based design of the contemporary inexact-speculative adder (ISA) which is fine grain pipelined to include few logic gates along its critical path and thereby, enhancing the frequency of operation. Additionally, various stages of the proposed ISA architecture has been clock gated to reduce the power consumed by this design. Functional verification and hardware implementation for various configurations of the suggested ISA is carried out on field-programmable gate-array (FPGA) platform. It could operate at a maximum clock frequency of 324 MHz which is 52% better then the conventional ISA. Thereafter, the synthesis and post-layout simulation of 32-bit proposed ISA is carried out using 90 nm complementary metal-oxide semiconductors (CMOS) technology node for power and area analysis. Our design occupied 5.11 mm^2 of chip area and consumed 9.68 mW of total power at 400 MHz clock frequency. The proposed ISA burns 52.8% lesser power than the state-of-the-art work.**

***Index Terms*—Inexact speculative adder, carry lookahead adder, pipelining, FPGA, very-large scale-integration (VLSI) and application-specific integrated-circuits (ASIC).**

I. Introduction

High-speed adders are highly desirable in the present day scenario, though power (or energy) and silicon area are equally vital. Spectrum sensors used in intelligent cognitive-radio environment [1], [2] as well as internet of everything (IoE) [3] devices focused on physical interfaces are largely-explored research areas in the recent time. Hardware for the algorithms of such applications is basically focused on sensing and actuating where the response time is key component to be optimized for real-time interfaces. Thereby, the design of highly optimized adders in terms of speed play significant role in the present era and hence this paper focuses in the design of same.

With tolerable degradation in accuracy and performance, it is feasible to conceive high-speed, low power and area efficient design using inexact and approximate circuit technique [4]. Accuracy of such circuits can be traded off to improve the power and speed by speculation. Thereby, such adders are referred as inexact speculative adder (ISA). Various optimized versions of such ISA have been reported in literature [5]-[9] and these works concentrated mostly on enhancing the accuracy of their results. However, there is space to further improve the speed of such adders by retaining the accuracy with minimum error. Thereby, our contributions in this work are as follows: design and analysis of the carry lookahead adder (CLA) based ISA has been carried out. Thereafter, this adder is fine grain pipelined to reduce the critical path delay that further enhances the operating speed. FPGA implementation of 8, 16 and 32 bit versions of the proposed ISA has been carried. Obtained post place-&-route results of these adders are compared with reported non-pipelined ISAs. Subsequently, clock signal fed to various stages of the deep pipelined ISA-architecture has been gated to reduce the power consumption. Eventually, ASIC synthesis and post-layout simulation of the proposed 32-bit ISA has been performed in 90 nm-CMOS technology node and is compared with the state-of-the-art ISA adder.

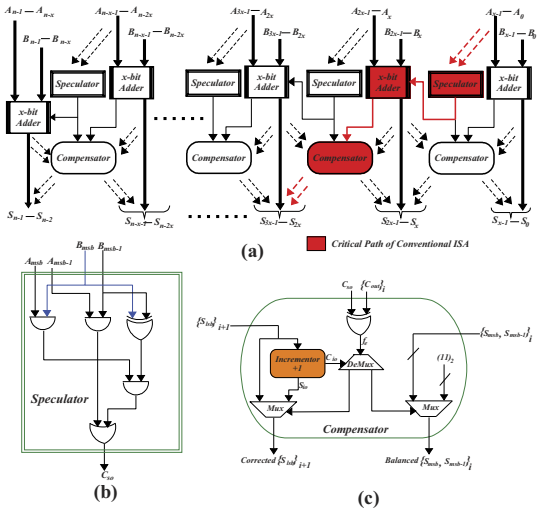

Fig. 1: (a) Basic block diagram of n-bit conventional inexact-speculation adder (ISA). (b) Gate-level circuit representation of speculator block. (c) Digital architecture of compensator block.

