SERBIAN JOURNAL OF ELECTRICAL ENGINEERING
Vol. 22, No. 1, February 2025, 1-16
UDC: 004.415.2:621.382 **DOI: https://doi.org/10.2298/SJEE2501001S**
Original scientific paper

Implementation of Serial Peripheral Interface Slave Device Based on Uncommitted Logic Arrays

Alexander Sinyukin[1], Mark Denisenko[1], Alina Isaeva[1], Ivan Kots[1], Andrey Kovalev[2]

Abstract: Microcontrollers and microprocessors link to peripheral devices (sensors, converters, transceivers, memory modules) via communication interfaces. One of the most widespread interfaces is the Serial Peripheral Interface (SPI), characterized by simplicity, energy efficiency, and high-speed performance. The purpose of the study was to design an SPI bus Slave device based on uncommitted logic arrays technology. Uncommitted logic arrays are integrated circuit technology that is intermediate between full-custom integrated circuits and programmable logic devices in terms of power consumption, dimensions, development, and manufacturing cost. The SPI Slave device was designed using the computer-aided design system Kovcheg dedicated to the chosen technology. Layout synthesis using Kovcheg can be executed based on a circuit developed with a built-in graphics editor or based on structural descriptions in the Verilog hardware description language. A technique of digital device design using Kovcheg based on behavioral descriptions is proposed to optimize the design process. The SPI Slave device was manufactured, and experimental results of the fabricated microchip agree well with simulation results. The device is able to work properly at clock frequency up to 5 MHz. Tests of speed performance, interference resistance, and ability to operate at different voltage levels were carried out. Attained results imply that the proposed device can function as an SPI Slave unit to communicate control devices with embedded peripherals.

Keywords: Serial Peripheral Interface, Uncommitted logic arrays, Digital design, Integrated circuit, Computer-aided design, Verilog, Measurements.

[1]Scientific Research Laboratory of Functional Nanomaterial Technology, Southern Federal University, Taganrog, Russia, sinyukin@sfedu.ru, https://orcid.org/0000-0003-0496-0087;
dema@sfedu.ru, https://orcid.org/0000-0001-5044-7482;
isaevaas@sfedu.ru, https://orcid.org/0000-0002-5145-0963;
inkots@sfedu.ru, https://orcid.org/0000-0002-1257-7304
[2]Engineering Center of Radio and Microelectronics Devices Design, Southern Federal University, Taganrog, Russia, avkovalev@sfedu.ru, https://orcid.org/0000-0003-0545-7416

Colour versions of the one or more of the figures in this paper are available online at https://sjee.ftn.kg.ac.rs

1 Introduction

Serial Peripheral Interface (SPI) [1] is a commonly used means for wired synchronous serial data communication within short distances. At the same time, SPI is not strictly standardized [2], which, along with its popularity, leads to a wide variety of protocols and implementations. The interface ensures data transmission between integrated circuits and embedded systems. This interface is able to control sensors of different types, data converters, memory modules, displays, integrated transceivers, and other devices.

Application-specific integrated circuits (ASICs) can be divided into three categories: full-custom integrated circuits (ICs), field-programmable gate arrays (FPGAs), and semi-custom-designed microcircuits based on uncommitted logic arrays (ULAs). FPGAs have a set of advantages: short development cycle, reproducibility, and versatility, but their energy consumption and dimensions are higher compared to other microcircuit types. Full-custom integrated circuits are flexible, reliable, have high performance, have a small footprint, and have low cost in the case of large-scale production; however, they require the highest expenses for development (therefore, the fabrication of small batches is not economically reasonable) and require precise fabrication equipment and adjusted technological processes. ULA microchips are in an intermediate position: time expended on development and manufacture is comparable with FPGA, and they are close to custom ICs in terms of energy consumption, reliability, and resistance to harmful effects [3].

In the case of full-custom ICs, fabricating a particular design requires wafers to go through all processing steps under the control of a full set of lithographic photomasks, all of which are made to order for this very design, and mask manufacturing is a dominant contribution to nonrecurring fabrication costs. But in semi-custom ICs such as ULA, only a small subset of fabrication layers is unique to each design. Customization starts from preprocessed wafers that include large quantities of prefabricated but largely uncommitted primitive items such as transistors or logic gates. These wafers then undergo a few more processing steps during which those primitives get interconnected such as to complete the electrical and logic circuitry required for a particular design. Due to the small number of design-specific photomasks and processing steps, semi-custom manufacturing significantly reduces the nonrecurring costs as well as design and fabrication time [4]. In summary, for large-scale manufacture, the most cost-effective route remains an uncommitted logic array or a fully custom-designed IC, whereas the complexity of FPGAs often turns out redundant [5].

A comparison of key performance metrics of SPI designed using ULA technology and full-custom IC processes is presented in **Table 1**. It can be seen from the table that with the development of technology, feature size and supply voltage of the devices decrease, so footprint and power consumption decrease,

respectively. Since the feature size of full-custom IC processes roughly corresponds to the used ULA technology (e.g., 0.5 μm and 1.6 μm), consumed power for these devices is approximately of the same magnitude, but it is obvious that with technological advancement, power consumption will be reduced.

Table 1
Comparison of ULA and full-custom ICs.

