STUDIA UBB PHYSICA, Vol. 68 (LXVIII), 1-2, 2023, pp. 57-66
(RECOMMENDED CITATION)
doi:10.24193/subbphys.2023.06

EXPLORING THE FEASIBILITY OF PARALLEL I^{2}C SENSOR INTERFACING ON AN EDUCATIONAL FPGA BOARD: RESOURCE UTILIZATION ANALYSIS AND DESIGN CONSIDERATIONS

A. OPREA[1,2], B.S. KIREI[1,2], R.V.F. TURCU[1,3,*]

ABSTRACT. The goal of this work is to showcase the capabilities of Field Programmable Gate Arrays (FPGAs) as powerful reconfigurable computing devices for educational and research purposes. The work will demonstrate the high processing power of an FPGA by controlling multiple I^{2}C buses to and from peripheral sensors. The article will provide a guide on implementing a state machine in VHDL hardware description language and will present the response of the sensors and the working capacity of Basys 3 FPGA. The I^{2}C signals will be visualized using a Logic Analyzer and explained. The conclusion of the article will highlight the capabilities of FPGAs and explore the potential for their use in other complex tasks.

Keywords: *FPGA, sensor, I2C, fast processing, resource usage*

INTRODUCTION

Field-Programmable Gate Arrays (FPGAs) are widely utilized in physics research for their high-performance computing and processing capabilities [1], data acquisition and control systems [2]. In addition, this device can be reprogrammed in the field, which is a crucial feature for experiments where changes to the system

[1] *National Institute for Research and Development of Isotopic and Molecular Technologies, Centre of Advanced Research and Technologies for Alternative Energies (CETATEA), Cluj-Napoca, Romania*
[2] *Faculty of Electronics, Telecommunications and Information Technology, Technical University, Cluj-Napoca, Romania*
[3] *Faculty of Physics, Babeş-Bolyai University, Cluj-Napoca, Romania*
* *Corresponding author: flaviu.turcu@ubbcluj.ro*

are often necessary. In research, FPGAs have been used to implement high-speed digital signal processing systems for particle detectors [3], to control and synchronize multiple data acquisition systems [4], and to implement real-time data processing algorithms [5]. Moreover, FPGAs can directly interface with digital and analog components, such as sensors, facilitating rapid integration of new hardware into experimental setups. Table 1 presents a comparison of key resources between the Basys 3 FPGA and the Arduino Uno [6], [7], providing an overview of their capabilities for research purposes.

Table 1. Comparison between Basys 3 and Arduino Uno key resources

Function	Basys 3	Arduino
Work frequency	100 MHz	16 MHz
Volatile Memory	4.9 Mb block RAM	2Kb SRAM
Non-volatile Memory	180 Kb distributed	32 Kb flash memory
I2C maximum clock speed	700kHz	400kHz
Built-in DSP	Yes	No

This work analyzes the I2C protocol for real-time data transmission between a development board and two sensors using the Basys 3 FPGA [6] (Figure 1) The goal is to explore the potential of developing multiple sensor devices for complex applications, such as human-computer interfaces. The Basys 3 FPGA offers a wide range of peripherals and connectivity options, making it a versatile tool for various applications.

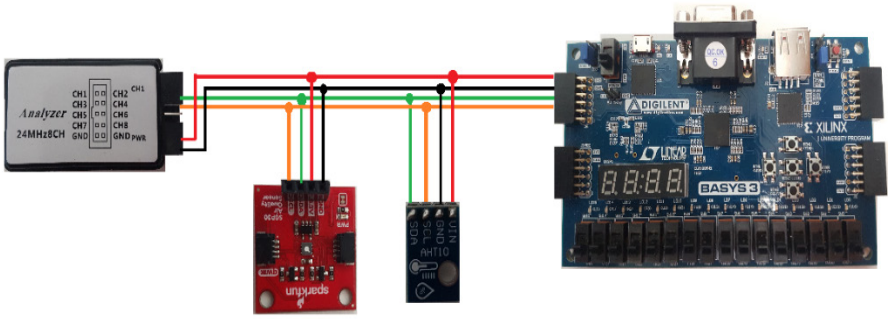

Fig. 1: External circuit designed for serial controlling of two sensors. From left to right: Logic Analyzer, SGP30 Air Quality sensor, AHT10 temperature and humidity sensor, Basys 3 board.