II. Proposed VLSI Architecture of ISA

Block diagram and data flow of conventional ISA for n-bit addition is shown in Fig. 1. In the proposed architecture, we have segregated the n-bit input into 4-bit blocks (i.e., the value of x = 4 in Fig. 1) and each of these blocks is fed as operands to the x-bit adder. Unlike the conventional ISA architecture, the adder unit has been replaced with 4-bit CLA to further enhance the speed of operation. Comprehensive explanation with circuit details of various sub blocks of this adder are presented as follows:

1) Speculator and Adder Blocks: Prior delving into the circuit details, it is necessary to understand the notations used in this paper. Two n-bit operands for addition are represented as $A = \{A_0, A_1, ...A_{n-1}\}$ and $B = \{B_0, B_1, ...B_{n-1}\}$; whereas, the sum, carry input and carry output are expressed as $S = \{S_0, S_1, ...S_{n-1}\}$, C_{in} and C_{out} respectively. Gate-level circuit diagram of the speculator used in our adder design is presented in Fig. 1(b). This block is based on CLA logic to speculate the output carry for each 4-bit adder block. Speculation is carried out for 'r' msb bits of each block where r is less than the size of block, (i.e., $r < x = 4$). Subsequently, the input carry for each speculator block is 0 (or 1) which introduces positive (or negative) errors respectively. The output carry, which is denoted as C_{so}, from each speculator block is fed as an input carry for the adder block succeeding it, as shown in Fig. 1. Now, each 4-bit adder block need not wait for the input carry from the preceding 4-bit adder block. Instead, all such adder blocks perform simultaneous

978-1-5090-3969-2/17/$31.00 ©2017 IEEE

additions on receiving input carries from the concerned speculator blocks. Speculator block computes carry based on the equation shown below:

$$\begin{aligned} P_i &= A_i \oplus B_i; \\ G_i &= A_i \cdot B_i; \\ C_{i+1} &= G_i + (P_i \cdot C_i); \end{aligned} \tag{1}$$

where $(i+1)^{th}$ carry bit is computed using the propagate (P_i), generate (G_i) and carry (C_i) of i^{th} bit. This block is situated along the critical path of ISA architecture; however, it doesn't produce much delay as it computes the carry for few bits. On the other hand, adder block performs addition of 4-bit input blocks using CLA logic based on the equation below.

$$S_i = P_i \oplus C_i. \tag{2}$$

This work employs CLA in the suggested architecture, as it has smaller propagation delay compared to the conventional ones. Here, the local sum obtained in parallel from each adder block is not the exact output because the addition has been performed using speculated carry inputs. Correction or Balancing of such sum value is carried out by the compensator block, as illustrated in Fig. 1. Adder and compensator blocks are the ones which consume maximum delay along the critical path of the architecture. Thereby, we tend to reduce this delay using the concept of pipelining which will be discussed in the later sections of this paper.

2) Compensator Block: Fig. 1(c) shows the digital architecture for compensator block used in the proposed ISA adder. This block compares the output carry from each 4-bit adder block with the corresponding speculated carry using a XOR gate. Thereafter, the output from XOR gate generates an error flag (f_e) that triggers the activation of one of the two compensation techniques: error correction and reduction. If the XOR-gate output is '0' then the local sum is directly passed to the final output. Similarly, if the XOR gate gives '1' then this indicates that an error has occurred which can be either positive or negative. A positive error indicates a speculation of '0' instead of '1' and, hence, induces too low sum. Albeit a negative error indicates a speculation of '1' instead of '0' which induces too high sum. The compensation block performs an unsigned increment or decrement to the group of LSBs in the direction of this potential error (too high error is solved by a '-1' and too low by a '+1'). This correction is possible only if it does not result overflow. In case of overflow, the compensation block balances a group of MSBs of the preceding sub-adder in the opposite direction of the error. Balancing is performed based on the following observation:

$$2^n > \{2^n + 2^{n-1} + 2^{n-2} + ... + 2^0\}. \tag{3}$$

When 2^n error is detected in the sum, it could be compensated by intentionally causing the LSB errors for the sum in the opposite direction(for example, S_{n-1} = 1 $\longrightarrow$ S_{n-1} = 0). In the best case where all the LSBs can be balanced in the opposite direction, the total error is reduced to 1, as follows:

$$\{2^n - 2^{n-1} - 2^{n-2} - ..... - 2^0\} = 1. \tag{4}$$

In general, if the number of bits used for correction is p then the first computed 'p' LSBs from the 4-bit adder block are passed on to the compensation block where it is checked whether any overflow occurs or not. It does then the right compensation technique which is balancing will be selected. All this is carried out prior the 4-bit adder block finishes computing the sum of all other bits. Preferably the value of p is 1 for the optimum results. Thus, a significant feature of this adder is that neither the pre-computing of error correction nor the compensation choice lies in the critical path of the ISA adder.