	Proposed device	[6]	[7]	[8]
Power, mW	155.8	60	19	1.9
Area, mm^2	17.1	0.48	0.45	0.27
Frequency, MHz	5	1	1	100
Supply, V	5	5	1.8	1.1
Technology	ULA 1.6 μm	BCD 0.5 μm	CMOS 0.18 μm	CMOS 40 nm

Power is one of the most challenging metrics to compare [9], and it is more complicated to directly compare the power consumption between devices implemented using ULA and FPGA in terms of the relative low prevalence of the former in the literature. But power consumption seems to be one of the most important issues separating FPGA and ASIC designs [9] and one of the most important limiting factors for FPGA [10]. FPGA power consumption is from 10 to 50 times greater than ASICs, and the ratio increases with the lowering of circuit complexity [9, 10]. The same applies to footprint: according to [9], the ratio of silicon area required to implement circuits in FPGAs and ASICs is on average from 20 to 40 depending on the circuit complexity. Addressing the modern FPGAs, using which some SPI devices have been implemented ([2, 11, 12]), the sizes of the smallest FPGAs of Altera Cyclone IV, Cyclone V, and Xilinx ZedBoard Zynq-7000 device families are not less than 120 mm^2, which is almost by an order of magnitude more than the proposed ULA design. Speed performance of SPI based on ULA is comparable with devices implemented using full-custom ICs (except advanced nanosized processes), as it can be seen from Table 1, whereas clock frequency of SPIs realized using FPGAs can vary from 0.5 MHz [12] to 100 MHz [2]. But the minimum internal frequency of FPGAs is usually several times higher, thus the fractional frequency module is required to step down the speed of the clock [12, 13], which raises power consumption.

In research [14], an SPI bus is developed using finite-state machines in the Verilog language, and an FPGA is chosen as the hardware platform for implementation. An IP core realizing the SPI protocol to provide intra-chip interconnections in systems-on-the-chip is proposed in [15]. In paper [6], a current-mode operating SPI as a part of an integrated circuit for battery voltage

monitoring of electric vehicles is presented. A configuration of a daisy-chain bus between a microcontroller and cascadable microcircuits for voltage monitoring is formed using the presented interface, which allows one to considerably reduce the number of input-output ports required for the microcontroller. Advanced SPI allowing the conversion of serially transmitted data to data transmitted in a parallel way and which is able to interact with peripheral devices working at different supply voltage levels is designed in [16]. Interface is characterized by the possibility to operate at high frequencies (up to 55 MHz) and in a wide temperature range. In [17], the SPI module is presented as a component of a flash analog-to-digital converter (ADC) developed using the 180 nm CMOS process to provide interaction of the ADC with microcontrollers. Reconfigurable SPI to ensure interaction between the digital baseband processor and wideband radiofrequency transceiver is developed in [11]. IP-core software of this SPI is realized in VHDL language, and hardware is implemented via FPGA for compliance of synchronization conditions and to control the transceiver in a real-time mode. SPI implementation on FPGA is proposed in work [13] too, in which three interfaces, SPI, I2C, and RS-232, are analyzed and realized in Verilog language, and as combined they form a multifunctional series communication interface adapting depending on application conditions. Another design of a multiprotocol conversion module by which communication and data conversion across SPI, I2C, and UART interfaces can be carried out is described in work [18]. In work [19], an SPI-to-I2C protocol communication approach is presented. It is described in Verilog language and is relevant for applications in which a control device needs to interact with many peripheral devices; thus, the possibility to quickly transmit commands and data from the controller to selected peripheral devices (advantage of SPI) and, meanwhile, there is a need to reduce the number of specialized pins of peripheral devices (advantage of the two-wire I2C interface) is required. In paper [12], improved UART and SPI protocols are realized on FPGA instead of a single-chip microcontroller with a modified RISC core for improving speed performance and providing scalability in Internet of Things nodes. Tests showed that speed performance of the SPI Slave almost doubled due to implementation on FPGA. As well as an SPI Slave device is implemented in work [7], where it is incorporated into a digital control core designed using a 40 nm CMOS process. In research [20], a distributed genetic algorithm is proposed for usage in inexpensive low-power embedded devices like 8-bit microcontrollers and SPI buses. SPI is preferred to I2C and UART since SPI is simpler to implement, faster, and consumes less power due to the absence of load resistors, which are common for I2C. A controller of the SPI Slave bus implemented on FPGA for performance and efficiency tests is developed in [2]. In this research, the usage of combinational pipeline data processing based on separate preliminary and subsequent addressing allows one to mitigate synchronization requirements substantially. In paper [21], during an SPI unit

design using the 130 nm CMOS process, a Relative Timing method has been used, enabling simple integration into computer-aided design (CAD) flow and allowing the reduction of the footprint and consumed power of developing digital systems in cases when peripheral devices stay inactive for a long time. Verification of the three-wired SPI protocol presented in [8] is performed through both FPGA and custom IC: the Slave is implemented on the custom microcircuit while the Master is reconstructed via FPGA.

It is seen from literature analysis that modern SPI research and development results are implemented either using FPGA or in the form of full-custom microcircuits. In this work, an SPI Slave device based on uncommitted logic arrays is proposed. This allows achieving small size, reliability, and low energy consumption on the one hand and reducing design and fabrication costs on the other hand.

2 Device Development

Serial Peripheral Interface functions in full-duplex mode between one Master device and one or several Slave device(s). The SPI bus can be configured in three-wired or four-wired variants. A typical four-wired SPI bus, as shown in Fig. 1, is defined by four logic signals, which designations can be distinct in different sources [22]: SCLK is serial clock, generated by Master; DIN (data input) is data transmitted from Master to Slave; SDO (serial data output) is data transmitted from Slave to Master; CNV (conversion) is a signal from the Master device to start data conversion with a particular Slave device.