The AHT10 sensor [8] is a high-precision temperature and humidity sensor that uses Inter-Integrated Circuit (I^2C) protocol [9] for serial communication. It has a wide measurement range, high accuracy, and low power consumption. I^2C communication involves data, SDA, and a clock, SCLK, signal transferred bit by bit along a single wire. The frame format used by the protocol consists of a start bit , the I2C address, and write or read bits for commands or data transfer, respectively. The sensor responds with ACK bits and sends requested data bytes, with the session ending with a stop bit from the master device.

The SGP30 [10] (Figure 1) is a digital gas sensor used for measuring indoor air quality (IAQ) with high accuracy. It can detect a wide range of volatile organic compounds (VOCs) and gases, including carbon dioxide (CO_2). The sensor communicates via I^2C interface. It provides real-time output of IAQ measurements.

The FPGA configuration files are generated by the vendors proprietary software, in present case is the VIVADO design suite [11]. The hardware description language used for designing the digital circuits was VHDL. VHDL is a high-level hardware description language used to model digital circuits and systems. It is used to design complex digital systems by breaking them down into smaller, more manageable parts. In VHDL, designers use constructs like processes, signals, and components to describe the behavior of a digital system [12]. Concurrent processes in digital circuits execute in parallel and communicate using signals. Signals represent inputs/outputs and transmit data between processes. Components represent reusable hardware blocks that can be instantiated multiple times. VHDL is commonly used to implement controller modules for I^2C communication. These modules are designed as state machines that manage different protocol phases, such as start/stop condition, address phase, and data phase. State transitions are based on signal timing. The controller also includes logic for generating SCL clock signal based on desired frequency.

EXPERIMENTAL DETAILS

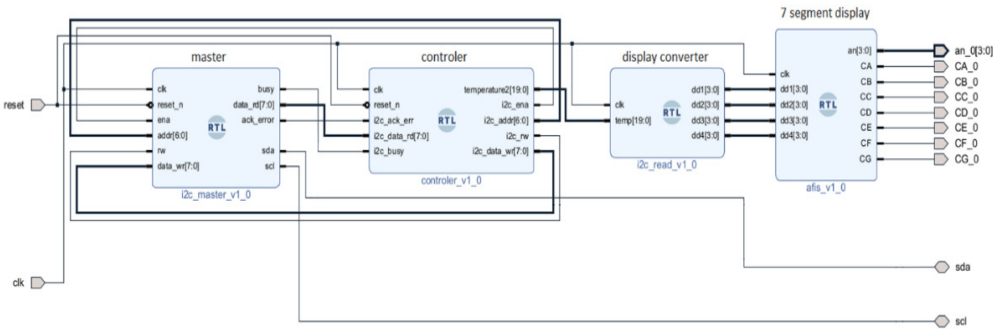

Fig. 2: Internal circuit made with VHDL for serial controlling of two sensors. From left to right, the following components are presented: master, controller, display converter and 7-segment display.

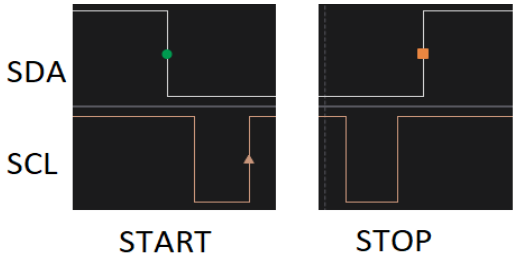

Fig. 3: Example of start and stop command. Falling edge of sda when scl is high, sets start command and rising edge of sda when scl is high sets stop command.