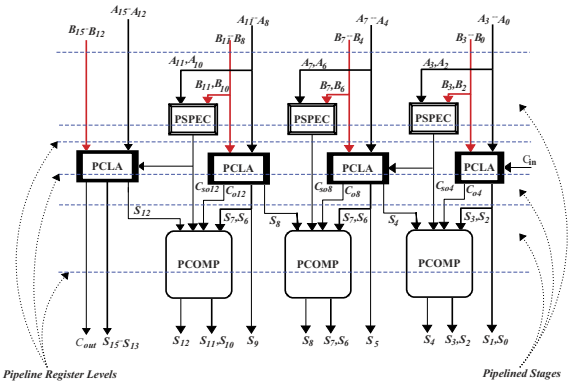

Fig. 2: Deep-pipelined VLSI architecture of the proposed ISA for n = 16 bits and x = 4 bits, with five pipeline stages, for high speed applications.

The components of compensation block which are involved in the overall critical path of ISA are the XOR gate, de-multiplexer and multiplexer.

A. Fine-Grain Pipelined Architecture

In the conventional ISA architecture, let us assume that the combinational delay of 4-bit adder, speculator and compensator blocks to be $\partial_{4b-adder}$, ∂_{spec} and ∂_{comp} respectively. In this architecture, carry-in is speculated for each 4-bit adder block and based on this; adder block calculates the local sum. Thereafter, the faulty speculation is detected by comparing speculated carry-in and prior carry-out from 4-bit adder. Subsequently, compensator block performs the correction and balancing operation. Thus, the critical path of the conventional ISA architecture includes delays of speculator of the i^{th} instant and the 4-bit adder plus compensator delays of $(i+1)^{th}$ instant, as shown in Fig. 1 (with colored lines and blocks) and expressed as follows:

$$\partial_{critical} = (\partial_{4b-adder})_i + (\partial_{spec})_{i+1} + (\partial_{comp})_{i+1}. \tag{5}$$

Thereby, based on the internal architectures of speculator and compensator blocks, the detailed version of critical path delay is given as:

$$\begin{aligned} \partial_{critical} = (\partial_{4b-adder})_i + (2{\times}\partial_{xor} + \partial_{and})_{i+1} \\ +(\partial_{xor} + \partial_{demux} + \partial_{mux})_{i+1}. \end{aligned} \tag{6}$$

where ∂_{xor}, ∂_{and}, ∂_{demux} and ∂_{mux} are the combinational delays of the logical XOR, logical AND, de-multiplexer and multiplexer respectively. It is to be noted that the speculator, compensator, 4-bit adder and the overall ISA design is feed-forward VLSI architecture. Thereby, carefully analyzing these circuits and on fine-grain pipelining this ISA architecture, we can shorten the critical path delay and hence realize fast VLSI-architecture for this adder.

For the ease of understanding, pipelining process of this work has been explained using n = 16 bit ISA architecture. Even though the value of n increases, critical path delay is unaffected because the value of x is always 4 bit (as discussed earlier) and the adder, speculator as well as compensator architectures remain unchanged. The proposed 16-bit ISA VLSI-architecture is shown in Fig. 2 where the conventional blocks has been replaced by the pipelined speculator (PSPEC), pipelined compensator (PCOMP) and pipelined 4-bit CLA (PCLA) units. Sub blocks PSPEC, PCLA and PCOMP contain two pipelined stages. Overall architecture of the suggested ISA adder has been designed with five pipelined stages and there are six levels