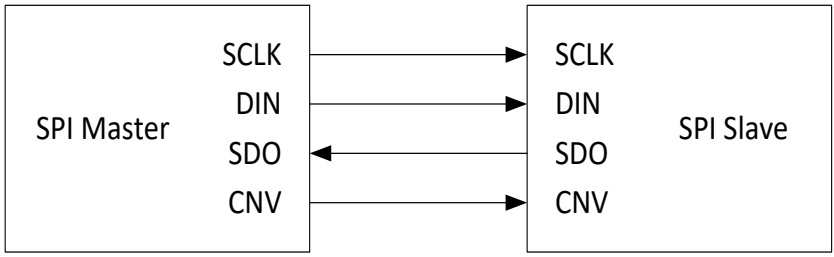


Fig. 1 – *SPI bus structure with one Master device (SPI Master) and one Slave device (SPI Slave).*

For data transmission between Master and Slave, shift registers are used. Each SPI register usually consists of eight bits (although there are cases of more bits being used), including bits for control, state monitoring, and other commands that regulate all device functions. Data are transmitted while the Master generates a clock signal, and the CNV signal is active (most often, the active level of the signal is a low level). A flowchart of the developed device operation is presented in Fig. 2.

Development of SPI Slave in this work was performed using a CAD Kovcheg V3.042 dedicated to designing large-scale integration circuits of the ULA type [3]. Kovcheg CAD allows one to perform a full design cycle from logic project to preparing the integrated circuit for production. The first step after choosing a particular ULA process and corresponding package is circuit diagram design. Descriptions of electronic circuits are often written in hardware description languages such as Verilog or VHDL using behavioral style due to their convenience and versatility.

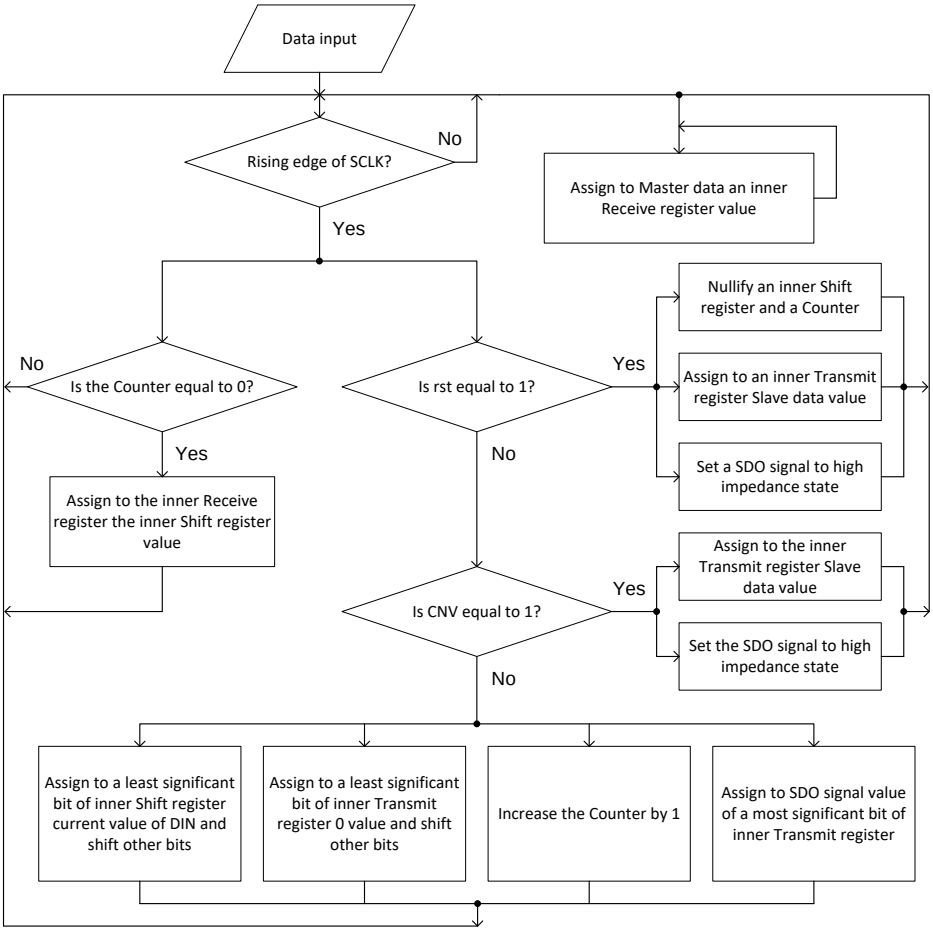


Fig. 2 – *Flowchart of designed SPI Slave operation.*

In the proposed work, SPI module operation was initially described in Verilog using behavioral description, which was structured in compliance with the flowchart in Fig. 2. In the first part of the description, inputs (1-bit rst, 1-bit

SCLK, 1-bit CNV, 1-bit DIN, and 8-bit Slave data), outputs (1-bit SDO and 8-bit Master data), and inner registers (8-bit Transmit register, 8-bit Receive register, 8-bit Shift register, and 3-bit Counter) are designated. Then continuous assignment is applied to constantly renewed Master data output. After that, three clock-controlled procedural blocks describing sequential logic are presented. The first one shifts the inner Shift and Transmit registers (the former one is shifted by DIN input) and increases a Counter value by one while CNV is enabled (low level). In the case of an active reset, nullification of the Counter and the inner Shift register as well as assigning the Slave data value to the Transmit register are executed in the first block. The second block is used to define the value of the series output SDO based on the value of the Transmit register's most significant bit while CNV is active and reset is inactive. And the third block interacts with inner registers to provide value (equal to the value of the Shift and Receive registers) for the parallel outputs of Master data when the Counter counts to eight (2^3) and takes zero value on the next clock pulse.