Figure 2 contains the main internal modules designed for controlling the sensor. The master is responsible for direct data transfer between board and sensors over SDA and SCL lines. Controller manages different states of the process and display modules help visualizing processed data. Start and stop conditions are depicted in Figure 3 below:

The two sensors' protocols are similar and depicted in Figure 4 using an algorithm for a finite state machine managed by the controller that guides the master through communication with sensor. The system starts in an idle state, then triggers the master to send slave address and command. After a pause for sensor data acquisition, the controller enters read command state, sends sensor address and read bit '1'. In read data state, the controller waits for master to send data and sends acknowledgement after each byte. Transaction ends with NACK bit

and stop. The external circuit for controlling AHT10 and SGP30 is shown in Figure 1. Signals captured with a Salea Logic Analyzer are displayed in Figure 5 [14]. It's important to note that AHT10 and SGP30 have different data acquisition times after triggering, with SGP30 requiring a longer waiting time of 12 ms compared to AHT10, which requires a significantly shorter time of less than 1 ms [8], [10].

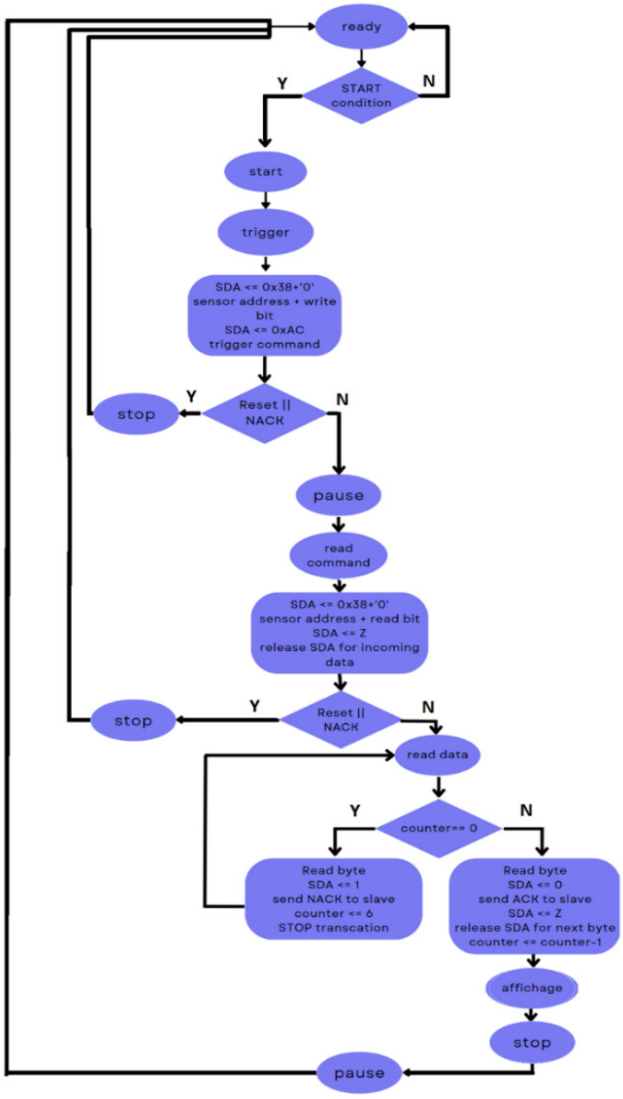

Fig. 4: Finite state machine implemented for controller communication with master. It follows all steps between start and data read and affichage and accounts for any NACK or reset.