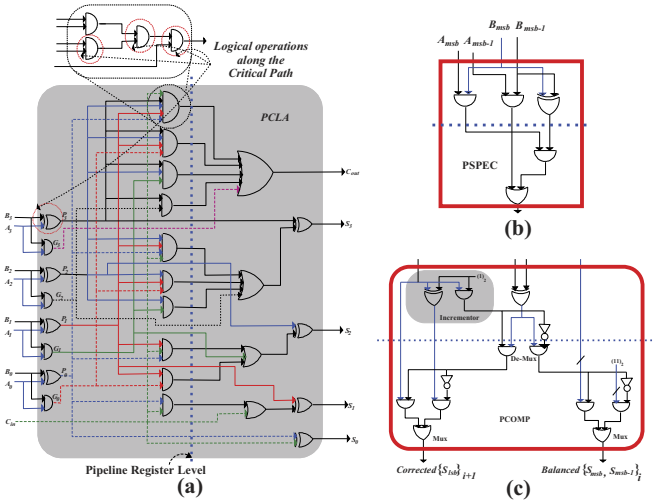

Fig. 3: Gate-level circuit of (a) Four-bit pipelined-carry look-ahead adder (PCLA) (b) Pipelined speculator (PSPEC) (c) Pipelined compensator (PCOMP) used in the proposed ISA VLSI architecture.

of registers included in this design, as shown in Fig. 2. This is a scalable architecture because the number of pipeline stages remains constant on increasing the bit widths of the operands, retaining the same critical path delay. The deep pipelined architectures of sub block have been illustrated in Fig. 3. It shows the gate-level designs of PSPEC, PCOMP, PCLA and their respective pipelined stages. On observing the proposed VLSI architectures from Fig. 2 and 3, it can been seen that the critical path of suggested architecture lies in PCLA and it includes only four two-input gate delays (one XOR and three AND gate delays). Thereby, the expression of critical path delay that decides the maximum clock frequency of the proposed ISA is given as

$$\partial_{crt-prop} = \partial_{clk-Q(ff)} + \partial_{xor} + 3\times\partial_{and} + \partial_{setup(ff)} \quad (7)$$

where $\partial_{clk-Q(ff)}$ and $\partial_{setup(ff)}$ represent clock-to-Q delay and setup time of launch and capture flip flops, respectively, in the design. Eventually, the maximum clock frequency that can be achieved by suggested ISA architecture in real world scenario is

$$F_{max} \leq 1/\{\partial_{crt-prop} - \partial_{skew}\} \quad (8)$$

where ∂_{skew} represents the clock skew which may occur in such scenario.

III. Experimental Result and Comparison

This section presents the functional verification and the board level implementation of the proposed architecture of ISA. Subsequently, the ASIC post-layout simulation results of this design have been presented and this section also includes the comparison of our results with the reported state-of-the-art work.

A. FPGA Implementation

In this work, the proposed ISA adder-architecture has been coded in hardware descriptive language (HDL) and then simulated as well as synthesized in ISE 14.3 design suite. We have synthesized this architecture for three different configurations: n = 8-bit, n = 16-bit and n = 32-bit. After the successful syntax check and synthesis of the design, the generated net-lists are placed and routed (P&R) on Spartan-3E version of Xilinx FPGA board. Therefore, post P&R simulated waveform of the proposed 32-bit ISA is shown in Fig. 2 (other waveforms are excluded due lack of space in the paper). On the other side, the resource utilization and the timing information for all the configurations of suggested adder is listed in Table I. Additionally; it includes the resource and timing information of the reported non-pipelined (NPL) adder architectures. It can be observed that the proposed adder can operate at a maximum clock frequency of 324 MHz. This value is precisely 42.15%, 48.48% and 52% better than the clock frequencies achieved by 8-bit, 16-bit and 32-bit NPL adders, respectively, reported in [9]. The comparison of exact area occupied and power consumed by these adders are possible by synthesizing as well as laying out ASIC for each of such adders. Such analysis on area and power is also carried out in the next subsection. However, Table I clearly shows some degradation in the FPGA resources utilization of the proposed ISAs. Nevertheless, the precise degradation will be demonstrated in subsequent subsection.

Table II: Comparison of area consumed by the proposed 32-bit pipelined adder (PLA) with respect to 32-bit non-pipelined adder (NPLA).