There are two ways to create a circuit in Kovcheg using the intrinsic library of electrical components and logic cells: graphically or in the form of a structural description. However, there is no feature set in Kovcheg for circuit synthesis based on behavioral descriptions. For this reason, a technique for digital integrated microcircuits design in the Kovcheg system using behavioral descriptions in hardware description languages has been developed. According to this technique, a composed behavioral description should be firstly uploaded to the design system, which allows for the conversion of behavioral descriptions to structural ones and for simulating them. Examples of such design systems are the Vivado Design Suite [23] application software package, which is applied to FPGA design, or the Genus Synthesis digital implementation module from Cadence [24], which was used in this work. Alternative free and open-source logic synthesis tools such as Yosys [25] could be used as well. It should be noted that neither Genus nor Vivado are suited to design the integrated circuits of ULA type, while Kovcheg is dedicated CAD, and though its scope of applicability is narrow, Kovcheg is almost a unique software tool for these tasks.

Structural descriptions synthesized using Genus included five types of logic cells: inverters, two-input NAND and NOR cells, D-latches with an active rising edge of the clock, and tristate buffers. In the Kovcheg system, accurate functional analogues from the intrinsic library were manually fitted for these cells. Corresponding cells in structural description were replaced by these analogues; e.g., inverter cells from structural description synthesized in Genus were replaced by inverter cells from the Kovcheg library, preserving the relative position of devices and their nodes in the circuit. In other words, the names of all cells and their nodes were replaced with corresponding ones from Kovcheg, but the whole circuit structure stayed the same. Besides that, digital input and output cells with controlling drivers were added to the synthesized circuit description.

There was a challenge during synthesis, or, more precisely, translation of structural descriptions from Genus to Kovcheg, related to signal propagation. Every logic cell in the Kovcheg library has specific fan-out capability; e.g., two-input NAND and NOR cells have this capability equal to 3, and the fan-out capability of a simple D-latch is 5 for true output. It means it could not be more than the corresponding value (3, 5, etc.) of other cells connected to the output of this cell in the circuit. But fan-out capability for cells in the Genus library exceeds the capability of corresponding cells in Kovcheg, and cases occurred when cells in Kovcheg had more cells at output than it was acceptable. That problem was solved by adding in parallel the same cells with common inputs to relieve outputs.

Results of behavioral description simulation using Genus Synthesis have been compared with simulation results of structural description synthesized using Genus and then with simulation results of structural description formed in Kovcheg software (Fig. 3). All simulation results functionally correspond to each other (it was considered that different process design kits were used in simulations).

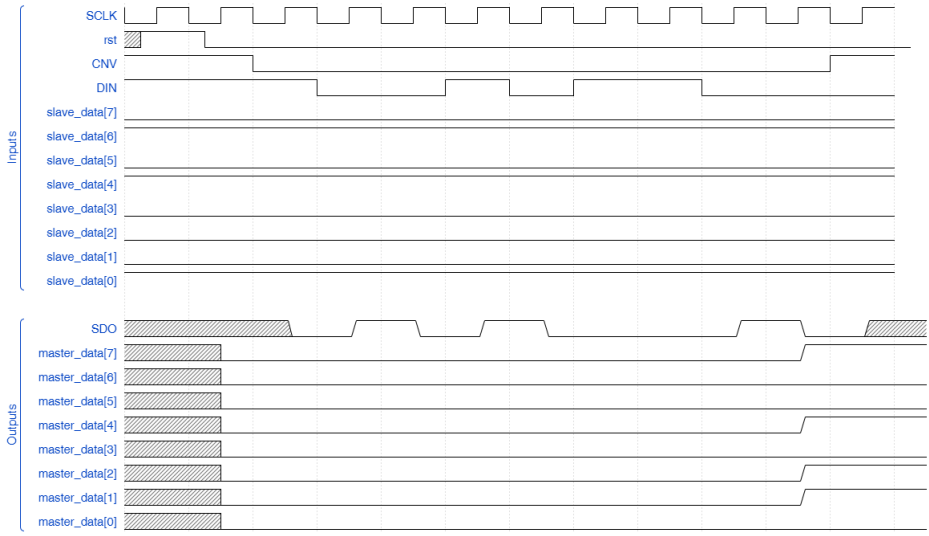


Fig. 3 – *Waveforms of SPI Slave unit operation simulated using Kovcheg software* (*and redrawn using WaveDrom software for better visualization*).

The purpose of the SPI Slave device is to transmit commands from the Master device to the peripheral and to transmit data from the peripheral to the Master. In our case, the command from the Master is expressed as a DIN sequence of pulses, and Master data is auxiliary output of the Slave for storing this command. The data from peripherals are described as Slave data, and the Master input to transmit this data is SDO. As it can be seen from Fig. 3, the series

data 10010110_2 at the DIN input are equal to the data at the parallel output Master data, while the parallel data 01010001_2 at the Slave data inputs are equal to the data at the series output SDO; therefore, the device operates as intended.

Simulation results show that layout delays were not greater than acceptable values at clock frequency f_{sclk} = 10 MHz. According to experiments, the highest clock frequency at which the device operates correctly is 5 MHz, and simulation results attained using Kovcheg are shown in Fig. 3. After circuit synthesis and verification, further SPI module design processes were performed according to design flow. At the stage of microcircuit pin arrangement, the pins were manually distributed between four sides of the previously chosen package. The next stages, namely layout synthesis, layout optimization, layout control, and delay analysis, were executed automatically. The final step was design validation. The validation is starting from the simulation of minimum acceptable, nominal, and maximum acceptable values set of transistors transconductance, layout delays, temperature, and supply voltage. Thus, complete enumeration was performed. The details for test parameter values are listed in **Table 2**. The layout delays were determined by the resistivity of polysilicon buses.