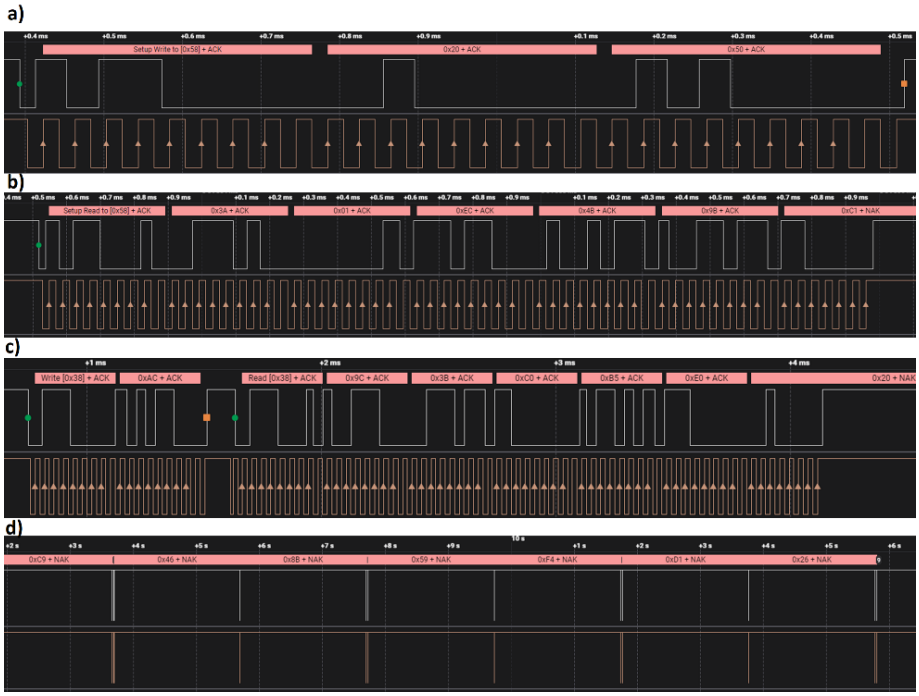

Fig. 5: SDA and SCL signals for serial communication**. a)**: Start transaction and data bytes sent from master to SGP30 sensor: 0x58 address + '0' write bit followed by '0' ACK bit sent from sensor; **b)**: command bytes: 0x20 and 0x50 followed by ACK from slave. Second: Data transfer from SGP30 sensor and stop command: 0x58 address + '1' read bit followed by '0' ACK bit sent from sensor; 5 data bytes sent from sensor to master followed by ACK sent from master, last data byte followed by stop command; **c)**: Full transaction between master and AHT10 sensor: Start command followed by address 0x38 and trigger command 0xAC. After 1ms pause the master sends address with read bit and starts receiving 6 bytes of data than it stops the transaction by sending NACK bit which means it sets sda to high impedance 'Z'; **d)**: Transaction buses over time for common sda and scl wires, the board alternates between the two sensors.

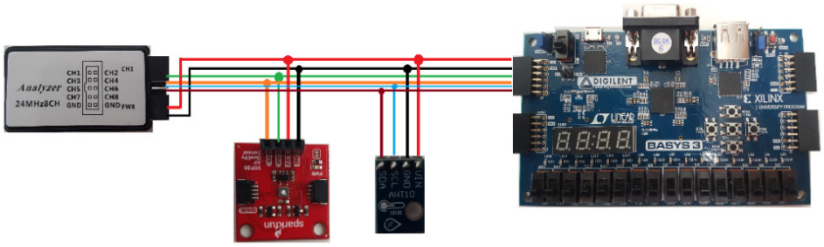

Fig. 6: External circuit designed for parallel controlling of the two sensors. From left to right: Logic Analyzer, SGP30 sensor, AHT10 sensor, Basys 3 board.

To resolve potential delays in processing multiple signals, the design was modified to utilize multiple PMOD ports and enable simultaneous data acquisition from both sensors using separate SDA and SCL wires for each sensor [15].

The updated design in Figure 6 incorporates additional pins on the Basys 3 board for the new SDA and SCL wires. Figure 7 shows the internal circuit of the parallel processing design with two masters, one for each sensor, allowing for customized working frequencies. Concurrent data transfer is illustrated in Figure 8, with data from each sensor transmitted via its respective SDA.

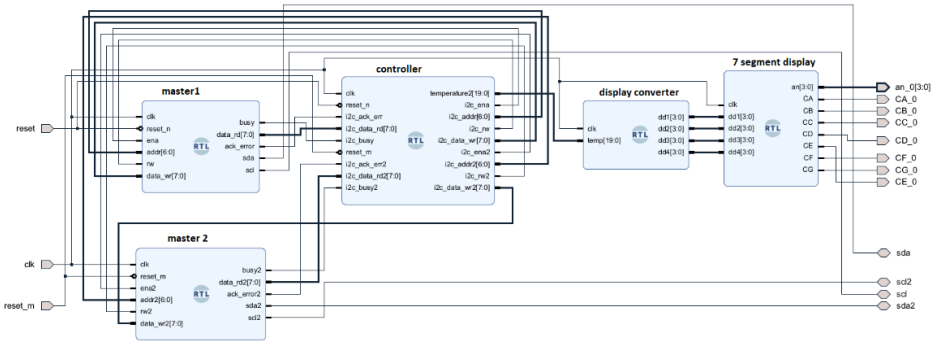

Fig. 7: Internal circuit made with VHDL for parallel controlling of two sensors. From left to right: two masters, controller, display modules.