ISA Architectures	32-bit PLA	32-bit NPLA
Technology (nm)	90	90
Supply Voltage (V)	0.9	0.9
Sequential Std. Cells	224 (3.25 mm^2)	98 (1.33 mm^2)
Inverter Std. Cells	95 (0.72 mm^2)	33 (0.57 mm^2)
Buffer Std. Cells	1 (3.12 μm^2)	–
Logic Std. Cells	190 (1.15 mm^2)	88 (1.12 mm^2)
Total Number of Std. Cells	510	219
Total Area (mm^2)	5.11	3.03

B. Power and Area Analysis

In the digital circuit design, pipelining is the process of shortening the critical path at the cost of area which is predominated by the registers used to create pipeline stages in the design. Therefore, the suggested ISA architecture that is deep pipelined definitely requires extra registers in comparison with the conventional ones. On the other side, we have segregated the suggested ISA architecture into different stages by pipelining it. Now, this makes our architecture suitable for clock gating. In the proposed design, we have gated the clock signal that is fed into every stage. On doing this, the ideal stages of our architecture can be deferred from the clock switching which significantly reduces the power consumption. Such gating is valid only during the beginning and ending sessions of the addition process. On the starting of addition, later pipeline stages (towards the output side) of the design are ideal and these stages can be clock gated. Unlike towards the end of addition process, earlier stages (near the input side) of the design seem to be ideal and are clock gated. For example: pipeline stages five, four, three and two are ideal while the process is being carried out in the first stage when the addition begins. Similarly, first stage will be ideal while rest keeps processing data when addition is towards the completion. However, while the adder is in-between the process of adding continuous stream of data then there is no point of gating the clock because all the stages are busy performing the operations.

In order to quantify the area occupied and power consumed by the proposed ISA, this work includes the ASIC synthesis and post-layout simulation results of 32-bit ISA architectures. Additionally, this process is carried out for non-pipelined 32-bit ISA as well for

Table I: Comparison of post P&R results obtained from FPGA implementations of 8, 16, 32-bit pipelined and non-pipelined ISA designs.

ISA Configurations	NPL✠ (*n*=8b)	PL‡ (*n*=8b)	NPL✠ (*n*=16b)	PL‡ (*n*=16b)	NPL✠ (*n*=32b) [9]	PL‡ (*n*=32b)
FPGA Family	Spartan3E	Spartan3E	Spartan3E	Spartan3E	Spartan3E	Spartan3E
FPGA Device	xc3s500e	xc3s500e	xc3s500e	xc3s500e	xc3s500e	xc3s500e
Slices	21	33	41	66	85	134
4-Input LUTs	27	51	52	105	115	213
IOBs	27	27	51	51	99	99
Crit. Path Delay (nS)	5.396	3.081	5.980	3.081	6.419	3.081
Max. Clk. Freq. (MHz)	187.76	324.57	167.22	324.57	155.79	324.57

‡ : Proposed fine-grain pipelined ISA VLSI-architecture.
✠ : Reported non-pipelined ISA VLSI-architecture in [9].

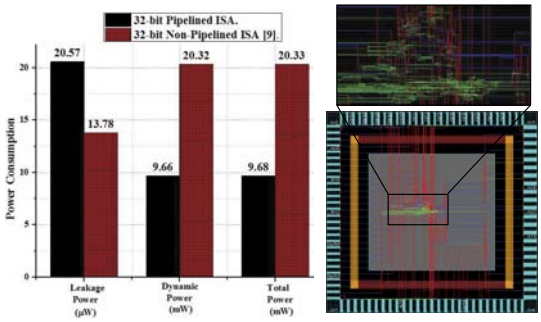

Fig. 4: Comparison plot of leakage, dynamic and total power consumed by 32-bit pipelined and non-pipelined ISA, when implemented in 90 nm-CMOS process and operated at 400 MHz clock frequency and chip layout of proposed 32-bit pipelined ISA with core area of 5111 μm^2 when implemented in UMC 90 nm-CMOS technology node.