Table 2
Validation tests in Kovcheg CAD.

Parameter	Minimum	Nominal	Maximum	Unit
Supply voltage	4.5	5	5.5	V
Temperature	–60	20	120	°C
PMOS transconductance	9.6	12.0	14.4	μA/V
NMOS transconductance	20.0	25.0	30.0	μA/V
Polysilicon resistance	432	540	648	Ω

Eventually all 243 tests were accomplished without errors. The CMOS 1.6 μm process was used, and the chip area is 17.1 mm^2 (3.8 mm × 4.5 mm). The layout and the photomicrograph of the designed chip are depicted in Fig. 4a and Fig. 4b, respectively, where interconnected transistors could be seen as light green (Fig. 4a) and light grey (Fig. 4b) areas in the middle of the picture. Forty-two pads are distributed on the edges.

3 Measurements Results

A test batch of microcircuits has been manufactured at the Scientific-Manufacturing Complex ‘Technological Centre’ under the Multi-Project Wafer (MPW) program [26]. The fabricated microcircuit implementing the SPI Slave device was placed on a printed circuit board (PCB, Fig. 4c), and tests of its functional performance were carried out.

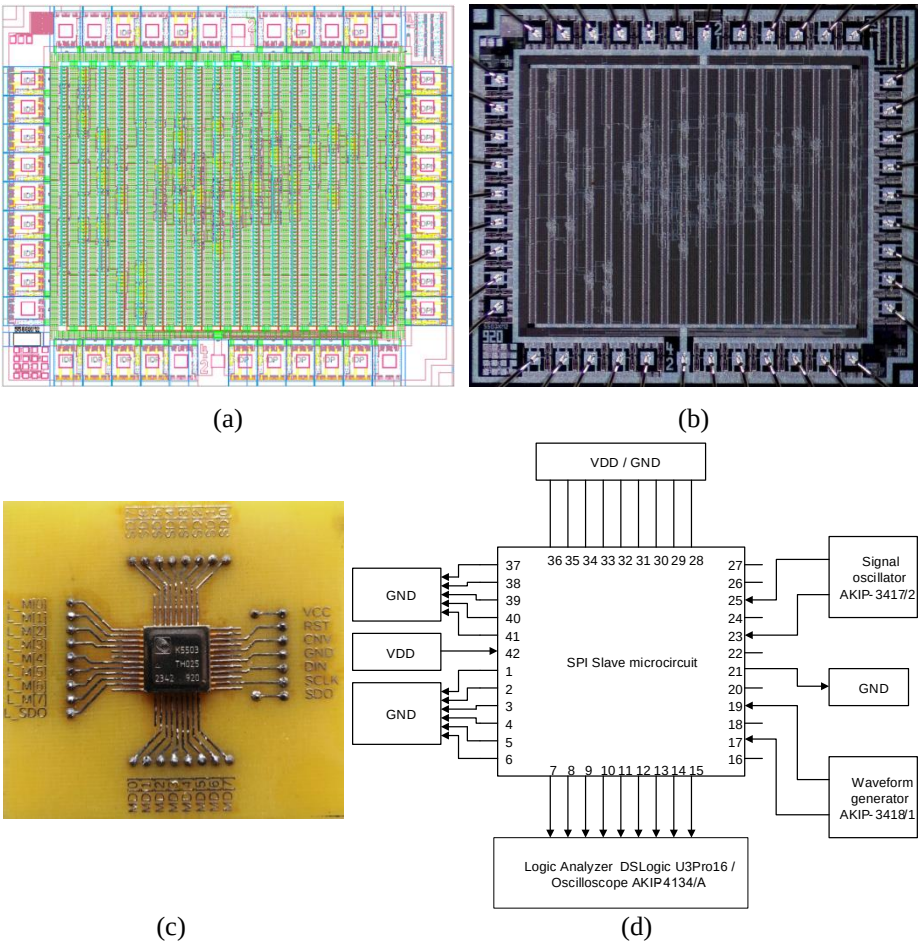


Fig. 4 – *Designed microcircuit:* (a) *Layout;* (b) *Photomicrograph;* (c) *PCB with mounted microcircuit;* (d) *Block diagram of the test setup.*

Tests included checking of data transmission consistency from Master to Slave and from Slave to Master, checking of device speed performance, checking of device capability to retain data integrity under noise influence, and supply voltage changing within acceptable limits. The measurements were performed using a high-frequency signal oscillator AKIP-3417/2, a waveform generator AKIP-3418/1, a digital oscilloscope AKIP-4134/A, and a logic analyzer DreamSourceLab DSLogic U3Pro16. The block diagram of the test setup is presented in Fig. 4d. Slave data was set through voltage supply (logical ones) and ground potential (logical zeros).

The test of data transmission consistency aimed to check the operation validity of the SPI bus Slave device. A criterion for proper implementation of the input data transmission function is accurate correspondence of an 8-bit number serially transmitted from Master through the DIN port and a number written to a special 8-bit Master data register of Slave at the end of each transmission cycle. A criterion for proper implementation of the output data transmission function is accurate correspondence of an 8-bit number written to a special 8-bit register Slave data of Slave and a number serially transmitted to the Master through the SDO port. Results of microcircuit tests with input signals equivalent to signals shown in Fig. 3 in terms of frequency, pulse width, and bit sequences are shown in Fig. 5.