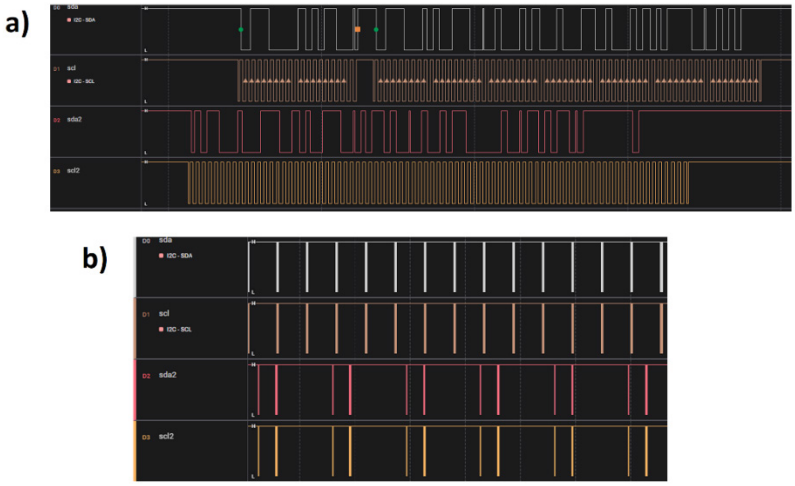

Fig. 8 a): Parallel transaction between board and the two sensors, up: AHT10 sensor, down: SGP30 sensor. Data from both sensors is transmitted at the same time. This method allows for different work frequencies; **b):** Transaction buses over time for parallel communication, the board uses multiple SDA and SCL wires and receives data from multiple sensors at the same time.

RESULTS AND DISCUSSION

Table 2 summarizes resource utilization in various scenarios, including multiple sensors and a simpler scenario with one sensor. Data was obtained from Vivado's report after bitstream generation, showing <0.5% increase in resource usage with new sensors or design development. Resource utilization is reported for three I^2C working frequencies (100 kHz, 400 kHz, and 700 kHz). Resource demand does not exhibit linear behavior with increasing I^2C working frequency. Metrics include Slice LUTs, Slice Registers, Slice, and Bonded IOB.

Table 2. VIVADO Summary of resource demand for driving one, two serial and two parallel sensors.

	Single sensor				Double sensor serial				Double sensor parallel			
Site Type	Frequency (kHz)	Used	Available	Utilized (%)	Frequency (kHz)	Used	Available	Utilized (%)	Frequency (kHz)	Used	Available	Utilized (%)
Slice LUTs	100	3364	20800	16.17	100	3415	20800	16.42	100	3507	20800	16.86
	400	3364		16.18	400	3414		16.41	400	3505		16.85
	700	3367		16.19	700	3415		16.42	700	3507		16.86
Slice Registers	100	166	41600	0.4	100	203	41600	0.49	100	246	41600	0.59
	400	164		0.39	400	201		0.48	400	242		0.58
	700	164		0.39	700	201		0.48	700	242		0.58
Slice	100	1004	8150	12.32	100	990	8150	12.15	100	1030	8150	12.64
	400	971		11.91	400	990		12.15	400	1055		12.94
	700	990		12.15	700	988		12.12	700	1035		12.7
Bonded IOB	-	17	106	16.04	-	17	106	16.04	-	18	106	16.98

The Slice LUTs are basic building blocks in an FPGA for implementing combinational logic functions; SliceRegisters (FFs) are Flip-flops used for storing data and implementing sequential logic; Slice are configurable portion of an FPGA for implementing digital circuits and Bonded IOB represent specialized blocks for interfacing with external signals.