the purpose of comparison. To begin with, HDL coded architectures of 32-bit ISAs are synthesized with realistic design constraints using the standard cell library of UMC 90 nm-CMOS technology node. This process continues iteratively until a timing violation free gate-level netlist is generated. Thereafter, it is instantiated along with the input-output (IO) cells using the LEF (five metal layer) file of 90 nm CMOS process in the SoC Encounter platform of Cadence tool. These standard cells are place, power cum signal routed, clock tree synthesized and time analyzed, couple of times during the backend (physical) design process. Then, the chip-layout is design rule check (DRC), layout versus schematic (LVS), density checked and finally the switch activity file (SAF) and the netlist is generated. Eventually, the final netlist along with the test bench, SAF, standard-parasitic exchange-format (SPEF) file are post-layout simulated to check the valid output of the layout and then calculate the total power consumed. The proposed architectures could operate up to maximum clock frequency of 444.64 MHz with a positive slack of 251 pS. At the supply voltage of 0.9 V and a clock frequency of 400 MHz, the power consumed (including static plus dynamic powers) by suggested 32-bit pipelined and non-pipelined [9] ISAs are illustrated in Fig. 4. The pipelined ISA consumes total power of 9.68 mW at 400 MHz and it can be observed that this pipelined ISA consumes lesser power by 52.38% with respect to its peer 32-bit non-pipelined ISA, as a result of the clock gating technique incorporated in the design. On the other side, Table II shows the total area consumed as well as the standard cell counts (with their respective area consumption) of the 32-bit suggested ISA and reported non-pipelined ISA. It can be observed that there is a surge in the number of sequential and inverter cells of the proposed adder due to additional register levels used in the design. Thereby, the pipelined adder presented in this work has a area degradation of 40.7% in comparison with the non-pipelined adder. Eventually, the chip layout of proposed 32-bit pipelined ISA has been shown in Fig. 4.

IV. Conclusion

In this paper, we presented high-speed and low-power version of the contemporary ISA design. This architecture has been fine grain pipelined and clock gated to escalate speed and alleviate power consumption respectively. Experimental results showed that the suggested ISA could operate at 324.57 MHz and 444.64 MHz of maximum clock frequency in FPGA and 90 nm-CMOS ASIC platforms respectively. Subsequently at this technology node, it occupied 5111 μm^2 of area and consumed 9.68 mW of total power at 400 MHz. Therefore, the proposed ISA can operate at 52% higher speed, needs 52.38% lower power and occupies 40.7% more area than the state-of-the-art ISA design. Thereby, such design would definitely play significant role in the design of contemporary as well as future electronic devices for IoE and many other contemporary applications. However, the area issue can be resolved to some extent by using lower technology nodes in the design process.

References

[1] Behzad Razavi, *"Cognitive Radio Design Challenges and Techniques," IEEE Journals of Solid-State Circuits (JSSC), vol. 45, no. 8, pp. 1542-1553, 2010.*

[2] Gyanendra Prasad Joshi, Seung Yeob Nam and Sung Won Kim, *"Cognitive Radio Wireless Sensor Networks: Applications, Challenges and Research Trends," Sensors, vol. 13, no. 9, pp. 11196-11228, 2013.*

[3] D. Blaauw et al., *"IoT Design Space Challenges: Circuits and Systems," IEEE Symposium on VLSI Technology (VLSI-Technology): Digest of Technical Papers, pp. 1-2, 2014.*

[4] T. Liu and S. L. Lu, *"Performance Improvement with Circuit-level Speculation," 33rd Annual IEEE ACM International Symposium on Microarchitecture (MICRO-33), pp. 348-355, 2000.*

[5] N. Zhu, W.-L. Goh, and K.-S. Yeo, *"An Enhanced Low-power High-speed Adder For Error-tolerant Application," 12th International Symposium on Integrated Circuits (ISIC), pp. 69-72, 2009.*

[6] M. Weber, M. Putic, H. Zhang, J. Lach, and J. Huang, *"Balancing Adder for Error Tolerant Applications," IEEE International Symposium on Circuits and Systems (ISCAS), pp. 3038-3041, 2013.*

[7] N. Zhu, W.-L. Goh, G. Wang, and K.-S. Yeo, *"Enhanced Low-power Highspeed Adder for Error-tolerant Application," IEEE International SoC Design Conference (ISOCC), pp. 323-327, 2010.*

[8] Y. Kim, Y. Zhang, and P. Li, *"An Energy Efficient Approximate Adder with Carry Skip for Error Resilient Neuromorphic VLSI Systems," IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 130-137, 2013.*

[9] Vincent Camus, Jeremy Schlachter and Christian Enz, *"Energy-Efficient Inexact Speculative Adder with High Performance and Accuracy Control," IEEE International Symposium on Circuits and Systems (ISCAS), pp. 45-48, 2015.*