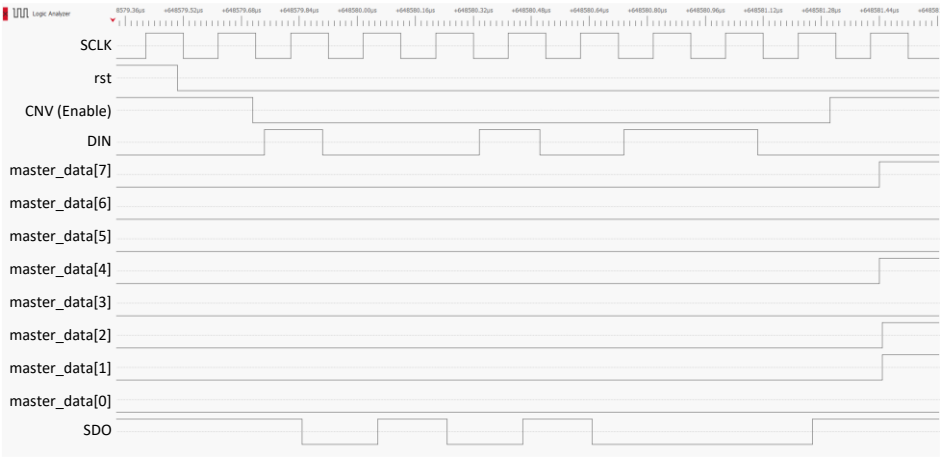


Fig. 5 – *Results of SPI Slave device tests, attained with a logic analyzer and visualized using DSView software* (*clock frequency* 5 MHz).

As seen from Fig. 5, measurement results agree exactly with simulation results (Fig. 3). After the reset command (rst) and setting logic zero on the inverse CNV input, an 8-bit number 01010001_2 (chosen for illustrative purposes and not shown in Fig. 5) is set on the parallel Slave data inputs, and an 8-bit number 10010110_2 (also chosen for illustrative purposes and set through the waveform generator and EasyWave software) is set on the DIN input. The number 01010001_2 from Slave data inputs is converted to serial form and transmitted to the SDO output (the lowest signal in Fig. 5) with accordance to the clock signal and considering delays. SDO signal appears as a sequence of high and low voltage levels. After completion of the transmission and setting the CNV signal at logic one, the number 10010110_2, which is also the sequence of pulses in the diagram, serially transmitted from the DIN input, appears on the parallel Master data outputs. Comparing diagrams in Fig. 3 (simulation) and Fig. 5 (measurements), it can be seen that output signals are the same as input signals

(in relation to pulse sequences and frequency), i.e., with DIN = 10010110_2 and Slave data = 01010001_2 at the inputs, there are 10010110_2 and 01010001_2 at the Master data and SDO outputs, respectively.

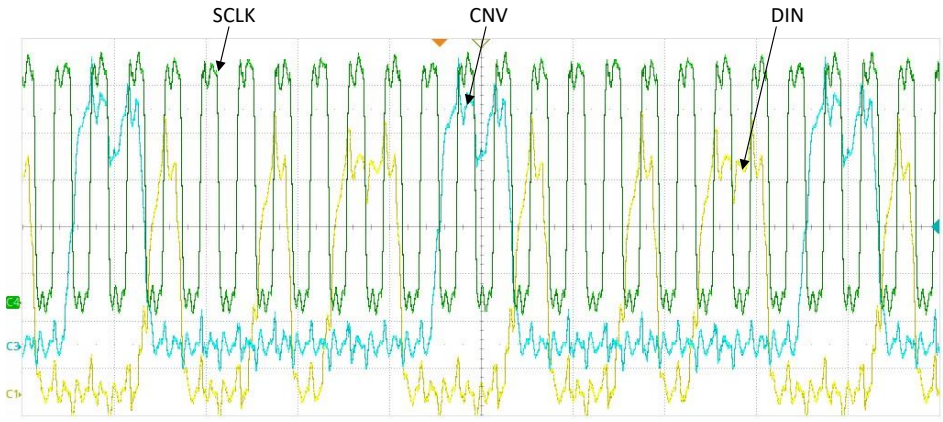
Results of the microcircuit measurements at clock frequencies of 5 MHz and 750 kHz using the oscilloscope are shown in Fig. 6a and Fig. 6b, respectively. In the case of a maximum-reached frequency equal to 5 MHz, signals are deformed. That is why waveforms for f_{sclk} = 750 kHz are additionally presented, where signals form are well-shaped. For better visualization, waveforms in Fig. 6 are biased vertically, and the amplitude of each of them is approximately 5 V. As seen from Fig. 6, attained results coincide with the results of the simulation and results of the device test using a logic analyzer, and these results verify the proper operation of the proposed device.

The speed performance tests of the developed SPI Slave device aimed to identify a maximum clock frequency value at which the device operates correctly. Measurements were carried out in the same way as in the previous case, but at the same time, frequencies of clock (SCLK) and DIN signals were varied proportionally. The clock frequency range was from 0.5 to 10 MHz. It is found that the device operates correctly at clock frequency up to 5 MHz, but at clock frequency equal to 7.5 MHz, bit sequences arriving to outputs do not correspond to combinations set on inputs (presumably signal latency occurs). The fails at the frequency higher than 5 MHz could be invoked by imperfections of wire connections between the chip and measurement equipment, crosstalk, or consequences of applying the proposed approach of translating the Verilog code since the structural description has been originally synthesized for another technological process.