The analysis indicates that the Basys 3 FPGA board can handle more complex tasks and has high performance potential for multi-sensor applications, as demonstrated by successful operation at 700 kHz I^2C frequency.

CONCLUSIONS

This study introduced an educational FPGA board as a powerful tool for sensor data recording and processing in experimental physics. The I^2C communication protocol and specifics of communication with the sensors were described. Data recording was successfully achieved at various SCL frequencies. The findings suggest potential for research, education, and industrial applications where high-speed sensor

communication and processing capabilities are needed. Further exploration could unlock the Basys 3 FPGA's full potential for complex tasks in embedded systems and digital design. More information about the VHDL code used can be found at [16].

Results highlight the versatility and suitability of the Basys 3 FPGA for advanced future applications requiring simultaneous control and communication with multiple sensors. A proposed such application is a human-machine interface in the shape of a pointer incorporating multiple positioning sensors and accelerometers for real time 3D painting in the air. This application could be a helpful tool in fields like education, arts, or presentations. The interface would allow users to create different shapes by moving the pointer in the air, capturing the position and acceleration data from the sensors and visualizing them on a display or with the help of VR glasses.

REFERENCES

[1] M. I. AlAli, K. M. Mhaidat, and I. A. Aljarrah, "Implementing image processing algorithms in FPGA hardware," in *2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, Amman, Jordan: IEEE, Dec. 2013, pp. 1–5. doi: 10.1109/AEECT.2013.6716446.

[2] S. Bao, H. Yan, Q. Chi, Z. Pang, and Y. Sun, "FPGA-Based Reconfigurable Data Acquisition System for Industrial Sensors," *IEEE Trans. Ind. Inform.*, vol. 13, no. 4, pp. 1503–1512, Aug. 2017, doi: 10.1109/TII.2016.2641462.

[3] L. Musa, "FPGAS in high energy physics experiments at CERN," in *2008 International Conference on Field Programmable Logic and Applications*, Heidelberg, Germany: IEEE, 2008, pp. 2–2. doi: 10.1109/FPL.2008.4629896.

[4] K. T. Pozniak, "FPGA-based, specialized trigger and data acquisition systems for high-energy physics experiments," *Meas. Sci. Technol.*, vol. 21, no. 6, p. 062002, Jun. 2010, doi: 10.1088/0957-0233/21/6/062002.

[5] G. Korcyl *et al.*, "Evaluation of Single-Chip, Real-Time Tomographic Data Processing on FPGA SoC Devices," *IEEE Trans. Med. Imaging*, vol. 37, no. 11, pp. 2526–2535, Nov. 2018, doi: 10.1109/TMI.2018.2837741.

[6] Digilent, "Basys 3 Reference Manual." https://digilent.com/reference/programmable-logic/basys-3/reference-manual?redirect=1

[7] ATMega, "Arduino® UNO R3 Product Reference Manual." [Online]. Available: https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf

[8] Asair, "AHT10 Technical Manual ASAIR® Temperature and humidity sensor," [Online]. Available: https://server4.eca.ir/eshop/AHT10/Aosong_AHT10_en_draft_0c.pdf

[9] Philips Semiconductors, *The I2c-bus specification*. 1998.

[10] SENSIRION, "Datasheet SGP30 Indoor Air Quality Sensor for TVOC and CO2eq Measurements."

[11] Xilinx, “Vivado Design Suite.” [Online]. Available:
https://caxapa.ru/thumbs/486666/wp416-Vivado-Design-Suite.pdf
[12] nandland, “VHDL Tutorials and Examples.” https://nandland.com/learn-vhdl/
[13] sparkfun, “I2C Introduction.” https://learn.sparkfun.com/tutorials/i2c/all
[14] saleae, “Saleae Logic Analyzer.” [Online]. Available: https://www.saleae.com/
[15] Digilent, “Basys 3 Artix-7 FPGA Trainer Board.” [Online]. Available:
https://digilent.com/reference/programmable-logic/basys-3/start
[16] “I2C sensor control with FPGA.” https://github.com/AlexOProject/FPGA-I2c-sensor-controll.