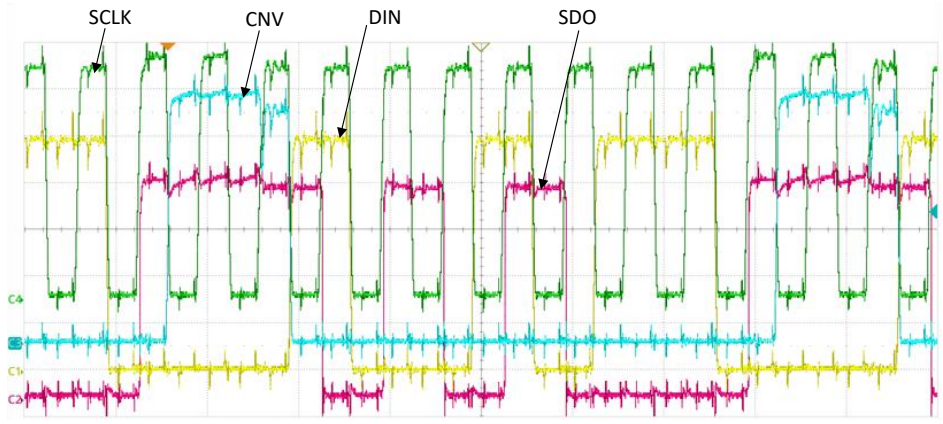
The noise resistance test aimed to check the correctness of the SPI Slave operation at signals with different interference level settings on the inputs. For noise resistance examination, additional noises were added to the DIN signal formed using EasyWave software. The noises were defined in the form of amplitude spikes (spike magnitude for different cases varied in the range of 0-4% and 0-10% of voltage supply) for each n^{th} voltage value in the defined pulse sequence (cases with n = 25 and n = 50 were investigated), considering that the defined digital signal of special form includes 16,384 voltage values, which is standard for the control software.

This DIN signal of special form with added noise described using EasyWave is shown in Fig. 7. Logical one is equal to 5 V here, and logical zero is equal to 0, but each 25^{th} value of the digital voltage signal amongst 16,384 distinct values is higher or lower than 5 V in the case of logical one and higher than 0 in the case of logical zero by value from 0 to 0.5 V (10% of voltage supply) randomly, what is supposed to be noise imitation. The clock frequency in the case shown in Fig. 7 was f_{sclk} = 1 MHz. Results showed that with such noise levels, the microcircuit

continues to correctly operate, and bit combinations on the Master data outputs correspond to combinations set on DIN.



(a)



(b)

Fig. 6 – *Results of the SPI operation tests (green - SCLK; cyan - CNV; yellow - DIN; magenta - SDO):* (a) f_{sclk} = 5 MHz*;* (b) f_{sclk} = 750 kHz.

The voltage supply alteration test aimed to check the correctness of SPI Slave operation under changing supply voltage applied to the VCC port. For that purpose, the supply voltage from a voltage source was varied in the range from 4.5 to 5.5 V at measurements (±10% of typical value VCC = 5 V). According to attained results, voltage changing within acceptable limits does not cause any data transmission errors. This conclusion results from the lack of discrepancy between input (DIN, Slave data) and output (Master data, SDO) signals, i.e., binary codes are not biased by one or some digits after transmission.

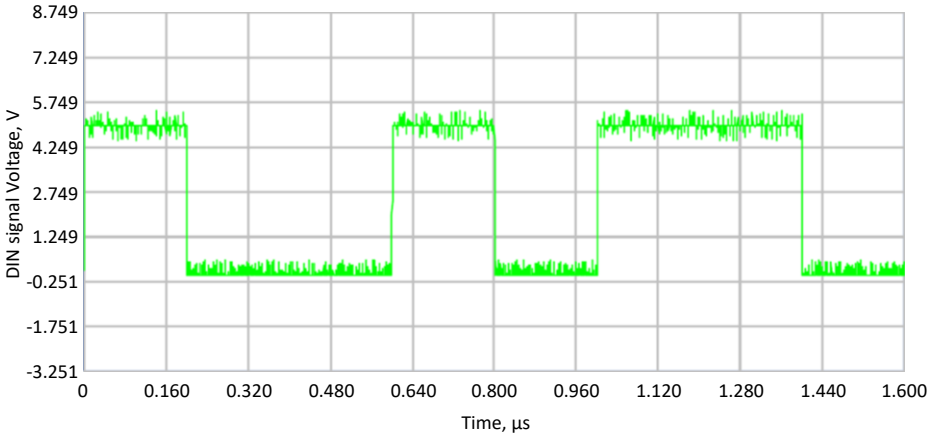


Fig. 7 – *Adding noises to special form signal (DIN) with amplitude in range 0-10% of voltage supply and n* = 25.

4 Conclusion

The module of one of the most widely accepted data transmission interfaces was implemented as an integrated circuit based on uncommitted logic arrays technology. For this purpose, the technique of behavioral descriptions integrated into the computer-aided design system Kovcheg, intended for microcircuits design based on uncommitted logic arrays, was developed and evaluated. The designed SPI Slave device is able to convert serial code from the Master device into parallel code and vice versa - parallel code from peripheral devices into sequential code. The ULA chip occupies an area of 17.1 mm2. The fabricated microcircuit was tested, and the tests showed that the limit clock frequency at which the microcircuit operates properly is 5 MHz. The microcircuit successfully passed tests for noise resistance and operates in the defined supply voltage range correctly.

5 Acknowledgments

Research was supported by the Ministry of Science and Higher Education of the Russian Federation in the framework of the state task in the field of scientific activity, grant number FENW-2022-0001.

The authors would like to express gratitude to Doctor of Sciences in Technical Sciences, Professor Konoplev B.G., for his aid during the research.

6 References

[1] SPI Block Guide V03.06, S12SPIV3/D, Motorola, Inc., 2003.

[2] V. Milovanovic, D. Tasovac: A Custom Serial Peripheral Interface (SPI) Bus Slave Controller with Read/Write Register Banks, Proceedings of the 4th International Conference on Electrical, Electronics and Computing Engineering (IcETRAN), Kladovo, Serbia, June 2017, p. ELI1.7.

[3] С. В. Гаврилов, А. Н. Денисов, В. В. Коняхин, М. М. Соколовская: Система автоматизированного проектирования «Ковчег 3.04», Техносфера, Москва, 2019.

[4] H. Kaeslin: Top-Down Digital VLSI Design: From Architectures to Gate-Level Circuits and FPGAs, Elsevier Inc., Waltham, 2015.

[5] B. Holdsworth, C. Woods: Digital Logic Design, 4th Edition, Elsevier, Oxford, 2002.

[6] X. Wang, H. Zhang, L. Zhang, J. Zhang, Y. Hao: A Daisy-Chain SPI Interface in a Battery Voltage Monitoring IC for Electric Vehicles, Proceedings of the 12th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT), Guilin, China, October 2014, pp. 1 – 3.

[7] D. Li, D. Zhao: A Low-Power Digital Control Core with High-Speed SPI Slave for Phased Array Application in 40nm CMOS, Proceedings of the IEEE International Conference on Integrated Circuits, Technologies and Applications (ICTA), Chengdu, China, November 2019, pp. 67 – 68.

[8] M.- C. Tuan, S.- L. Chen, Y.- K. Lai, C.- C. Chen, H.- Y. Lee: A 3-Wire SPI Protocol Chip Design with Application-Specific Integrated Circuit (ASIC) and FPGA Verification, Proceedings of the 3rd World Congress on Electrical Engineering and Computer Systems and Science (EECSS), Rome, Italy, June 2017, p. EEE 110.

[9] I. Kuon, J. Rose: Measuring the Gap Between FPGAs and ASICs, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 26, No. 2, February 2007, pp. 203 – 215.

[10] A. Amara, F. Amiel, T. Ea: FPGA vs. ASIC for Low Power Applications, Microelectronics Journal, Vol. 37, No. 8, August 2006, pp. 669 – 677.

[11] H. K. Boyapati, H. B. Nimmala, M. Jain: Design and Development of Flexible Reconfigurable SPI Interface Between Baseband and RF Subsystems for Wireless Radio Prototyping, Proceedings of the International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, March 2016, pp. 2468 – 2472.

[12] R. R. Pahlevi, A. G. Putrada, M. Abdurohman: Fast UART and SPI Protocol for Scalable IoT Platform, Proceedings of the 6th International Conference on Information and Communication Technology (ICoICT), Bandung, Indonesia, May 2018, pp. 239 – 244.

[13] X. Liu, Y. Liu: Multi-Functional Serial Communication Interface Design Based on FPGA, Proceedings of the 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, December 2017, pp. 758 – 761.

[14] J. Qiang, Y. Gu, G. Chen: FPGA Implementation of SPI Bus Communication Based on State Machine Method, Journal of Physics: Conference Series, Vol. 1449, February 2020, p. 012027.

[15] D. Wang, J. Yan, Y. Qiao: Design and Verification of SPI Bus IP Core, Journal of Physics: Conference Series, Vol. 1971, July 2021, p. 012032.

[16] I.- B. Brezeanu, S. Isari, A. Vasile, G. Brezeanu: Design and Verification of a High Frequency, SPI Control Block, Proceedings of the International Semiconductor Conference (CAS), Sinaia, Romania, October 2023, pp. 175 – 178.

[17] H.- H. Thai, C.- K. Pham, D.- H. Le: Design of a Low-Power and Low-Area 8-Bit Flash ADC Using a Double-Tail Comparator on 180 nm CMOS Process, Sensors, Vol. 23, No. 1, January 2023, p. 76.

[18] M. Trehan, P. Kumar, N. Gaur: Design and Analysis of Multi-Protocol Conversion Unit for SPI, I2C and UART, Proceedings of the 2nd International Conference on Device Intelligence, Computing and Communication Technologies (DICCT), Dehradun, India, March 2024, pp. 324 – 329.

[19] D. Trivedi, A. Khade, K. Jain, R. Jadhav: SPI to I2C Protocol Conversion Using Verilog, Proceedings of the 4th International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, August 2018, pp. 1 – 4.

[20] D. R. da S. Medeiros, M. A. C. Fernandes: Distributed Genetic Algorithms for Low-Power, Low-Cost and Small-Sized Memory Devices, Electronics, Vol. 9, No. 11, November 2020, p. 1891.

[21] G. H. Makar, F. J. Badenas, R. G. Simone, A. Furfaro, K. S. Stevens, R. Suaya: Low Power SPI Design Based on Relative Timing Techniques, Proceedings of the 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS), Genoa, Italy, November 2019, pp. 166 – 169.

[22] Data Sheet AD7689-KGD: 16-Bit, 8-Channel, 250 kSPS PulSAR ADC, Analog Devices, 2022.

[23] AMD, AMD VivadoTM Design Suite, 2024.1, Available at:
https://www.xilinx.com/products/design-tools/vivado.html (accessed on 2024-07-29),

[24] Cadence, Genus Synthesis Solution, Available at:
https://www.cadence.com/en_US/home/tools/digital-design-and-signoff/synthesis/genus-synthesis-solution.html (accessed on 2024-07-29)

[25] Yosys Open Synthesis Suite, Available at:
https://yosyshq.net/yosys/ (accessed on 2024-10-22)

[26] Scientific-Manufacturing Complex ‘Technological Centre’: Integrated Circuits Department, Available at: http://www.asic.ru/ (accessed on 2024-10-22)